SIEMENS

# SINUMERIK

# SINUMERIK 840D sl
# NC-Programmierung

Programmierhandbuch


Vorwort

Grundlegende Sicherheitshinweise 1

Grundlagen 2

Arbeitsvorbereitung 3

Tabellen 4

Anhang A


Gültig für

Steuerung
SINUMERIK 840D sl / 840DE sl

CNC-Software Version 4.93


12/2019
6FC5398-2EP40-0AA0



Промэлектроника, sales@prom-elec.com

## Rechtliche Hinweise

### Warnhinweiskonzept

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise zu Ihrer persönlichen Sicherheit sind durch ein Warndreieck hervorgehoben, Hinweise zu alleinigen Sachschäden stehen ohne Warndreieck. Je nach Gefährdungsstufe werden die Warnhinweise in abnehmender Reihenfolge wie folgt dargestellt.

| ⚠ GEFAHR |
| --- |
| bedeutet, dass Tod oder schwere Körperverletzung eintreten **wird**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden. |

| ⚠ WARNUNG |
| --- |
| bedeutet, dass Tod oder schwere Körperverletzung eintreten **kann**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden. |

| ⚠ VORSICHT |
| --- |
| bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden. |

| ACHTUNG |
| --- |
| bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden. |

Beim Auftreten mehrerer Gefährdungsstufen wird immer der Warnhinweis zur jeweils höchsten Stufe verwendet. Wenn in einem Warnhinweis mit dem Warndreieck vor Personenschäden gewarnt wird, dann kann im selben Warnhinweis zusätzlich eine Warnung vor Sachschäden angefügt sein.

### Qualifiziertes Personal

Das zu dieser Dokumentation zugehörige Produkt/System darf nur von für die jeweilige Aufgabenstellung **qualifiziertem Personal** gehandhabt werden unter Beachtung der für die jeweilige Aufgabenstellung zugehörigen Dokumentation, insbesondere der darin enthaltenen Sicherheits- und Warnhinweise. Qualifiziertes Personal ist auf Grund seiner Ausbildung und Erfahrung befähigt, im Umgang mit diesen Produkten/Systemen Risiken zu erkennen und mögliche Gefährdungen zu vermeiden.

### Bestimmungsgemäßer Gebrauch von Siemens-Produkten

Beachten Sie Folgendes:

| ⚠ WARNUNG |
| --- |
| Siemens-Produkte dürfen nur für die im Katalog und in der zugehörigen technischen Dokumentation vorgesehenen Einsatzfälle verwendet werden. Falls Fremdprodukte und -komponenten zum Einsatz kommen, müssen diese von Siemens empfohlen bzw. zugelassen sein. Der einwandfreie und sichere Betrieb der Produkte setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung, Montage, Installation, Inbetriebnahme, Bedienung und Instandhaltung voraus. Die zulässigen Umgebungsbedingungen müssen eingehalten werden. Hinweise in den zugehörigen Dokumentationen müssen beachtet werden. |

### Marken

Alle mit dem Schutzrechtsvermerk ® gekennzeichneten Bezeichnungen sind eingetragene Marken der Siemens AG. Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

### Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten.

Siemens AG
Digital Industries
Postfach 48 48
90026 NÜRNBERG
DEUTSCHLAND

Dokumentbestellnummer: 6FC5398-2EP40-0AA0
℗ 12/2019 Änderungen vorbehalten

Copyright © Siemens AG 2019.
Alle Rechte vorbehalten

# Vorwort

## SINUMERIK-Dokumentation

Die SINUMERIK-Dokumentation ist in folgende Kategorien gegliedert:

- Allgemeine Dokumentation/Kataloge
- Anwender-Dokumentation
- Hersteller-/Service-Dokumentation

## Weiterführende Informationen

Unter folgender Adresse (https://support.industry.siemens.com/cs/de/de/view/108464614) finden Sie Informationen zu den Themen:

- Dokumentation bestellen/Druckschriftenübersicht
- Weiterführende Links für den Download von Dokumenten
- Dokumentation online nutzen (Handbücher/Informationen finden und durchsuchen)

Bei Fragen zur technischen Dokumentation (z. B. Anregungen, Korrekturen) senden Sie eine E-Mail an folgende Adresse (mailto:docu.motioncontrol@siemens.com).

## mySupport/Dokumentation

Unter folgender Adresse (https://support.industry.siemens.com/My/ww/de/documentation) finden Sie Informationen, wie Sie Ihre Dokumentation auf Basis der Siemensinhalte individuell zusammenstellen und für die eigene Maschinendokumentation anpassen.

## Training

Unter folgender Adresse (http://www.siemens.de/sitrain) finden Sie Informationen zu SITRAIN - dem Training von Siemens für Produkte, Systeme und Lösungen der Antriebs- und Automatisierungstechnik.

## FAQs

Frequently Asked Questions finden Sie in den Service&Support-Seiten unter Produkt Support (https://support.industry.siemens.com/cs/de/de/ps/faq).

## SINUMERIK

Informationen zu SINUMERIK finden Sie unter folgender Adresse (http://www.siemens.de/sinumerik).

## Zielgruppe

Die vorliegende Druckschrift wendet sich an:

- Programmierer
- Projekteure

## Nutzen

Das Programmierhandbuch befähigt die Zielgruppe, Programme und Software-Oberflächen zu entwerfen, zu schreiben, zu testen und Fehler zu beheben.

## Standardumfang

In dem vorliegenden Programmierhandbuch ist die Funktionalität des Standardumfangs beschrieben. Ergänzungen oder Änderungen, die durch den Maschinenhersteller vorgenommen werden, werden vom Maschinenhersteller dokumentiert.

Es können in der Steuerung weitere, in dieser Dokumentation nicht erläuterte Funktionen ablauffähig sein. Es besteht jedoch kein Anspruch auf diese Funktionen bei der Neulieferung bzw. im Servicefall.

Ebenso enthält diese Dokumentation aus Gründen der Übersichtlichkeit nicht sämtliche Detailinformationen zu allen Typen des Produkts und kann auch nicht jeden denkbaren Fall der Aufstellung, des Betriebs und der Instandhaltung berücksichtigen.

## Hinweis zur Datenschutzgrundverordnung

Siemens beachtet die Grundsätze des Datenschutzes, insbesondere die Gebote der Datenminimierung (privacy by design). Für dieses Produkt bedeutet dies:

Das Produkt verarbeitet/speichert keine personenbezogenen Daten, lediglich technische Funktionsdaten (z. B. Zeitstempel). Verknüpft der Anwender diese Daten mit anderen Daten (z. B. Schichtpläne) oder speichert er personenbezogene Daten auf dem gleichen Medium (z. B. Festplatte) und stellt so einen Personenbezug her, hat er die Einhaltung der datenschutzrechtlichen Vorgaben selbst sicherzustellen.

## Technical Support

Landesspezifische Telefonnummern für technische Beratung finden Sie im Internet unter folgender Adresse (https://support.industry.siemens.com/sc/ww/de/sc/2090) im Bereich "Kontakt".

Um eine technische Frage zu stellen, nutzen Sie das Online-Formular im Bereich "Support Request".

## Informationen zu Struktur und Inhalt der Dokumentation

Die Dokumentation zur NC-Programmierung verteilt sich auf zwei Hauptkapitel:

1. **Grundlagen**
   Das Kapitel "Grundlagen" dient dem Maschinenfacharbeiter und setzt entsprechende Kenntnisse für Bohr-, Fräs- und Drehbearbeitungen voraus. An einfachen Programmierbeispielen werden die auch nach DIN 66025 bekannten Befehle und Anweisungen erläutert.
2. **Arbeitsvorbereitung**
   Das Kapitel "Arbeitsvorbereitung" dient dem Technologen mit Kenntnissen über die gesamten Programmiermöglichkeiten. Die SINUMERIK-Steuerung ermöglicht mit einer speziellen Programmiersprache die Programmierung eines komplexen Werkstückprogramms (z. B. Freiformflächen, Kanalkoordinierung, ...) und erleichtert dem Technologen eine aufwendige Programmierung.

# Inhaltsverzeichnis


Vorwort ..... 3

1 Grundlegende Sicherheitshinweise ..... 23

1.1 Allgemeine Sicherheitshinweise ..... 23

1.2 Gewährleistung und Haftung für Applikationsbeispiele ..... 24

1.3 Industrial Security ..... 25

2 Grundlagen ..... 27

2.1 Geometrische Grundlagen ..... 27
2.1.1 Werkstück-Positionen ..... 27
2.1.1.1 Bezugssystem der Positionsangaben ..... 27
2.1.1.2 Kartesische Koordinaten ..... 27
2.1.1.3 Polarkoordinaten ..... 29
2.1.1.4 Absolutmaß ..... 30
2.1.1.5 Kettenmaß ..... 32
2.1.2 Arbeitsebenen ..... 33
2.1.3 Nullpunkte und Bezugspunkte ..... 34
2.1.4 Koordinatensysteme ..... 36
2.1.4.1 Maschinen-Koordinatensystem (MKS) ..... 36
2.1.4.2 Basis-Koordinatensystem (BKS) ..... 38
2.1.4.3 Basis-Nullpunktsystem (BNS) ..... 39
2.1.4.4 Einstellbares Nullpunktsystem (ENS) ..... 40
2.1.4.5 Werkstück-Koordinatensystem (WKS) ..... 41
2.1.4.6 Wie hängen die verschiedenen Koordinatensysteme zusammen? ..... 42

2.2 Grundlagen der NC-Programmierung ..... 43
2.2.1 Benennung eines NC-Programms ..... 43
2.2.2 Aufbau und Inhalte eines NC-Programms ..... 44
2.2.2.1 Sätze und Satzkomponenten ..... 44
2.2.2.2 Satzregeln ..... 47
2.2.2.3 Wertzuweisungen ..... 48
2.2.2.4 Kommentare ..... 48
2.2.2.5 Ausblenden von Sätzen ..... 49

2.3 Anlegen eines NC-Programms ..... 51
2.3.1 Grundsätzliches Vorgehen ..... 51
2.3.2 Verfügbare Zeichen ..... 52
2.3.3 Programmkopf ..... 53
2.3.4 Programmbeispiele ..... 54
2.3.4.1 Beispiel 1: Erste Programmierschritte ..... 54
2.3.4.2 Beispiel 2: NC-Programm zum Drehen ..... 55
2.3.4.3 Beispiel 3: NC-Programm zum Fräsen ..... 57

2.4 Werkzeugwechsel ..... 60
2.4.1 Werkzeugwechsel bei aktiver Werkzeugverwaltung ..... 61
2.4.1.1 Werkzeugwechsel bei aktiver WZV mit der Funktion T=Platz ..... 62
2.4.1.2 Werkzeugwechsel bei aktiver WZV mit M6 ..... 64

2.4.2 Werkzeugwechsel bei nicht aktiver Werkzeugverwaltung ..... 65
2.4.2.1 Werkzeugwechsel bei nicht aktiver WZV mit T-Nummer ..... 65
2.4.2.2 Werkzeugwechsel bei nicht aktiver WZV mit M6 ..... 66
2.4.3 Verhalten bei fehlerhafter T-Programmierung ..... 67

2.5 Werkzeugkorrekturen ..... 68
2.5.1 Programmierte Kontur und Werkzeugbahn ..... 68
2.5.2 Werkzeuglängenkorrektur ..... 68
2.5.3 Werkzeugradiuskorrektur ..... 69
2.5.4 Werkzeug-Korrekturspeicher ..... 70
2.5.5 Werkzeugtypen ..... 71
2.5.5.1 Werkzeugtypnummer und Werkzeuggruppen ..... 71
2.5.5.2 Fräswerkzeuge ..... 72
2.5.5.3 Bohrer ..... 74
2.5.5.4 Schleifwerkzeuge ..... 75
2.5.5.5 Drehwerkzeuge ..... 77
2.5.5.6 Sonderwerkzeuge ..... 79
2.5.6 Werkzeugkorrekturen aktivieren/deaktivieren (D, D0) ..... 81
2.5.7 Werkzeugkorrekturen unterdrücken (SUPD) ..... 83
2.5.8 Programmierbarer Werkzeugkorrektur-Offset (TOFFL, TOFF, TOFFR, TOFFLR) ..... 85

2.6 Spindelbewegung ..... 91
2.6.1 Spindeldrehzahl (S), Spindeldrehrichtung (M3, M4, M5) ..... 91
2.6.2 Werkzeug-Schnittgeschwindigkeit (SVC) ..... 94
2.6.3 Konstante Schnittgeschwindigkeit (G96/G961/G962, G97/G971/G972, G973, LIMS, SCC) ..... 100
2.6.4 Konstante Scheibenumfangsgeschwindigkeit ein-/ausschalten (GWPSON, GWPSOF) ..... 105
2.6.5 Programmierbare Spindeldrehzahlbegrenzung (G25, G26) ..... 106

2.7 Vorschubregelung ..... 108
2.7.1 Vorschub (G93, G94, G95, F, FGROUP, FL, FGREF) ..... 108
2.7.2 Positionierachsen verfahren (POS, POSA, POSP, FA, WAITP, WAITMC) ..... 116
2.7.3 Lagegeregelter Spindelbetrieb (SPCON, SPCOF) ..... 120
2.7.4 Spindeln positionieren (SPOS, SPOSA, M19, M70, WAITS) ..... 121
2.7.5 Vorschub für Positionierachsen/Spindeln (FA, FPR, FPRAON, FPRAOF) ..... 126
2.7.6 Programmierbare Vorschubkorrektur (OVR, OVRRAP, OVRA) ..... 129
2.7.7 Programmierbare Beschleunigungskorrektur (ACC) ..... 130
2.7.8 Vorschub mit Handradüberlagerung (FD, FDA) ..... 131
2.7.9 Vorschuboptimierung bei gekrümmten Bahnstücken (CFTCP, CFC, CFIN) ..... 135
2.7.10 Mehrere Vorschubwerte in einem Satz (F, ST, SR, FMA, STA, SRA) ..... 137
2.7.11 Satzweiser Vorschub (FB) ..... 140
2.7.12 Zahnvorschub (G95 FZ) ..... 141

2.8 Geometrie-Einstellungen ..... 147
2.8.1 Einstellbare Nullpunktverschiebung (G54 ... G57, G505 ... G599, G53, G500, SUPA, G153) ..... 147
2.8.2 Einstellbare Nullpunktverschiebung (G54 ... G57, G505 ... G599, G53, G500, SUPA, G153): Weitere Informationen ..... 148
2.8.3 Wahl der Arbeitsebene (G17/G18/G19) ..... 149
2.8.4 Maßangaben ..... 152
2.8.4.1 Absolutmaßangabe (G90, AC) ..... 153
2.8.4.2 Kettenmaßangabe (G91, IC) ..... 155
2.8.4.3 Absolut- und Kettenmaßangabe beim Drehen und Fräsen (G90/G91) ..... 159
2.8.4.4 Absolutmaßangabe für Rundachsen (DC, ACP, ACN) ..... 160

2.8.4.5 Maßsystem inch/metrisch (G70/G71, G700/G710) ... 162
2.8.4.6 Kanalspezifische Durchmesser-/Radius-Programmierung (DIAMON, DIAM90, DIAMOF, DIAMCYCOF) ... 165
2.8.4.7 Achsspezifische Durchmesser-/Radius-Programmierung (DIAMONA, DIAM90A, DIAMOFA, DIACYCOFA, DIAMCHANA, DIAMCHAN, DAC, DIC, RAC, RIC) ... 167
2.8.5 Lage des Werkstücks beim Drehen ... 171

2.9 Wegbefehle ... 173
2.9.1 Allgemeine Informationen zu den Wegbefehlen ... 173
2.9.2 Fahrbefehle mit kartesischen Koordinaten (G0, G1, G2, G3, X..., Y..., Z...) ... 175
2.9.3 Fahrbefehle mit Polarkoordinaten ... 176
2.9.3.1 Bezugspunkt der Polarkoordinaten (G110, G111, G112) ... 176
2.9.3.2 Fahrbefehle mit Polarkoordinaten (G0, G1, G2, G3, AP, RP) ... 178
2.9.4 Eilgangbewegungen ... 182
2.9.4.1 Eilgang aktivieren (G0) ... 182
2.9.4.2 Lineare Interpolation für Eilgangbewegungen ein-/ausschalten (RTLION, RTLIOF) ... 183
2.9.4.3 Toleranzen für Eilgangbewegungen anpassen (STOLF, CTOLG0, OTOLG0) ... 185
2.9.5 Geradeninterpolation (G1) ... 189
2.9.6 Kreisinterpolation ... 191
2.9.6.1 Übersicht ... 191
2.9.6.2 Kreisinterpolation mit Mittelpunkt und Endpunkt (G2/G3, X... Y... Z..., I... J... K...) ... 192
2.9.6.3 Kreisinterpolation mit Radius und Endpunkt (G2/G3, X... Y... Z..., CR) ... 195
2.9.6.4 Kreisinterpolation mit Öffnungswinkel und Endpunkt / Mittelpunkt (G2/G3, X... Y... Z... / I... J... K..., AR) ... 197
2.9.6.5 Kreisinterpolation mit Polarkoordinaten (G2/G3, AP, RP) ... 199
2.9.6.6 Kreisinterpolation mit Zwischen- und Endpunkt (CIP, X... Y... Z..., I1... J1... K1...) ... 200
2.9.6.7 Kreisinterpolation mit tangentialem Übergang (CT, X... Y... Z...) ... 203
2.9.7 Schraubenlinien-Interpolation (G2/G3, TURN) ... 207
2.9.8 Evolventen-Interpolation (INVCW, INVCCW) ... 209
2.9.9 Konturzüge ... 214
2.9.9.1 Konturzug-Programmierung ... 214
2.9.9.2 Konturzüge: Eine Gerade ... 215
2.9.9.3 Konturzüge: Zwei Geraden ... 217
2.9.9.4 Konturzüge: Drei Geraden ... 220
2.9.9.5 Konturzüge: Endpunktprogrammierung mit Winkel ... 223
2.9.10 Gewindeschneiden ... 224
2.9.10.1 Gewindeschneiden mit konstanter Steigung (G33, SF) ... 224
2.9.10.2 Gewindeschneiden mit zu- oder abnehmender Steigung (G34, G35) ... 231
2.9.10.3 Programmierter Einlauf- und Auslaufweg bei G33, G34 und G35 (DITS, DITE) ... 232
2.9.10.4 Schnellrückzug während Gewindeschneiden (LFON, LFOF, DILF, ALF, LFTXT, LFWP, LFPOS, POLF, POLFMASK, POLFMLIN) ... 234
2.9.10.5 Ballige Gewinde (G335, G336) ... 238
2.9.11 Gewindebohren ohne Ausgleichsfutter ... 243
2.9.11.1 Gewindebohren ohne Ausgleichsfutter und Rückzugsbewegung (G331, G332) ... 243
2.9.11.2 Beispiel: Gewindebohren mit G331 / G332 ... 244
2.9.11.3 Beispiel: Programmierte Bohrdrehzahl in der aktuellen Getriebestufe ausgeben ... 245
2.9.11.4 Beispiel: Anwendung des zweiten Getriebestufen-Datensatzes ... 245
2.9.11.5 Beispiel: Keine Drehzahlprogrammierung, Überwachung der Getriebestufe ... 246
2.9.11.6 Beispiel: Getriebestufenwechsel nicht möglich, Überwachung der Getriebestufe ... 246
2.9.11.7 Beispiel: Programmierung ohne SPOS ... 247
2.9.12 Gewindebohren mit Ausgleichsfutter ... 247
2.9.12.1 Gewindebohren mit Ausgleichsfutter und Rückzugsbewegung (G63) ... 247
2.9.13 Fase, Rundung (CHF, CHR, RND, RNDM, FRC, FRCM) ... 248

2.10 Werkzeugradiuskorrekturen ....254
2.10.1 Werkzeugradiuskorrektur (G40, G41, G42, OFFN) ....254
2.10.2 Kontur anfahren und verlassen (NORM, KONT, KONTC, KONTT) ....263
2.10.3 Korrektur an den Außenecken (G450, G451, DISC) ....270
2.10.4 Weiches An- und Abfahren ....273
2.10.4.1 An- und Abfahren (G140 bis G143, G147, G148, G247, G248, G347, G348, G340, G341, DISR, DISCL, DISRP, FAD, PM, PR) ....273
2.10.4.2 An- und Abfahren mit erweiterten Abfahrstrategien (G460, G461, G462) ....284
2.10.5 Kollisionsüberwachung ("Flaschenhalserkennung") ein-/ausschalten (CDON, CDOF, CDOF2) ....287
2.10.6 2 1/2 D-Werkzeugkorrektur (CUT2D, CUT2DD, CUT2DF, CUT2DFD) ....289
2.10.7 Werkzeugradiuskorrektur konstant halten (CUTCONON, CUTCONOF) ....291
2.10.8 Werkzeuge mit relevanter Schneidenlage ....293

2.11 Bahnfahrverhalten ....296
2.11.1 Genauhalt (G60, G9, G601, G602, G603) ....296
2.11.2 Bahnsteuerbetrieb (G64, G641, G642, G643, G644, G645, ADIS, ADISPOS) ....298

2.12 Koordinatentransformationen (Frames) ....308
2.12.1 Frames ....308
2.12.2 Frame-Anweisungen ....310
2.12.3 Programmierbare Nullpunktverschiebung (TRANS, ATRANS) ....313
2.12.4 Programmierbare Nullpunktverschiebung (G58, G59) ....317
2.12.5 Programmierbare Drehung (ROT, AROT, RPL) ....319
2.12.6 Programmierbare Framedrehungen mit Raumwinkeln (ROTS, AROTS, CROTS) ....325
2.12.7 Programmierbarer Maßstabsfaktor (SCALE, ASCALE) ....328
2.12.8 Programmierbare Spiegelung (MIRROR, AMIRROR) ....331
2.12.9 Frame-Erzeugung nach Werkzeugausrichtung (TOFRAME, TOROT, PAROT) ....336
2.12.10 Frame abwählen (G53, G153, SUPA, G500) ....339
2.12.11 Programmierung: Überlagerungen achsspezifisch abwählen (CORROF) ....340
2.12.12 Additive Nullpunktverschiebungen abwählen (DRFOF) ....342
2.12.13 Schleifspezifische Nullpunktverschiebungen (GFRAME0, GFRAME1 ... GFRAME100) ....344

2.13 Hilfsfunktionsausgaben ....345
2.13.1 M-Funktionen ....348

2.14 Ergänzende Befehle ....352
2.14.1 Meldung ausgeben (MSG) ....352
2.14.2 Zeichenkette in BTSS-Variable schreiben (WRTPR) ....353
2.14.3 Arbeitsfeldbegrenzung ....355
2.14.3.1 Arbeitsfeldbegrenzung im BKS (G25/G26, WALIMON, WALIMOF) ....355
2.14.3.2 Arbeitsfeldbegrenzung im WKS/ENS (WALCS0 ... WALCS10) ....358
2.14.4 Referenzpunktfahren (G74) ....361
2.14.5 Festpunkt anfahren (G75) ....362
2.14.6 Fahren auf Festanschlag (FXS, FXST, FXSW) ....366
2.14.7 Verweilzeit (G4) ....371
2.14.8 Interner Vorlaufstopp ....373

2.15 Sonstige Informationen ....374
2.15.1 Achsen ....374
2.15.1.1 Hauptachsen/Geometrieachsen ....374
2.15.1.2 Zusatzachsen ....376
2.15.1.3 Hauptspindel, Masterspindel ....376
2.15.1.4 Maschinenachsen ....376

2.15.1.5 Kanalachsen ....377
2.15.1.6 Bahnachsen ....377
2.15.1.7 Positionierachsen ....377
2.15.1.8 Synchronachsen ....378
2.15.1.9 Kommandoachsen ....379
2.15.1.10 PLC-Achsen ....379
2.15.1.11 Linkachsen ....379
2.15.1.12 Lead-Linkachsen ....381
2.15.2 Vom Fahrbefehl zur Maschinenbewegung ....383
2.15.3 Wegberechnung ....383
2.15.4 Adressen ....384
2.15.5 Namen ....386
2.15.6 Konstanten ....388
2.15.7 Operatoren und Rechenfunktionen ....389

**3 Arbeitsvorbereitung ....393**

3.1 Flexible NC-Programmierung ....393
3.1.1 Variablen ....393
3.1.1.1 Systemdaten ....394
3.1.1.2 Vordefinierte Anwendervariablen: Kanalspezifische Rechenparameter (R) ....396
3.1.1.3 Vordefinierte Anwendervariablen: Globale Rechenparameter (RG) ....398
3.1.1.4 Vordefinierte Anwendervariablen: Link-Variablen ....399
3.1.1.5 Definition von Anwendervariablen (DEF) ....401
3.1.1.6 Redefinition von Systemdaten, Anwenderdaten und NC-Sprachbefehlen (REDEF) ....407
3.1.1.7 Attribut: Initialisierungswert ....410
3.1.1.8 Attribut: Grenzwerte (LLI, ULI) ....413
3.1.1.9 Attribut: Physikalische Einheit (PHU) ....415
3.1.1.10 Attribut: Zugriffsrechte (APR, APW, APRP, APWP, APRB, APWB) ....417
3.1.1.11 Übersicht definierbarer und redefinierbarer Attribute ....421
3.1.1.12 Definition und Initialisierung von Feldvariablen (DEF, SET, REP) ....422
3.1.1.13 Datentypen ....428
3.1.1.14 Minimum, Maximum und Bereich von Variablen (MINVAL, MAXVAL, BOUND) ....429
3.1.1.15 Vorhandensein einer Variablen prüfen (ISVAR) ....430
3.1.1.16 Attributwerte/Datentyp lesen (GETVARPHU, GETVARAP, GETVARLIM, GETVARDIM, GETVARDFT, GETVARTYP) ....432
3.1.1.17 Mögliche Typenkonvertierungen ....437
3.1.2 Indirekte Programmierung ....438
3.1.2.1 Indirekte Programmierung von Adressen ....438
3.1.2.2 Indirekte Programmierung von G-Befehlen ....441
3.1.2.3 Indirekte Programmierung von Positionsattributen (GP) ....442
3.1.2.4 Indirekte Programmierung von Teileprogrammzeilen (EXECSTRING) ....444
3.1.3 Operationen ....445
3.1.3.1 Rechenfunktionen ....445
3.1.3.2 Vergleichs- und logische Operationen ....447
3.1.3.3 Priorität der Operationen ....449
3.1.3.4 Genauigkeitskorrektur bei Vergleichsfehlern (TRUNC) ....450
3.1.3.5 Aufrunden (ROUNDUP) ....451
3.1.4 Stringoperationen ....452
3.1.4.1 Typenkonvertierung nach STRING (AXSTRING) ....453
3.1.4.2 Typenkonvertierung von STRING (NUMBER, ISNUMBER, AXNAME) ....454
3.1.4.3 Verkettung von Strings (<<) ....455
3.1.4.4 Wandlung in Klein-/Großbuchstaben (TOLOWER, TOUPPER) ....456

3.1.4.5 Länge eines Strings bestimmen (STRLEN) ....457
3.1.4.6 Zeichen/String in String suchen (INDEX, RINDEX, MINDEX, MATCH) ....457
3.1.4.7 Auswahl eines Teilstrings (SUBSTR) ....458
3.1.4.8 Lesen und Schreiben von einzelnen Zeichen ....459
3.1.4.9 String formatieren (SPRINT) ....461
3.1.5 Programmsprünge und -verzweigungen ....469
3.1.5.1 Rücksprung auf Programmanfang (GOTOS) ....469
3.1.5.2 Programmsprünge auf Sprungmarken (GOTOB, GOTOF, GOTO, GOTOC) ....470
3.1.5.3 Programmverzweigung (CASE ... OF ... DEFAULT ...) ....473
3.1.6 Programmteilwiederholung (REPEAT, REPEATB, ENDLABEL, P) ....475
3.1.7 Kontrollstrukturen ....480
3.1.7.1 Bedingte Anweisung und Verzweigung (IF, ELSE, ENDIF) ....482
3.1.7.2 Endlos-Programmschleife (LOOP, ENDLOOP) ....484
3.1.7.3 Zählschleife (FOR ... TO ..., ENDFOR) ....485
3.1.7.4 Programmschleife mit Bedingung am Schleifenanfang (WHILE, ENDWHILE) ....486
3.1.7.5 Programmschleife mit Bedingung am Schleifenende (REPEAT, UNTIL) ....487
3.1.7.6 Programmbeispiel mit verschachtelten Kontrollstrukturen ....487
3.1.8 Kanalübergreifende Programmkoordinierung (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) ....488
3.1.9 Makrotechnik (DEFINE ... AS) ....494

3.2 Unterprogrammtechnik ....497
3.2.1 Allgemeines ....497
3.2.1.1 Unterprogramm ....497
3.2.1.2 Unterprogrammnamen ....498
3.2.1.3 Schachtelung von Unterprogrammen ....499
3.2.1.4 Suchpfad ....500
3.2.1.5 Formal- und Aktualparameter ....501
3.2.1.6 Parameterübergabe ....501
3.2.2 Definition eines Unterprogramms ....503
3.2.2.1 Unterprogramm ohne Parameterübergabe ....503
3.2.2.2 Unterprogramm mit Parameterübergabe Call-by-Value (PROC) ....504
3.2.2.3 Unterprogramm mit Parameterübergabe Call-by-Reference (PROC, VAR) ....506
3.2.2.4 Modale G-Funktionen sichern (SAVE) ....508
3.2.2.5 Einzelsatzbearbeitung unterdrücken (SBLOF, SBLON) ....509
3.2.2.6 Aktuelle Satzanzeige unterdrücken (DISPLOF, DISPLON, ACTBLOCNO) ....514
3.2.2.7 Unterprogramme mit Vorbereitung kennzeichnen (PREPRO) ....517
3.2.2.8 Unterprogrammrücksprung M17 ....518
3.2.2.9 Unterprogrammrücksprung RET ....519
3.2.2.10 Parametrierbarer Unterprogrammrücksprung (RET ...) ....520
3.2.2.11 Parametrierbarer Unterprogrammrücksprung (RETB ...) ....526
3.2.3 Aufruf eines Unterprogramms ....530
3.2.3.1 Unterprogrammaufruf ohne Parameterübergabe ....530
3.2.3.2 Unterprogrammaufruf mit Parameterübergabe (EXTERN) ....532
3.2.3.3 Anzahl der Programmwiederholungen (P) ....534
3.2.3.4 Modaler Unterprogrammaufruf (MCALL) ....535
3.2.3.5 Indirekter Unterprogrammaufruf (CALL) ....538
3.2.3.6 Indirekter Unterprogrammaufruf mit Angabe des auszuführenden Programmteils (CALL BLOCK ... TO ...) ....539
3.2.3.7 Indirekter Aufruf eines in ISO-Sprache programmierten Programms (ISOCALL) ....540
3.2.3.8 Unterprogramm mit Pfadangabe und Parametern aufrufen (PCALL) ....541
3.2.3.9 Suchpfad bei Unterprogrammaufrufen erweitern (CALLPATH) ....541
3.2.3.10 Externes Unterprogramm abarbeiten (EXTCALL) ....543

3.3 Interruptroutine (ASUP) ..... 547
3.3.1 Funktion einer Interruptroutine ..... 547
3.3.2 Interruptroutine erstellen ..... 548
3.3.3 Interruptroutine zuordnen und starten (SETINT, PRIO, BLSYNC) ..... 548
3.3.4 Zuordnung einer Interruptroutine deaktivieren/reaktivieren (DISABLE, ENABLE) ..... 551
3.3.5 Zuordnung einer Interruptroutine löschen (CLRINT) ..... 551
3.3.6 Schnellabheben von der Kontur (SETINT LIFTFAST, ALF) ..... 552
3.3.7 Verfahrrichtung beim Schnellabheben von der Kontur ..... 554
3.3.8 Bewegungsablauf bei Interruptroutinen ..... 557

3.4 Datei- und Programmverwaltung ..... 559
3.4.1 Programmspeicher ..... 559
3.4.1.1 Programmspeicher im NCK ..... 559
3.4.1.2 Externe Programmspeicher ..... 561
3.4.1.3 Adressierung von Dateien des Programmspeichers ..... 563
3.4.1.4 Suchpfad bei Unterprogrammaufruf ..... 568
3.4.1.5 Abfrage von Pfad und Dateiname ..... 569
3.4.2 Arbeitsspeicher (CHANDATA, COMPLETE, INITIAL) ..... 570

3.5 Dateihandling ..... 574
3.5.1 Datei schreiben (WRITE) ..... 574
3.5.2 Datei löschen (DELETE) ..... 578
3.5.3 Zeilen in Datei lesen (READ) ..... 579
3.5.4 Vorhandensein einer Datei prüfen (ISFILE) ..... 581
3.5.5 Datei-Informationen auslesen (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO) ..... 582

3.6 Schutzbereiche ..... 584
3.6.1 Schutzbereiche definieren (CPROTDEF, NPROTDEF) ..... 584
3.6.2 Schutzbereiche aktivieren/deaktivieren (CPROT, NPROT) ..... 587
3.6.3 Überprüfung auf Schutzbereichsverletzung, Arbeitsfeldbegrenzung und Software-Endschalter (CALCPOSI) ..... 591

3.7 Spezielle Wegbefehle ..... 601
3.7.1 Codierte Positionen anfahren (CAC, CIC, CDC, CACP, CACN) ..... 601
3.7.2 Spline-Interpolation (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) ..... 601
3.7.3 Spline-Verbund (SPLINEPATH) ..... 612
3.7.4 NC-Satz-Kompression ein-/ausschalten (COMPON, COMPCURV, COMPCAD, COMPSURF, COMPOF) ..... 613
3.7.5 Polynom-Interpolation (POLY, POLYPATH, PO, PL) ..... 614
3.7.6 Einstellbarer Bahnbezug (SPATH, UPATH) ..... 619
3.7.7 Kanalspezifisches Messen (MEAS, MEAW) ..... 621
3.7.8 Achsspezifisches Messen (MEASA, MEAWA, MEAC) (Option) ..... 624
3.7.9 Spezielle Funktionen für den OEM-Anwender (OMA1 ... OMA5, OEMIPO1, OEMIPO2, G810 ... G829) ..... 632
3.7.10 Vorschubreduzierung mit Eckenverzögerung (FENDNORM, G62, G621) ..... 633
3.7.11 Programmierbares Bewegungsendekriterium (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA) ..... 634

3.8 Koordinatentransformationen (Frames) ..... 637
3.8.1 Koordinatentransformation über Framevariable ..... 637
3.8.1.1 Vordefinierte Framevariable ($P_CHBFRAME, $P_IFRAME, $P_PFRAME, $P_ACTFRAME) ..... 639

3.8.2 Wertzuweisungen an Frames ....642
3.8.2.1 Direkte Werte zuweisen (Achswert, Winkel, Maßstab) ....642
3.8.2.2 Framekomponenten lesen und verändern (TR, FI, RT, SC, MI) ....644
3.8.2.3 Rechnen mit Frames ....646
3.8.2.4 Definition von Framevariablen (DEF FRAME) ....647
3.8.3 Grob- und Feinverschiebung (CTRANS, CFINE) ....648
3.8.4 Externe Nullpunktverschiebung ($AA_ETRANS) ....649
3.8.5 Istwertsetzen mit Verlust des Referenzierstatus (PRESETON) ....650
3.8.6 Istwertsetzen ohne Verlust des Referenzierstatus (PRESETONS) ....652
3.8.7 Frame-Berechnung aus 3 Messpunkten im Raum (MEAFRAME) ....653
3.8.8 Globale Frames ....657
3.8.8.1 Kanalspezifische Frames ($P_CHBFR, $P_UBFR) ....658
3.8.8.2 Im Kanal wirksame Frames ....658

3.9 Transformationen ....664
3.9.1 Allgemeine Programmierung der Transformationsarten ....664
3.9.1.1 Orientierungsbewegungen bei den Transformationen ....666
3.9.1.2 Übersicht der Orientierungstransformation TRAORI ....670
3.9.2 Drei-, Vier- und Fünf-Achs-Transformation (TRAORI) ....671
3.9.2.1 Allgemeine Zusammenhänge Kardanischer Werkzeugkopf ....671
3.9.2.2 Drei, Vier, und Fünf- Achs-Transformation (TRAORI) ....674
3.9.2.3 Varianten der Orientierungsprogrammierung und Grundstellung (ORIRESET) ....676
3.9.2.4 Programmierung der Werkzeugorientierung (A..., B..., C..., LEAD, TILT) ....677
3.9.2.5 Stirnfräsen (A4, B4, C4, A5, B5, C5) ....683
3.9.2.6 Bezug der Orientierungsachsen (ORIWKS, ORIMKS) ....684
3.9.2.7 Programmierung der Orientierungsachsen (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2) ....686
3.9.2.8 Orientierungsprogrammierung entlang einer Kegelmantelfläche (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) ....688
3.9.2.9 Orientierungsvorgabe zweier Kontaktpunke (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=) ....691
3.9.3 Orientierungspolynome (PO[Winkel], PO[Koordinate]) ....693
3.9.4 Drehungen der Werkzeugorientierung (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) ....695
3.9.5 Bahnrelative Orientierungen ....697
3.9.5.1 Orientierungsarten relativ zur Bahn ....697
3.9.5.2 Bahnrelative Drehung der Werkzeugorientierung (ORIPATH, ORIPATHS, Drehwinkel) ....698
3.9.5.3 Bahnrelative Interpolation der Werkzeugdrehung (ORIROTC, THETA) ....700
3.9.5.4 Glättung des Orientierungsverlaufs (ORIPATHS A8=, B8=, C8=) ....702
3.9.6 Komprimierung der Orientierung (COMPON, COMPCURV, COMPCAD, COMPSURF) ....703
3.9.7 Komprimierung der Orientierung (COMPCAD, COMPSURF) ....706
3.9.8 Glättung des Orientierungsverlaufs ein-/ausschalten (ORISON, ORISOF) ....708
3.9.9 Kinematische Transformation ....709
3.9.9.1 Stirnseitentransformation einschalten (TRANSMIT) ....709
3.9.9.2 Zylindermanteltransformation einschalten (TRACYL) ....710
3.9.9.3 Schiefwinkeltransformation mit programmierbarem Winkel einschalten (TRAANG) ....713
3.9.9.4 Schräges Einstechen an Schleifmaschinen (G5, G7) ....714
3.9.10 Verkettete Transformation einschalten (TRACON) ....715
3.9.11 Kartesisches PTP-Fahren ....717
3.9.11.1 Kartesisches PTP-Fahren ein-/ausschalten (PTP, PTPG0, PTPWOC, CP) ....717
3.9.11.2 Stellung der Gelenke angeben (STAT) ....718
3.9.11.3 Vorzeichen der Achswinkel angeben (TU) ....722
3.9.11.4 Beispiel 1: PTP-Fahren eines 6-Achs-Roboters mit ROBX-Transformation ....725

3.9.11.5 Beispiel 2: PTP-Fahren bei generischer 5-Achs-Transformation ..... 726
3.9.11.6 Beispiel 3: PTPG0 und TRANSMIT ..... 727
3.9.12 Randbedingungen bei der Anwahl einer Transformation ..... 728
3.9.13 Transformation abwählen (TRAFOOF) ..... 730

3.10 Kinematische Ketten ..... 731
3.10.1 Löschen von Komponenten (DELOBJ) ..... 731
3.10.2 Indexermittlung per Namen (NAMETOINT) ..... 734

3.11 Kollisionsvermeidung mit kinematischen Ketten ..... 735
3.11.1 Prüfen auf Kollisionspaar (COLLPAIR) ..... 735
3.11.2 Neuberechnung des Maschinenmodells der Kollisionsvermeidung anfordern (PROTA) ..... 736
3.11.3 Schutzbereichszustand setzen (PROTS) ..... 737
3.11.4 Abstandsbestimmung zweier Schutzbereiche (PROTD) ..... 738

3.12 Transformationen mit kinematischen Ketten ..... 740
3.12.1 Transformation einschalten (TRAFOON) ..... 740
3.12.2 Orientierungstransformation nach Maschinenvermessung modifizieren (CORRTRAFO) ..... 741

3.13 Werkzeugkorrekturen ..... 749
3.13.1 Korrekturspeicher ..... 749
3.13.2 Additive Korrekturen ..... 751
3.13.2.1 Additive Korrekturen anwählen (DL) ..... 751
3.13.2.2 Verschleiß- und Einrichtewerte festlegen ($TC_SCPxy[t,d], $TC_ECPxy[t,d]) ..... 753
3.13.2.3 Additive Korrekturen löschen (DELDL) ..... 753
3.13.3 Werkzeugkorrektur - Sonderbehandlung ..... 754
3.13.3.1 Werkzeuglängen spiegeln ..... 756
3.13.3.2 Vorzeichenbewertung Verschleiß ..... 756
3.13.3.3 Koordinatensystem der aktiven Bearbeitung (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS) ..... 757
3.13.3.4 Werkzeuglänge und Ebenenwechsel ..... 760
3.13.4 Online-Werkzeugkorrektur ..... 761
3.13.4.1 Polynom-Funktion definieren (FCTDEF) ..... 761
3.13.4.2 Online-Werkzeugkorrektur schreiben, kontinuierlich (PUTFTOCF) ..... 763
3.13.4.3 Online-Werkzeugkorrektur schreiben, diskret (PUTFTOC) ..... 764
3.13.4.4 Online-Werkzeugkorrektur ein-/ausschalten (FTOCON/FTOCOF) ..... 764
3.13.5 3D-Werkzeugradiuskorrektur ..... 765
3.13.5.1 3D-Werkzeugradiuskorrektur für das 3D-Umfangsfräsen anwählen (CUT3DC, CUT3DCD, ISD) ..... 765
3.13.5.2 3D-Werkzeugradiuskorrektur für das 3D-Stirnfräsen anwählen (CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD) ..... 769
3.13.5.3 3D-Umfangsfräsen unter Berücksichtigung einer Begrenzungsfläche (CUT3DCC, CUT3DCCD) ..... 775
3.13.6 Werkzeugorientierung (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST) ..... 780
3.13.7 Freie D-Nummernvergabe, Schneidennummer ..... 786
3.13.7.1 Freie D-Nummernvergabe, Schneidennummer (Adresse CE) ..... 786
3.13.7.2 Freie D-Nummernvergabe: D-Nummern prüfen (CHKDNO) ..... 786
3.13.7.3 Freie D-Nummernvergabe: D-Nummern umbenennen (GETDNO, SETDNO) ..... 787
3.13.7.4 Freie D-Nummernvergabe: T-Nummer zur vorgegebenen D-Nummer ermitteln (GETACTTD) ..... 788
3.13.7.5 Freie D-Nummernvergabe: D-Nummern ungültig setzen (DZERO) ..... 788
3.13.8 Werkzeugträgerkinematik ..... 789
3.13.9 Werkzeuglängenkorrektur für orientierbare Werkzeugträger (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ) ..... 794

3.13.10 Orientierbare Werkzeugträger nach Maschinenvermessung modifizieren (CORRTC) ..... 796
3.13.11 Online-Werkzeuglängenkorrektur (TOFFON, TOFFOF) ..... 800
3.13.12 Modifikation der Korrekturdaten bei drehbaren Werkzeugen ..... 802
3.13.12.1 Orientierungen berechnen (ORISOLH) ..... 802
3.13.12.2 Modifikation der Korrekturdaten bei drehbaren Werkzeugen aktivieren (CUTMOD, CUTMODK) ..... 813
3.13.13 Mit Werkzeugumgebungen arbeiten ..... 821
3.13.13.1 Werkzeugumgebung speichern (TOOLENV) ..... 821
3.13.13.2 Werkzeugumgebung löschen (DELTOOLENV) ..... 824
3.13.13.3 T-, D- und DL-Nummer lesen (GETTENV) ..... 825
3.13.13.4 Informationen zu gespeicherten Werkzeugumgebungen lesen ($P_TOOLENVN, ($P_TOOLENV) ..... 826
3.13.13.5 Werkzeuglängen bzw. -längenkomponenten lesen (GETTCOR) ..... 826
3.13.13.6 Werkzeugkomponenten ändern (SETTCOR) ..... 833
3.13.14 Zuordnung der Werkzeuglängen L1, L2, L3 zu den Koordinatenachsen lesen (LENTOAX) ..... 845

3.14 Bahnverhalten ..... 849
3.14.1 Vorschubverlauf (FNORM, FLIN, FCUB, FPO) ..... 849
3.14.2 Beschleunigungsverhalten ..... 854
3.14.2.1 Beschleunigungsmodus (BRISK, BRISKA, SOFT, SOFTA, DRIVE, DRIVEA) ..... 854
3.14.2.2 Beeinflussung der Beschleunigung bei Folgeachsen (VELOLIMA, ACCLIMA, JERKLIMA) ..... 856
3.14.2.3 Aktivierung von Technologie-spezifischen Dynamikwerten (DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN, DYNFINISH, DYNPREC) ..... 858
3.14.3 Fahren mit Vorsteuerung (FFWON, FFWOF) ..... 859
3.14.4 Programmierbare Konturgenauigkeit (CPRECON, CPRECOF) ..... 860
3.14.5 Programmablauf mit Vorlaufspeicher (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE) ..... 862
3.14.6 Stop-Delay-Bereiche definieren (DELAYFSTON, DELAYFSTOF) ..... 864
3.14.7 Programmstelle für SERUPRO verhindern (IPTRLOCK, IPTRUNLOCK) ..... 866
3.14.8 Wiederanfahren an Kontur (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL) ..... 869
3.14.9 Beeinflussung der Bewegungsführung ..... 877
3.14.9.1 Maximalen Achsruck anpassen (JERKLIM) ..... 877
3.14.9.2 Maximale Achsgeschwindigkeit oder Spindeldrehzahl anpassen (VELOLIM) ..... 878
3.14.10 Kontur-/Orientierungtoleranz programmieren (CTOL, OTOL, ATOL) ..... 880
3.14.11 Satzwechselverhalten bei aktiver Kopplung (CPBC) ..... 884

3.15 Achsfunktionen ..... 886
3.15.1 Achstausch, Spindeltausch (RELEASE, GET, GETD) ..... 886
3.15.2 Achse einem anderen Kanal übergeben (AXTOCHAN) ..... 890
3.15.3 Achsfunktionen (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) ..... 891
3.15.4 Umschaltbare Geometrieachsen (GEOAX) ..... 893
3.15.5 Achscontainer (AXCTSWE, AXCTSWED, AXCTSWEC) ..... 898
3.15.6 Warten auf gültige Achsposition (WAITENC) ..... 900
3.15.7 Programmierbare Parametersatzumschaltung (SCPARA) ..... 901
3.15.8 Adaption ein-/ausschalten (CADAPTON, CADAPTOF) ..... 902
3.15.9 FIR-Ruckfilter an Dynamikmodus anpassen (CALCFIR) ..... 905

3.16 Achskopplungen ..... 908
3.16.1 Mitschleppen (TRAILON, TRAILOF) ..... 908
3.16.2 Kurventabellen (CTAB) ..... 912
3.16.2.1 Kurventabellen definieren (CTABDEF, CATBEND) ..... 912
3.16.2.2 Vorhandensein einer Kurventabelle prüfen (CTABEXISTS) ..... 919

3.16.2.3 Kurventabellen löschen (CTABDEL) ..... 919
3.16.2.4 Kurventabellen gegen Löschen und Überschreiben sperren (CTABLOCK, CTABUNLOCK) ..... 920
3.16.2.5 Kurventabellen: Tabelleneigenschaften ermitteln (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD) ..... 921
3.16.2.6 Kurventabellenwerte lesen (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) ..... 923
3.16.2.7 Kurventabellen: Ressourcennutzung prüfen (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) ..... 927
3.16.3 Axiale Leitwertkopplung (LEADON, LEADOF) ..... 929
3.16.4 Elektronisches Getriebe (EG) ..... 934
3.16.4.1 Elektronisches Getriebe definieren (EGDEF) ..... 934
3.16.4.2 Elektronisches Getriebe einschalten (EGON, EGONSYN, EGONSYNE) ..... 935
3.16.4.3 Elektronisches Getriebe ausschalten (EGOFS, EGOFC) ..... 938
3.16.4.4 Definition eines Elektronischen Getriebes löschen (EGDEL) ..... 939
3.16.4.5 Umdrehungsvorschub (G95) / Elektronisches Getriebe (FPR) ..... 939
3.16.5 Synchronspindel ..... 940
3.16.5.1 Synchronspindel: Programmierung (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC) ..... 940
3.16.6 Generische Kopplung (CP...) ..... 950
3.16.7 Tangentialsteuerung ..... 958
3.16.7.1 Kopplung definieren (TANG) ..... 958
3.16.7.2 Zwischensatzerzeugung einschalten (TLIFT) ..... 959
3.16.7.3 Kopplung einschalten (TANGON) ..... 960
3.16.7.4 Kopplung ausschalten (TANGOF) ..... 962
3.16.7.5 Kopplung löschen (TANGDEL) ..... 962
3.16.8 Master/Slave-Kopplung (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS) ..... 964

3.17 Synchronaktionen ..... 967
3.17.1 Definition einer Synchronaktion ..... 967

3.18 Pendeln ..... 968
3.18.1 Asynchrones Pendeln (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) ..... 968
3.18.2 Über Synchronaktionen gesteuertes Pendeln (OSCILL) ..... 973

3.19 Stanzen und Nibbeln ..... 980
3.19.1 Aktivierung/Deaktivierung ..... 980
3.19.1.1 Stanzen und Nibbeln ein-/ausschalten (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) ..... 980
3.19.2 Automatische Wegaufteilung ..... 985
3.19.2.1 Wegaufteilung bei Bahnachsen ..... 987
3.19.2.2 Wegaufteilung bei Einzelachsen ..... 989

3.20 Schleifen ..... 991
3.20.1 Schleifspezifische Werkzeugüberwachung ein-/ausschalten (TMON, TMOF) ..... 991

3.21 Erweitertes Stillsetzen und Rückziehen (ESR) ..... 992
3.21.1 NC-geführtes ESR ..... 993
3.21.1.1 NC-geführtes Rückziehen (POLF, POLFA, POLFMASK, POLFMLIN) ..... 993
3.21.1.2 NC-geführtes Stillsetzen ..... 997
3.21.2 Antriebsautarkes ESR ..... 997
3.21.2.1 Antriebsautarkes Stillsetzen projektieren (ESRS) ..... 997
3.21.2.2 Antriebsautarkes Rückziehen projektieren (ESRR) ..... 998

3.22 Programmlaufzeit / Werkstückzähler ... 1000
3.22.1 Programmlaufzeit ... 1000
3.22.2 Werkstückzähler ... 1003

3.23 Weitere Funktionen ... 1005
3.23.1 Maschinendaten wirksam setzen (NEWCONF) ... 1005
3.23.2 Vorhandenen NC-Sprachumfang prüfen (STRINGIS) ... 1006
3.23.3 Fenster aus dem Teileprogramm interaktiv aufrufen (MMC) ... 1009
3.23.4 Process DataShare - Ausgabe auf ein externes Gerät/Datei (EXTOPEN, WRITE, EXTCLOSE) ... 1014
3.23.5 Alarme (SETAL) ... 1018
3.23.6 Rohteil definieren (WORKPIECE) ... 1020
3.23.7 Sprachmodus umschalten (G290, G291) ... 1024

3.24 Eigene Abspanprogramme ... 1027
3.24.1 Unterstützende Funktionen für das Abspanen ... 1027
3.24.2 Konturtabelle erstellen (CONTPRON) ... 1027
3.24.3 Codierte Konturtabelle erstellen (CONTDCON) ... 1033
3.24.4 Schnittpunkt zwischen zwei Konturelementen ermitteln (INTERSEC) ... 1036
3.24.5 Konturelemente einer Tabelle satzweise abfahren (EXECTAB) ... 1038
3.24.6 Kreisdaten berechnen (CALCDAT) ... 1038
3.24.7 Konturaufbereitung ausschalten (EXECUTE) ... 1040

3.25 Zyklen extern programmieren ... 1042
3.25.1 Technologische Zyklen ... 1042
3.25.1.1 Einleitung ... 1042
3.25.1.2 Technologie-spezifische Übersicht ... 1043
3.25.1.3 HOLES1 - Positionsmuster Reihe ... 1045
3.25.1.4 HOLES2 - Positionsmuster Kreis oder Teilkreis ... 1045
3.25.1.5 POCKET3 - Rechtecktasche ... 1047
3.25.1.6 POCKET4 - Kreistasche ... 1050
3.25.1.7 SLOT1 - Längsnut ... 1052
3.25.1.8 SLOT2 - Kreisnut ... 1055
3.25.1.9 LONGHOLE - Langloch ... 1057
3.25.1.10 CYCLE60 - Gravur ... 1059
3.25.1.11 CYCLE61 - Planfräsen ... 1062
3.25.1.12 CYCLE62 - Konturaufruf ... 1064
3.25.1.13 CYCLE63 - Konturtasche fräsen / Konturtasche Restmaterial / Konturzapfen fräsen / Konturzapfen Restmaterial ... 1064
3.25.1.14 CYCLE64 - Konturtasche vorbohren ... 1067
3.25.1.15 CYCLE70 - Gewindefräsen ... 1069
3.25.1.16 CYCLE72 - Bahnfräsen ... 1070
3.25.1.17 CYCLE76 - Rechteckzapfen ... 1073
3.25.1.18 CYCLE77 - Kreiszapfen ... 1076
3.25.1.19 CYCLE78 - Bohrgewinde fräsen ... 1078
3.25.1.20 CYCLE79 - Mehrkant ... 1080
3.25.1.21 CYCLE81 - Bohren, Zentrieren ... 1082
3.25.1.22 CYCLE82 - Bohren, Plansenken ... 1083
3.25.1.23 CYCLE83 - Tieflochbohren 1 ... 1086
3.25.1.24 CYCLE84 - Gewindebohren ohne Ausgleichsfutter ... 1089
3.25.1.25 CYCLE85 - Reiben ... 1092
3.25.1.26 CYCLE86 - Ausdrehen ... 1093
3.25.1.27 CYCLE92 - Abstich ... 1094

3.25.1.28 CYCLE95 - Konturabspanen ..... 1096
3.25.1.29 CYCLE98 - Gewindekette ..... 1098
3.25.1.30 CYCLE99 - Gewindedrehen ..... 1102
3.25.1.31 CYCLE435 - Abrichterkoordinatensystem setzen ..... 1107
3.25.1.32 CYCLE495 - Profilieren ..... 1107
3.25.1.33 CYCLE782 - An Beladung anpassen ..... 1109
3.25.1.34 CYCLE800 - Schwenken Ebene / Schwenken Werkzeug / Ausrichten Werkzeug ..... 1112
3.25.1.35 CYCLE801 - Positionsmuster Gitter oder Rahmen ..... 1115
3.25.1.36 CYCLE802 - Beliebige Positionen ..... 1117
3.25.1.37 CYCLE830 - Tieflochbohren 2 ..... 1120
3.25.1.38 CYCLE832 - High Speed Settings ..... 1126
3.25.1.39 CYCLE840 - Gewindebohren mit Ausgleichsfutter ..... 1130
3.25.1.40 CYCLE899 - Offene Nut ..... 1132
3.25.1.41 CYCLE930 - Einstich ..... 1135
3.25.1.42 CYCLE940 - Freistich Form E und F / Freistich Gewinde ..... 1137
3.25.1.43 CYCLE951 - Abspanen ..... 1140
3.25.1.44 CYCLE952 - Abspanen / Abspanen Rest / Stechen / Stechen Rest / Stechdrehen / Stechdrehen Rest ..... 1143
3.25.1.45 CYCLE4071 - Längsschleifen mit Zustellung am Umkehrpunkt ..... 1149
3.25.1.46 CYCLE4072 - Längsschleifen mit Zustellung am Umkehrpunkt und Abbruchsignal ..... 1151
3.25.1.47 CYCLE4073 - Längschleifen mit kontinuierlicher Zustellung ..... 1155
3.25.1.48 CYCLE4074 - Längschleifen mit kontinuierlicher Zustellung und Abbruchsignal ..... 1156
3.25.1.49 CYCLE4075 - Flachschleifen mit Zustellung am Umkehrpunkt ..... 1159
3.25.1.50 CYCLE4077 - Flachschleifen mit Zustellung am Umkehrpunkt und Abbruchsignal ..... 1162
3.25.1.51 CYCLE4078 - Flachschleifen mit kontinuierlicher Zustellung ..... 1166
3.25.1.52 CYCLE4079 - Flachschleifen mit intermittierender Zustellung ..... 1168
3.25.1.53 GROUP_BEGIN - Anfang Programmblock ..... 1171
3.25.1.54 GROUP_END - Ende Programmblock ..... 1171
3.25.1.55 GROUP_ADDEND - Ende Einfahrzusatz ..... 1172
3.25.1.56 Randbedingungen ..... 1172
3.25.2 Messzyklen ..... 1173
3.25.3 Übersicht Messzyklenparameter ..... 1174
3.25.3.1 Messzyklenparameter CYCLE973 ..... 1174
3.25.3.2 Messzyklenparameter CYCLE974 ..... 1176
3.25.3.3 Messzyklenparameter CYCLE994 ..... 1179
3.25.3.4 Messzyklenparameter CYCLE976 ..... 1182
3.25.3.5 Messzyklenparameter CYCLE978 ..... 1184
3.25.3.6 Messzyklenparameter CYCLE998 ..... 1187
3.25.3.7 Messzyklenparameter CYCLE977 ..... 1190
3.25.3.8 Messzyklenparameter CYCLE961 ..... 1193
3.25.3.9 Messzyklenparameter CYCLE979 ..... 1196
3.25.3.10 Messzyklenparameter CYCLE997 ..... 1198
3.25.3.11 Messzyklenparameter CYCLE995 ..... 1201
3.25.3.12 Messzyklenparameter CYCLE996 ..... 1202
3.25.3.13 Messzyklenparameter CYCLE9960 ..... 1205
3.25.3.14 Messzyklenparameter CYCLE982 ..... 1207
3.25.3.15 Messzyklenparameter CYCLE971 ..... 1209
3.25.3.16 Messzyklenparameter CYCLE150 ..... 1212

**4 Tabellen ..... 1215**

4.1 Anweisungen ..... 1215
4.2 Adressen ..... 1255

4.2.1 Adressbuchstaben ... 1255
4.2.2 Feste Adressen ... 1256
4.2.3 Einstellbare Adressen ... 1260

4.3 G-Befehle ... 1267
4.3.1 G-Befehle ... 1267
4.3.2 G-Gruppe 1: Modal wirksame Bewegungsbefehle ... 1267
4.3.3 G-Gruppe 2: Satzweise wirksame Bewegungen, Verweilzeit ... 1268
4.3.4 G-Gruppe 3: Programmierbarer Frame, Arbeitsfeldbegrenzung und Polprogrammierung ... 1268
4.3.5 G-Gruppe 4: FIFO ... 1269
4.3.6 G-Gruppe 6: Ebenenanwahl ... 1269
4.3.7 G-Gruppe 7: Werkzeugradiuskorrektur ... 1270
4.3.8 G-Gruppe 8: Einstellbare Nullpunktverschiebung ... 1270
4.3.9 G-Gruppe 9: Frame- und Werkzeugkorrektur-Unterdrückung ... 1270
4.3.10 G-Gruppe 10: Genauhalt - Bahnsteuerbetrieb ... 1271
4.3.11 G-Gruppe 11: Genauhalt satzweise ... 1271
4.3.12 G-Gruppe 12: Satzwechselkriterien bei Genauhalt (G60/G9) ... 1271
4.3.13 G-Gruppe 13: Werkstückvermaßung Inch/metrisch ... 1271
4.3.14 G-Gruppe 14: Werkstückvermaßung absolut/inkremental ... 1272
4.3.15 G-Gruppe 15: Vorschubtyp ... 1272
4.3.16 G-Gruppe 16: Vorschub-Override an Innen- und Außenkrümmung ... 1273
4.3.17 G-Gruppe 17: An-/Abfahrverhalten Werkzeugkorrektur ... 1273
4.3.18 G-Gruppe 18: Eckenverhalten Werkzeugkorrektur ... 1273
4.3.19 G-Gruppe 19: Kurvenübergang bei Spline-Beginn ... 1273
4.3.20 G-Gruppe 20: Kurvenübergang bei Spline-Ende ... 1274
4.3.21 G-Gruppe 21: Beschleunigungsprofil ... 1274
4.3.22 G-Gruppe 22: Werkzeugkorrekturtyp ... 1274
4.3.23 G-Gruppe 23: Kollisionsüberwachung an Innenkonturen ... 1275
4.3.24 G-Gruppe 24: Vorsteuerung ... 1275
4.3.25 G-Gruppe 25: Bezug Werkzeugorientierung ... 1275
4.3.26 G-Gruppe 26: Wiederanfahrmodus für REPOS (modal wirksam) ... 1276
4.3.27 G-Gruppe 27: Werkzeugkorrektur bei Orientierungsänderung an Außenecken ... 1276
4.3.28 G-Gruppe 28: Arbeitsfeldbegrenzung ... 1276
4.3.29 G-Gruppe 29: Radius-/Durchmesserprogrammierung ... 1276
4.3.30 G-Gruppe 30: NC-Satz-Kompression ... 1277
4.3.31 G-Gruppe 31: OEM-G-Befehle ... 1277
4.3.32 G-Gruppe 32: OEM-G-Befehle ... 1278
4.3.33 G-Gruppe 33: Einstellbare Werkzeugfeinkorrektur ... 1278
4.3.34 G-Gruppe 34: Glättung Werkzeugorientierung ... 1278
4.3.35 G-Gruppe 35: Stanzen und Nibbeln ... 1279
4.3.36 G-Gruppe 36: Stanzen mit Verzögerung ... 1279
4.3.37 G-Gruppe 37: Vorschubprofil ... 1279
4.3.38 G-Gruppe 38: Zuordnung schnelle Ein-/Ausgänge für Stanzen/Nibbeln ... 1279
4.3.39 G-Gruppe 39: Programmierbare Konturgenauigkeit ... 1280
4.3.40 G-Gruppe 40: Werkzeugradiuskorrektur konstant ... 1280
4.3.41 G-Gruppe 41: Gewindeschneiden unterbrechbar ... 1280
4.3.42 G-Gruppe 42: Werkzeugträger ... 1280
4.3.43 G-Gruppe 43: Anfahrrichtung WAB ... 1281
4.3.44 G-Gruppe 44: Wegaufteilung WAB ... 1281
4.3.45 G-Gruppe 45: Bahnbezug der FGROUP-Achsen ... 1281
4.3.46 G-Gruppe 46: Ebenenanwahl für Schnellabheben ... 1281
4.3.47 G-Gruppe 47: Modus-Umschaltung für externen NC-Code ... 1282
4.3.48 G-Gruppe 48: An- / Abfahrverhalten bei Werkzeugradiuskorrektur ... 1282

4.3.49 G-Gruppe 49: Punkt-zu-Punkt-Bewegung ..... 1282
4.3.50 G-Gruppe 50: Orientierungsprogrammierung ..... 1282
4.3.51 G-Gruppe 51: Interpolationsart Orientierungsprogrammierung ..... 1283
4.3.52 G-Gruppe 52: Werkstückbezogene Frame-Drehung ..... 1283
4.3.53 G-Gruppe 53: Werkzeugbezogene Frame-Drehung ..... 1284
4.3.54 G-Gruppe 54: Vektordrehung bei Polynomprogrammierung ..... 1284
4.3.55 G-Gruppe 55: Eilgangbewegung mit/ohne Linear-Interpolation ..... 1284
4.3.56 G-Gruppe 56: Einrechnung des Werkzeugverschleißes ..... 1285
4.3.57 G-Gruppe 57: Eckenverzögerung ..... 1285
4.3.58 G-Gruppe 59: Dynamikmodus für Bahninterpolation ..... 1285
4.3.59 G-Gruppe 60: Arbeitsfeldbegrenzung ..... 1286
4.3.60 G-Gruppe 61: Glättung Werkzeugorientierung ..... 1286
4.3.61 G-Gruppe 62: Wiederanfahrmodus für REPOS (satzweise wirksam) ..... 1286
4.3.62 G-Gruppe 64: Schleifframes ..... 1286
4.4 Vordefinierte Prozeduren ..... 1288
4.5 Vordefinierte Prozeduren in Synchronaktionen ..... 1310
4.6 Vordefinierte Funktionen ..... 1312
4.7 Aktuelle Sprache im HMI ..... 1326
**A Anhang ..... 1327**
A.1 Liste der Abkürzungen ..... 1327
A.2 Dokumentationsübersicht ..... 1336
**Index ..... 1337**

# Grundlegende Sicherheitshinweise 1

## 1.1 Allgemeine Sicherheitshinweise

**⚠ WARNUNG**

**Lebensgefahr bei Nichtbeachtung von Sicherheitshinweisen und Restrisiken**

Bei Nichtbeachtung der Sicherheitshinweise und Restrisiken in der zugehörigen Hardware-Dokumentation können Unfälle mit schweren Verletzungen oder Tod auftreten.

- Halten Sie die Sicherheitshinweise der Hardware-Dokumentation ein.
- Berücksichtigen Sie bei der Risikobeurteilung die Restrisiken.

**⚠ WARNUNG**

**Fehlfunktionen der Maschine infolge fehlerhafter oder veränderter Parametrierung**

Durch fehlerhafte oder veränderte Parametrierung können Fehlfunktionen an Maschinen auftreten, die zu Körperverletzungen oder Tod führen können.

- Schützen Sie die Parametrierung vor unbefugtem Zugriff.
- Beherrschen Sie mögliche Fehlfunktionen durch geeignete Maßnahmen, z. B. NOT-HALT oder NOT-AUS.

## 1.2 Gewährleistung und Haftung für Applikationsbeispiele

Applikationsbeispiele sind unverbindlich und erheben keinen Anspruch auf Vollständigkeit hinsichtlich Konfiguration und Ausstattung sowie jeglicher Eventualitäten. Applikationsbeispiele stellen keine kundenspezifischen Lösungen dar, sondern sollen lediglich Hilfestellung bieten bei typischen Aufgabenstellungen.

Als Anwender sind Sie für den sachgemäßen Betrieb der beschriebenen Produkte selbst verantwortlich. Applikationsbeispiele entheben Sie nicht der Verpflichtung zu sicherem Umgang bei Anwendung, Installation, Betrieb und Wartung.

# 1.3 Industrial Security

**Hinweis**

**Industrial Security**

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen.

Um Anlagen, Systeme, Maschinen und Netzwerke gegen Cyber-Bedrohungen zu sichern, ist es erforderlich, ein ganzheitliches Industrial Security-Konzept zu implementieren (und kontinuierlich aufrechtzuerhalten), das dem aktuellen Stand der Technik entspricht. Die Produkte und Lösungen von Siemens formen einen Bestandteil eines solchen Konzepts.

Die Kunden sind dafür verantwortlich, unbefugten Zugriff auf ihre Anlagen, Systeme, Maschinen und Netzwerke zu verhindern. Diese Systeme, Maschinen und Komponenten sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn und soweit dies notwendig ist und nur wenn entsprechende Schutzmaßnahmen (z. B. Nutzung von Firewalls und/oder Netzwerksegmentierung) ergriffen wurden.

Weiterführende Informationen zu möglichen Schutzmaßnahmen im Bereich Industrial Security finden Sie unter:

Industrial Security (https://www.siemens.com/industrialsecurity)

Die Produkte und Lösungen von Siemens werden ständig weiterentwickelt, um sie noch sicherer zu machen. Siemens empfiehlt ausdrücklich, Produkt-Updates anzuwenden, sobald sie zur Verfügung stehen und immer nur die aktuellen Produktversionen zu verwenden. Die Verwendung veralteter oder nicht mehr unterstützter Versionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Produkt-Updates informiert zu sein, abonnieren Sie den Siemens Industrial Security RSS Feed unter:

Industrial Security (https://www.siemens.com/industrialsecurity)

Weitere Informationen finden Sie im Internet:

Projektierungshandbuch Industrial Security (https://support.industry.siemens.com/cs/ww/de/view/108862708)

**⚠ WARNUNG**

**Unsichere Betriebszustände durch Manipulation der Software**

Manipulationen der Software, z. B. Viren, Trojaner oder Würmer, können unsichere Betriebszustände in Ihrer Anlage verursachen, die zu Tod, schwerer Körperverletzung und zu Sachschäden führen können.

- Halten Sie die Software aktuell.
- Integrieren Sie die Automatisierungs- und Antriebskomponenten in ein ganzheitliches Industrial Security-Konzept der Anlage oder Maschine nach dem aktuellen Stand der Technik.
- Berücksichtigen Sie bei Ihrem ganzheitlichen Industrial Security-Konzept alle eingesetzten Produkte.
- Schützen Sie die Dateien in Wechselspeichermedien vor Schadsoftware durch entsprechende Schutzmaßnahmen, z. B. Virenscanner.
- Prüfen Sie beim Abschluss der Inbetriebnahme alle security-relevanten Einstellungen.
- Schützen Sie den Antrieb vor unberechtigten Änderungen, indem Sie die Umrichterfunktion "Know-How-Schutz" aktivieren.

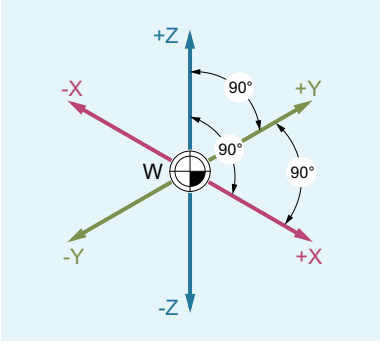
# Grundlagen 2

## 2.1 Geometrische Grundlagen

### 2.1.1 Werkstück-Positionen

#### 2.1.1.1 Bezugssystem der Positionsangaben

Damit die Maschine bzw. Steuerung mit den im NC-Programm angegebenen Positionen arbeiten kann, müssen sich diese Positionsangaben auf ein Bezugssystem beziehen, das auf die Bewegungsrichtungen der Maschinenachsen übertragen werden kann. Dafür werden bei Werkzeugmaschinen für das Werkstück-Koordinatensystem kartesische, d.h. rechtsdrehende, rechtwinkelige, Koordinatensysteme nach DIN 66217 benutzt.



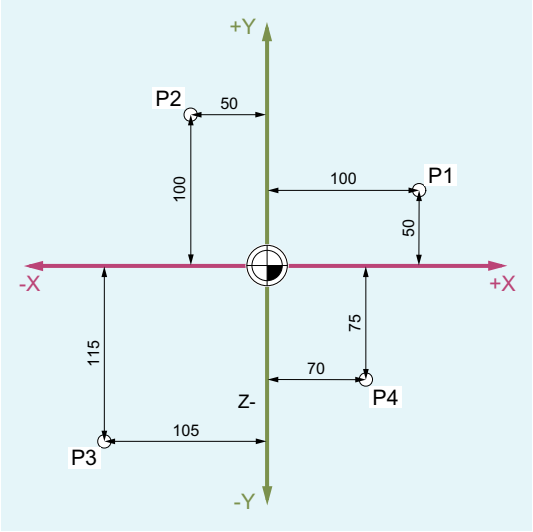

Der Werkstück-Nullpunkt (W) ist Der Ursprung des Werkstück-Koordinatensystems.

#### 2.1.1.2 Kartesische Koordinaten

Die Achsen im Koordinatensystem sind vermaßt. Dadurch ist es möglich, jeden Punkt im Koordinatensystem und damit jede Werkstück-Position durch die Richtung (X, Y und Z) und drei Zahlenwerte eindeutig zu beschreiben. Der Werkstück-Nullpunkt hat immer die Koordinaten X0, Y0 und Z0.

## Positionsangaben in Form kartesischer Koordinaten

Der Einfachheit halber betrachten wir bei dem folgenden Beispiel nur eine Ebene des Koordinatensystems, die X/Y-Ebene:

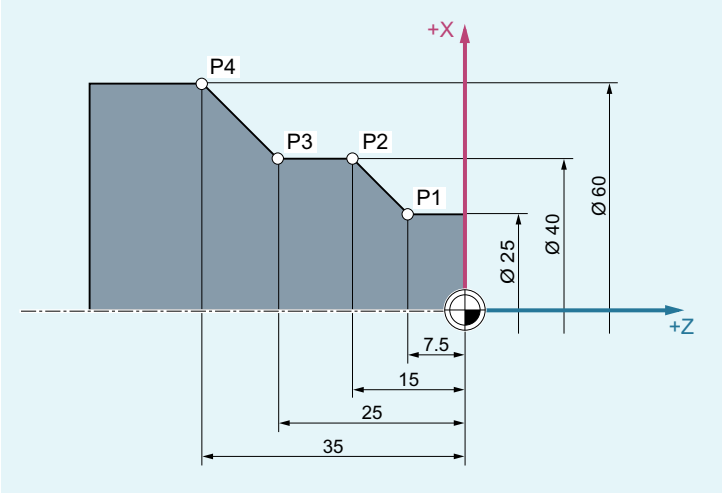



Die Punkte P1 bis P4 besitzen folgende Koordinaten:

| Position | Koordinaten |
|---|---|
| P1 | X100 Y50 |
| P2 | X-50 Y100 |
| P3 | X-105 Y-115 |
| P4 | X70 Y-75 |

## Beispiel: Werkstück-Positionen beim Drehen

Bei Drehmaschinen genügt eine Ebene, um die Kontur zu beschreiben:

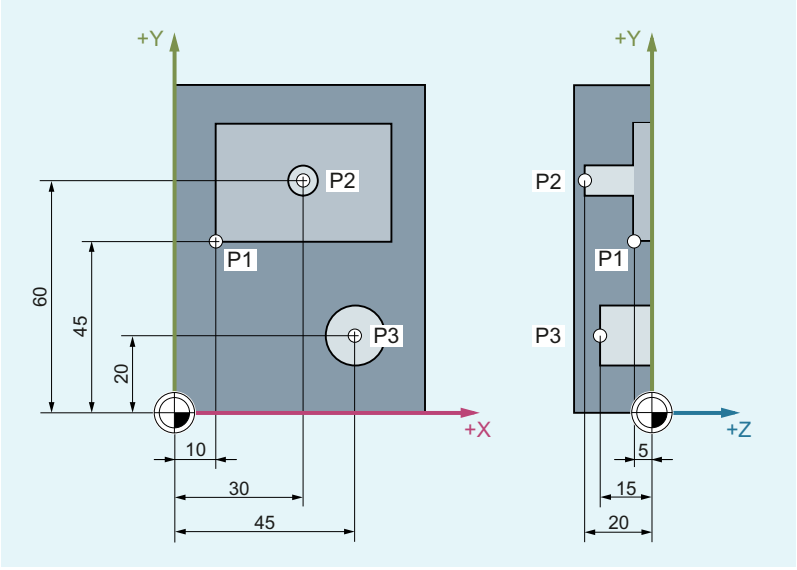


Die Punkte P1 bis P4 besitzen folgende Koordinaten:

| Position | Koordinaten |
|---|---|
| P1 | X25 Z-7.5 |
| P2 | X40 Z-15 |
| P3 | X40 Z-25 |
| P4 | X60 Z-35 |

### Beispiel: Werkstück-Positionen beim Fräsen

Bei Fräsbearbeitungen muss auch die Zustelltiefe beschrieben werden, d. h., es muss auch der dritten Koordinate (in diesem Fall Z) ein Zahlenwert zugeordnet werden.




Die Punkte P1 bis P3 besitzen folgende Koordinaten:

| Position | Koordinaten |
|---|---|
| P1 | X10 Y45 Z-5 |
| P2 | X30 Y60 Z-20 |
| P3 | X45 Y20 Z-15 |

## 2.1.1.3 Polarkoordinaten

Anstelle von kartesischen Koordinaten können zur Beschreibung von Werkstück-Positionen auch Polarkoordinaten verwendet werden. Das ist dann sinnvoll, wenn ein Werkstück oder ein Teil eines Werkstücks mit Radius und Winkel vermaßt ist. Der Punkt, von dem die Vermaßung ausgeht, heißt "Pol".

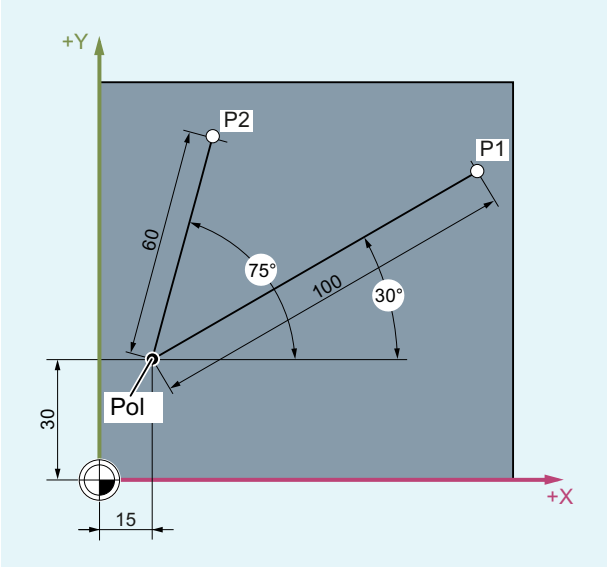
### Positionsangaben in Form von Polarkoordinaten

Polarkoordinaten setzen sich zusammen aus dem **Polarradius** und dem **Polarwinkel**.

Der Polarradius ist der Abstand zwischen dem Pol und der Position.

Der Polarwinkel ist der Winkel zwischen dem Polarradius und der waagrechten Achse der Arbeitsebene. Negative Polarwinkel verlaufen im Uhrzeigersinn, positive im Gegenuhrzeigersinn.

### Beispiel




Die Punkte P1 und P2 können, bezogen auf den Pol, wie folgt beschrieben werden:

| Position | Polarkoordinaten |
|---|---|
| P1 | RP=100 AP=30 |
| P2 | RP=60 AP=75 |
| RP: Polarradius<br>AP: Polarwinkel | |

## 2.1.1.4 Absolutmaß

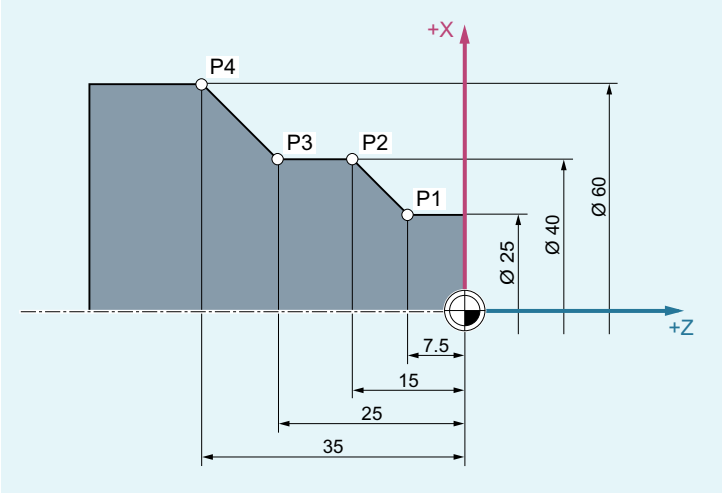
### Positionsangaben im Absolutmaß

Beim Absolutmaß beziehen sich alle Positionsangaben immer auf den gerade gültigen Nullpunkt.

Im Hinblick auf die Werkzeugbewegung bedeutet das:

**Die Absolutmaßangabe beschreibt die Position, auf die das Werkzeug fahren soll.**

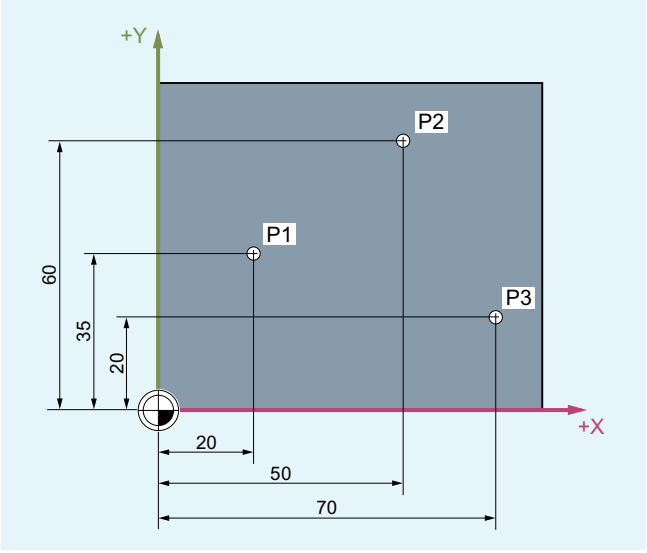
## Beispiel: Drehen




Im Absolutmaß ergeben sich für die Punkte P1 bis P4 folgende Positionsangaben:

| Position | Positionsangabe im Absolutmaß |
|---|---|
| P1 | X25 Z-7,5 |
| P2 | X40 Z-15 |
| P3 | X40 Z-25 |
| P4 | X60 Z-35 |

## Beispiel: Fräsen



Im Absolutmaß ergeben sich für die Punkte P1 bis P3 folgende Positionsangaben:

| Position | Positionsangabe im Absolutmaß |
|---|---|
| P1 | X20 Y35 |
| P2 | X50 Y60 |
| P3 | X70 Y20 |

## 2.1.1.5 Kettenmaß

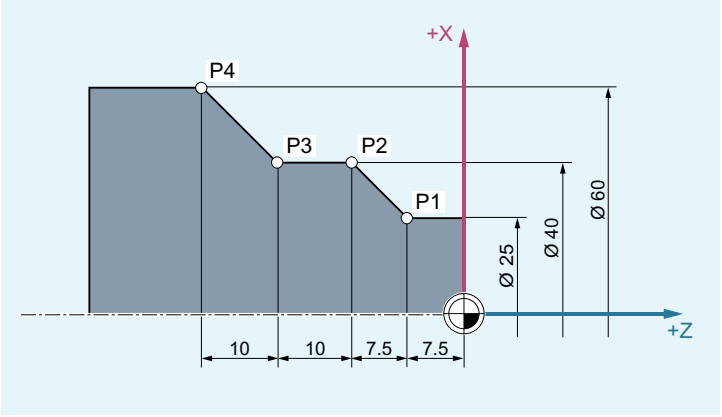
### Positionsangaben im Kettenmaß (Inkrementalmaß)

In Fertigungszeichnungen beziehen sich die Maße häufig nicht auf den Nullpunkt, sondern auf einen anderen Werkstückpunkt. Um solche Maße nicht umrechnen zu müssen, gibt es die Möglichkeit der Ketten- oder Inkrementalmaßangabe. Bei dieser Art der Maßangabe bezieht sich eine Positionsangabe auf den jeweils vorherigen Punkt.

Im Hinblick auf die Werkzeugbewegung bedeutet das:

**Die Kettenmaßangabe beschreibt, um wie viel das Werkzeug verfahren soll.**

### Beispiel: Drehen




Im Kettenmaß ergeben sich für die Punkte P2 bis P4 folgende Positionsangaben:

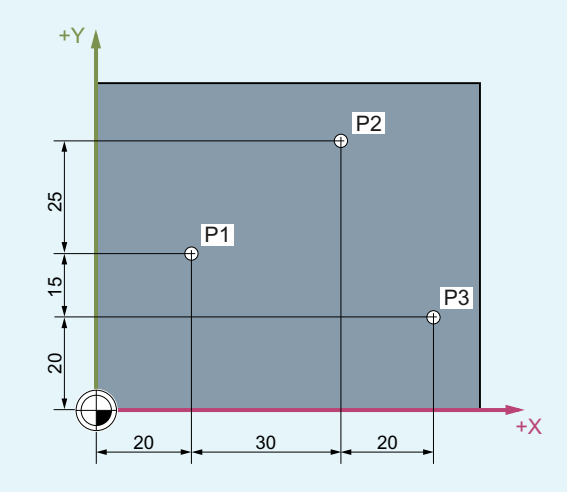
| Position | Positionsangabe im Kettenmaß | Die Angabe bezieht sich auf: |
|---|---|---|
| P2 | X15 Z-7,5 | P1 |
| P3 | Z-10 | P2 |
| P4 | X20 Z-10 | P3 |

**Hinweis**

Bei anstehendem DIAMOF oder DIAM90 (Seite 165) wird der Sollweg bei Kettenmaßangabe (G91) als Radiusmaß programmiert.

### Beispiel: Fräsen

Die Positionsangaben für die Punkte P1 bis P3 im Kettenmaß lauten:




Im Kettenmaß ergeben sich für die Punkte P1 bis P3 folgende Positionsangaben:

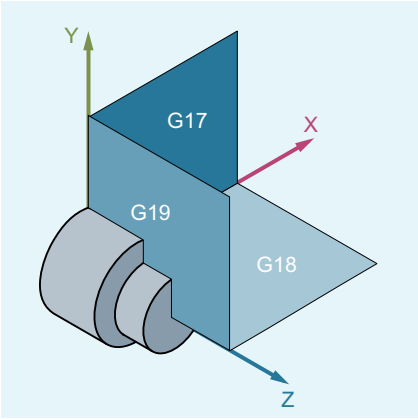
| Position | Positionsangabe im Kettenmaß | Die Angabe bezieht sich auf: |
|---|---|---|
| P1 | X20 Y35 | Nullpunkt |
| P2 | X30 Y20 | P1 |
| P3 | X20 Y-35 | P2 |

## 2.1.2 Arbeitsebenen

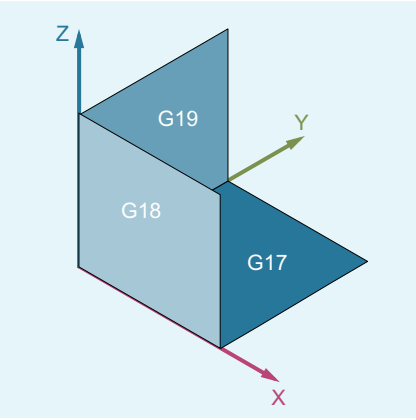
Ein NC-Programm benötigt die Information, in welcher Ebene die Bearbeitung stattfindet. Nur dann kann die Steuerung z. B. Werkzeugkorrekturwerte korrekt einrechnen. Des Weiteren wird die Angabe der Arbeitsebene für bestimmte Arten der Kreisprogrammierung und bei Polarkoordinaten benötigt.

Die Arbeitsebene wird im zu Grunde liegenden kartesischen Werkstück-Koordinatensystem durch zwei Koordinatenachsen festgelegt. Die dritte Koordinatenachse steht senkrecht auf dieser Arbeitsebene und bestimmt die Zustellrichtung des Werkzeugs (z. B. für 2D-Bearbeitung).

### Arbeitsebenen beim Drehen / Fräsen




Arbeitsebenen beim Drehen




Arbeitsebenen beim Fräsen

### Aktivieren einer Arbeitsebene

Die Arbeitsebenen werden im NC-Programm mit den G-Befehlen G17, G18 und G19 aktiviert. Der Zusammenhang ist wie folgt definiert:

| G-Befehl | Arbeitsebene | Abszisse | Ordinate | Applikate ≙ Zustellrichtung |
|---|---|---|---|---|
| G17 | X/Y | X | Y | Z |
| G18 | Z/X | Z | X | Y |
| G19 | Y/Z | Y | Z | X |

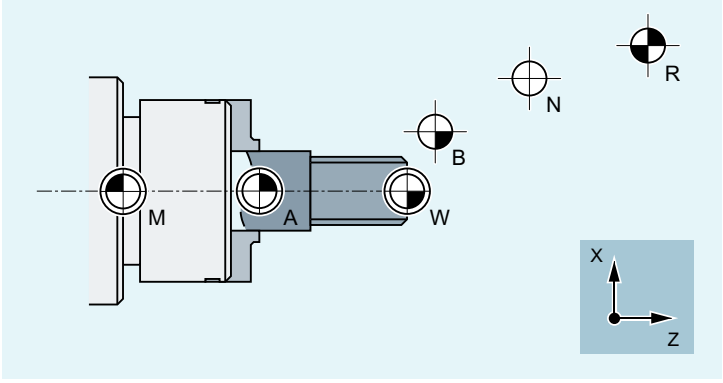
## 2.1.3 Nullpunkte und Bezugspunkte

An einer NC-Maschine sind verschiedene Nullpunkte und Bezugspunkte definiert:

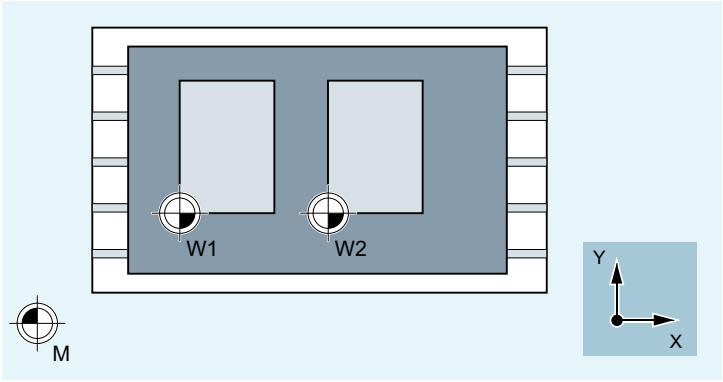
| Nullpunkte | | |
|---|---|---|
| | **M** | Maschinen-Nullpunkt<br>Mit dem Maschinen-Nullpunkt wird das Maschinen-Koordinatensystem (MKS) festgelegt. Auf den Maschinennullpunkt beziehen sich alle anderen Bezugspunkte. |
| | **W** | Werkstück-Nullpunkt = Programm-Nullpunkt<br>Der Werkstücknullpunkt legt das Werkstück-Koordinatensystem in Bezug auf den Maschinennullpunkt fest. |
| | **A** | Anschlagpunkt<br>Kann mit dem Werkstück-Nullpunkt zusammenfallen (nur bei Drehmaschinen). |

| Bezugspunkte | | |
|---|---|---|
| | **R** | Referenzpunkt<br>Durch Nocken und Messsystem festgelegte Position. Der Abstand zum Maschinen-Nullpunkt **M** muss bekannt sein, so dass die Achsposition an dieser Stelle exakt auf diesen Wert gesetzt werden kann. |
| | **B** | Startpunkt<br>Per Programm festlegbar. Hier beginnt das 1. Werkzeug der Bearbeitung. |
| | **T** | Werkzeugträgerbezugspunkt<br>Befindet sich an der Werkzeughalteraufnahme. Durch Eingabe der Werkzeuglängen berechnet die Steuerung den Abstand der Werkzeugspitze vom Werkzeugträgerbezugspunkt. |
| | **N** | Werkzeugwechselpunkt |

## Null- und Bezugspunkte beim Drehen




## Nullpunkte beim Fräsen



## 2.1.4 Koordinatensysteme

Folgende Koordinatensysteme werden unterschieden:

- Maschinen-Koordinaten-System (MKS) (Seite 36) mit dem Maschinen-Nullpunkt **M**
- Basis-Koordinaten-System (BKS) (Seite 38)
- Basis-Nullpunkt-System (BNS) (Seite 39)
- Einstellbares Nullpunkt-System (ENS) (Seite 40)
- Werkstück-Koordinaten-System (WKS) (Seite 41) mit dem Werkstück-Nullpunkt **W**

### 2.1.4.1 Maschinen-Koordinatensystem (MKS)

Das Maschinen-Koordinatensystem wird aus allen physikalisch vorhandenen Maschinenachsen gebildet.

Im Maschinen-Koordinatensystem sind Referenzpunkte, Werkzeug- und Palettenwechselpunkte (Maschinenfestpunkte) definiert.

Wenn direkt im Maschinen-Koordinatensystem programmiert wird (bei einigen G-Befehlen möglich), so werden die physikalischen Achsen der Maschine direkt angesprochen. Eine eventuell vorhandene Werkstückaufspannung wird dabei nicht berücksichtigt.

**Hinweis**

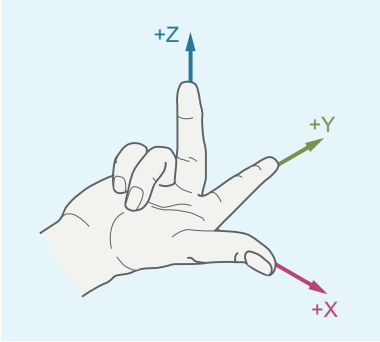
Falls es verschiedene Maschinen-Koordinatensysteme gibt (z. B. 5-Achs-Transformation), dann wird durch interne Transformation die Maschinenkinematik auf das Koordinatensystem abgebildet, in dem programmiert wird.

#### Drei-Finger-Regel

Wie das Koordinatensystem relativ zur Maschine liegt, ist abhängig vom Maschinentyp. Die Achsrichtungen folgen der sogenannten "Drei-Finger-Regel" der **rechten** Hand (nach DIN 66217).

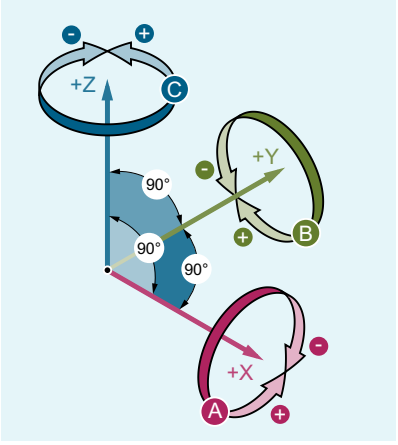
Steht man vor der Maschine so zeigt der Mittelfinger der rechten Hand gegen die Zustellrichtung der Hauptspindel. Dann bezeichnet:

- der Daumen die Richtung +X
- der Zeigefinger die Richtung +Y
- der Mittelfinger die Richtung +Z




Drehbewegungen um die Koordinatenachsen X, Y und Z werden mit A, B und C bezeichnet. Der Drehsinn der Drehung ergibt sich aus der Richtung der Drehbewegung bei Blick in die positive Richtung der Koordinatenachse:

| Richtung der Drehbewegung | Drehsinn |
|---|---|
| im Uhrzeigersinn | positiv |
| gegen den Uhrzeigersinn | negativ |

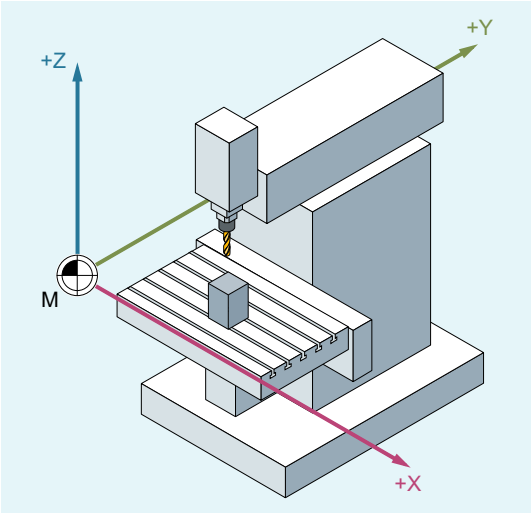



X, Y, Z Senkrecht aufeinanderstehende Koordinatenachsen

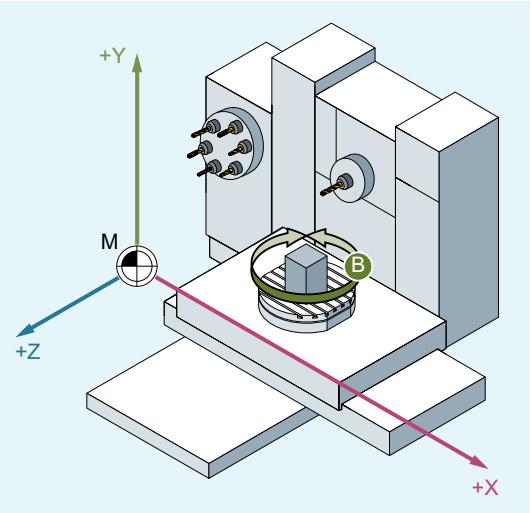
A, B, C Rundachsen, um X, Y, Z drehend

### Lage des Koordinatensystems bei unterschiedlichen Maschinentypen

Die Lage des Koordinatensystems, die sich aus der "Drei-Finger-Regel" ergibt, kann bei unterschiedlichen Maschinentypen unterschiedlich ausgerichtet sein, wie die beiden folgenden Beispiele zeigen:




Vertikale 3-Achsen-Fräsmaschine




Horizontale 4-Achsen-Fräsmaschine

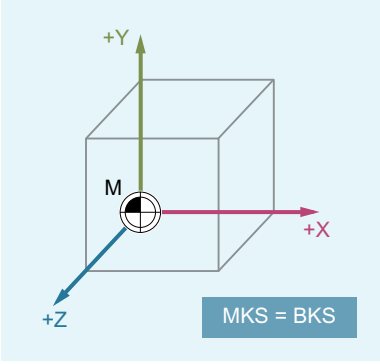
## 2.1.4.2 Basis-Koordinatensystem (BKS)

Das Basiskoordinatensystem (BKS) besteht aus drei rechtwinklig angeordneten Achsen (Geometrieachsen), sowie aus weiteren Achsen (Zusatzachsen) ohne geometrischen Zusammenhang.

### WZ-Maschinen ohne kinematische Transformation

Das BKS und das MKS fallen immer dann zusammen, wenn das BKS ohne kinematische Transformation (z. B. 5-Achstransformation, TRANSMIT / TRACYL / TRAANG) auf das MKS abgebildet werden kann.

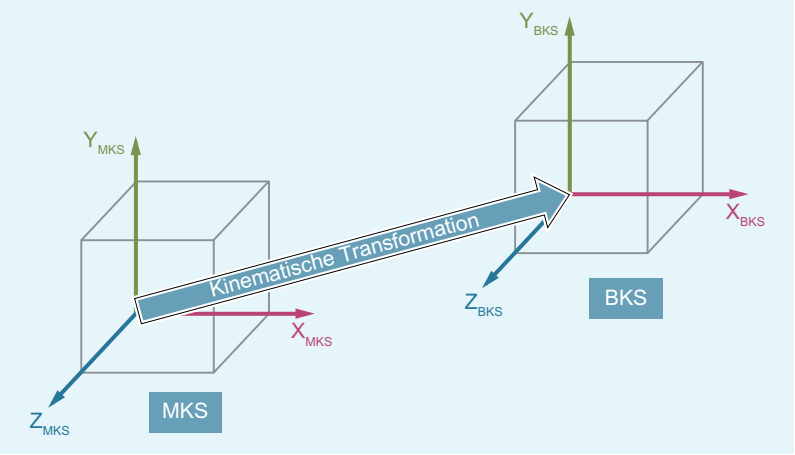
Bei diesen Maschinen können Maschinenachsen und Geometrieachsen den gleichen Namen haben.



### WZ-Maschinen mit kinematischer Transformation

Das BKS und das MKS fallen nicht zusammen, wenn das BKS mit kinematischer Transformation (z. B. 5-Achstransformation, TRANSMIT / TRACYL / TRAANG) auf das MKS abgebildet wird.

Bei diesen Maschinen müssen Maschinenachsen und Geometrieachsen unterschiedliche Namen haben.

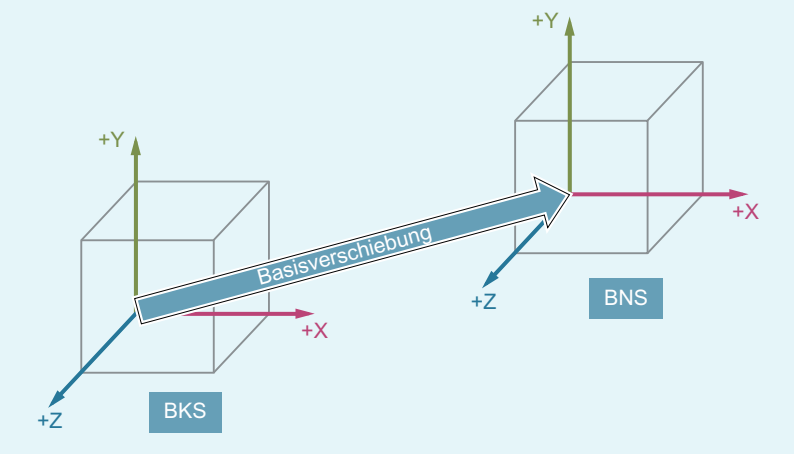



### Maschinenkinematik

Das Werkstück wird immer in einem zwei- oder dreidimensionalen rechtwinkligen Koordinatensystem (WKS) programmiert. Zur Fertigung dieser Werkstücke werden aber immer häufiger Werkzeugmaschinen mit Rundachsen oder nicht rechtwinklig angeordneten Linearachsen eingesetzt. Zur Abbildung der im WKS programmierten Koordinaten (rechtwinklig) in reale Maschinenachsbewegungen dient die kinematische Transformation.

## 2.1.4.3 Basis-Nullpunktsystem (BNS)

Das Basis-Nullpunktsystem (BNS) ergibt sich aus dem Basis-Koordinatensystem durch die Basisverschiebung.



### Basisverschiebung

Die Basisverschiebung beschreibt die Koordinatentransformation zwischen dem BKS und BNS. Mit ihr kann z. B. der Paletten-Nullpunkt festgelegt werden.

Die Basisverschiebung setzt sich zusammen aus:

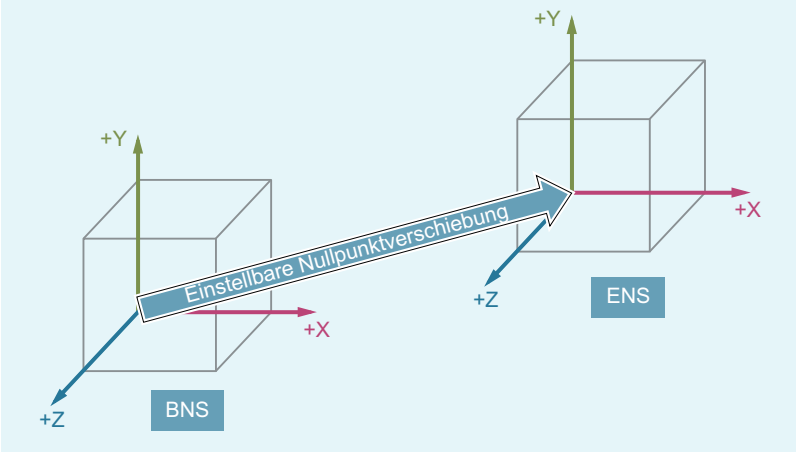
- Externe Nullpunktverschiebung
- DRF-Verschiebung
- Überlagerte Bewegung
- Verkettete Systemframes
- Verkettete Basisframes

## 2.1.4.4 Einstellbares Nullpunktsystem (ENS)

### Einstellbare Nullpunktverschiebung

Durch die einstellbare Nullpunktverschiebung ergibt sich aus dem Basis-Nullpunktsystem (BNS) das "Einstellbare Nullpunktsystem" (ENS).

Einstellbare Nullpunktverschiebungen werden im NC-Programm mit den G-Befehlen G54 ... G57 und G505 ... G599 aktiviert.




Wenn keine programmierbaren Koordinatentransformationen (Frames) aktiv sind, dann ist das "Einstellbare Nullpunktsystem" das Werkstück-Koordinatensystem (WKS).

### Programmierbare Koordinatentransformationen (Frames)

Manchmal erweist es sich als sinnvoll bzw. notwendig, innerhalb eines NC-Programms das ursprünglich gewählte Werkstück-Koordinatensystem (bzw. das "Einstellbare Nullpunktsystem") an eine andere Stelle zu verschieben und ggf. zu drehen, zu spiegeln und / oder zu skalieren. Dies erfolgt über programmierbare Koordinatentransformationen (Frames) (Seite 308).

**Hinweis**

Programmierbare Koordinatentransformationen (Frames) beziehen sich immer auf das "Einstellbare Nullpunktsystem".

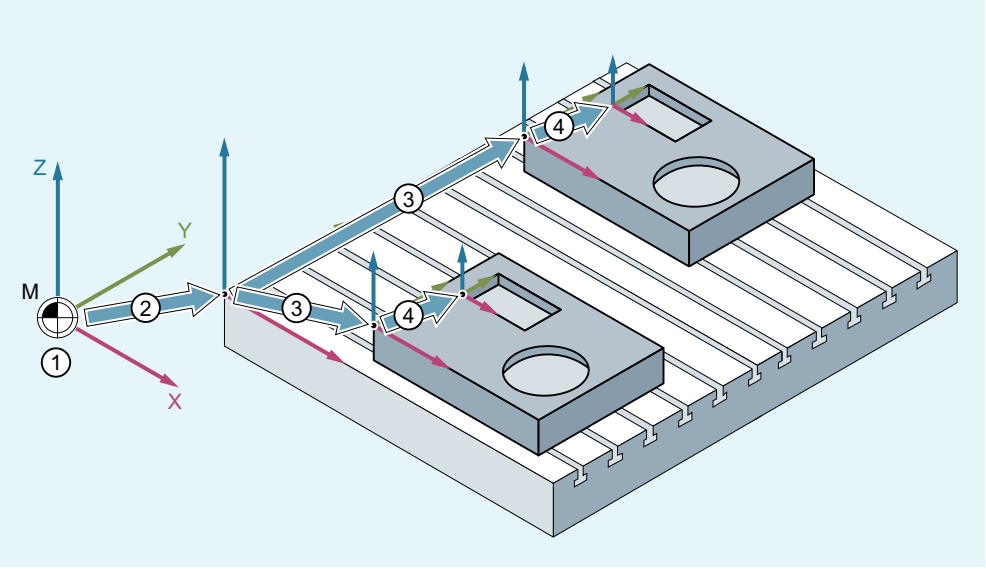
## 2.1.4.5 Werkstück-Koordinatensystem (WKS)

Im Werkstück-Koordinatensystem (WKS) wird die Geometrie eines Werkstücks beschrieben. Oder anders ausgedrückt: Die Angaben im NC-Programm beziehen sich auf das Werkstück-Koordinatensystem.

Das Werkstück-Koordinatensystem ist immer ein kartesisches Koordinatensystem und einem bestimmten Werkstück zugeordnet.

### 2.1.4.6 Wie hängen die verschiedenen Koordinatensysteme zusammen?

Das folgende Beispiel soll die Zusammenhänge zwischen den verschiedenen Koordinatensystemen verdeutlichen:




① Eine kinematische Transformation ist nicht aktiv. D. h., das Maschinenkoordinatensystem und das Basiskoordinatensystem fallen zusammen.

② Durch die Basisverschiebung ergibt sich das Basis-Nullpunktsystem (BNS) mit dem Paletten-Nullpunkt.

③ Durch die einstellbare Nullpunktverschiebung G54 bzw. G55 wird das "Einstellbare Nullpunktsystem" (ENS) für Werkstück 1 bzw. Werkstück 2 festgelegt.

④ Durch programmierbare Koordinatentransformation ergibt sich das Werkstückkoordinatensystem (WKS).

# 2.2 Grundlagen der NC-Programmierung

**Hinweis**

Richtlinie für die NC-Programmierung ist DIN 66025.

## 2.2.1 Benennung eines NC-Programms

### Regeln

Jedem NC-Programm muss beim Erstellen ein Programmname (Bezeichner) zugewiesen werden. Der Programmname kann unter Einhaltung folgender Regeln frei gewählt werden:

- Erlaubte Zeichen:
  - Buchstaben: A ... Z, a ... z
  - Ziffern: 0 ... 9
  - Unterstrich: _
- Die ersten beiden Zeichen sollten zwei Buchstaben oder ein Unterstrich gefolgt von einem Buchstaben sein.

  **Hinweis**

  Wenn diese Bedingung erfüllt ist, dann kann ein NC-Programm allein durch Angabe des Programmnamens als Unterprogramm aus einem anderen Programm heraus aufgerufen werden. Beginnt der Programmname hingegen mit Ziffern, dann ist der Unterprogramm-Aufruf nur über die CALL-Anweisung möglich.

- Maximale Länge: 24 Zeichen

**Hinweis**

**Groß- / Kleinbuchstaben**

In der SINUMERIK NC-Sprache wird **nicht** zwischen Groß- und Kleinbuchstaben unterschieden.

**Hinweis**

**Unzulässige Programmnamen**

Um Probleme mit Windows-Applikationen zu vermeiden, dürfen folgende Programmnamen **nicht** verwendet werden:

- CON, PRN, AUX, NUL
- COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9
- LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9

Weitere Einschränkungen siehe "Namen (Seite 386)".

#### Steuerungsinterne Erweiterungen

Der bei der Programmerstellung vergebene Programmname wird steuerungsintern mit einem Pre- und Postfix erweitert:

- Prefix: _N_
- Postfix:
  - Hauptprogramme: _MPF
  - Unterprogramme: _SPF

#### Dateien im Lochstreifenformat

Extern erstellte Programmdateien, die über die V.24-Schnittstelle eingelesen werden sollen, müssen im Lochstreifenformat vorliegen.

Für den Programmname einer Datei im Lochstreifenformat gelten folgende zusätzliche Regeln:

- Erstes Zeichen: %
- Abschließend eine vier Zeichen lange Dateikennung: _xxx

Beispiele:

- %_N_WELLE123_MPF
- %Flansch3_MPF

## 2.2.2 Aufbau und Inhalte eines NC-Programms

### 2.2.2.1 Sätze und Satzkomponenten

#### Sätze

Ein NC-Programm besteht aus einer Folge von NC-Sätzen. Jeder Satz enthält die Daten zur Ausführung eines Arbeitsschritts bei der Werkstückbearbeitung.

#### Satzkomponenten

NC-Sätze bestehen aus folgenden Komponenten:

- Befehle (Anweisungen) nach DIN 66025
- Elemente der NC-Hochsprache

#### Befehle nach DIN 66025

Die Befehle nach DIN 66025 bestehen aus einem Adresszeichen und einer Ziffer bzw. einer Ziffernfolge, die einen arithmetischen Wert darstellt.

**Adresszeichen (Adresse)**

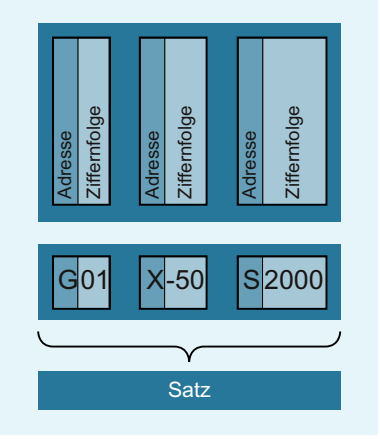
Das Adresszeichen (im Allgemeinen ein Buchstabe) definiert die Bedeutung des Befehls.

Beispiele:

| Adresszeichen | Bedeutung |
|---|---|
| G | G-Befehl (Wegbedingung) |
| X | Weginformation für Achse X |
| S | Spindeldrehzahl |

### Ziffernfolge

Die Ziffernfolge ist der dem Adresszeichen zugewiesene Wert. Die Ziffernfolge kann Vorzeichen und Dezimalpunkt beinhalten, wobei ein Vorzeichen immer zwischen dem Adressbuchstaben und der Ziffernfolge steht. Positive Vorzeichen (+) und führende Nullen (0) müssen nicht geschrieben werden.




## Elemente der NC-Hochsprache

Da der Befehlssatz nach DIN 66025 für die Programmierung der komplexen Bearbeitungsabläufe in modernen Werkzeugmaschinen nicht mehr ausreichend ist, wurde er um die Elemente der NC-Hochsprache erweitert.

Dazu gehören u. a.:

- Befehle der NC-Hochsprache
  Im Unterschied zu den Befehlen nach DIN 66025 bestehen die Befehle der NC-Hochsprache aus mehreren Adressbuchstaben, z. B.:
  - `OVR` für Drehzahlkorrektur (Override)
  - `SPOS` für Spindelpositionieren
- Bezeichner (definierte Namen) für:
  - Systemvariablen
  - Anwenderdefinierte Variablen
  - Unterprogramme
  - Schlüsselwörter
  - Sprungmarken
  - Makros

  **Hinweis**

  Ein Bezeichner muss eindeutig sein und darf nicht für verschiedene Objekte verwendet werden.
- Vergleichsoperatoren
- Logische Operatoren
- Rechenfunktionen
- Kontrollstrukturen

## Wirksamkeit von Befehlen

Befehle können modal oder satzweise wirken:

- Modal
  Modal wirksame Befehle behalten mit dem programmierten Wert so lange ihre Gültigkeit (in allen Folgesätzen), bis:
  - unter dem gleichen Befehl ein neuer Wert programmiert wird.
  - ein Befehl programmiert wird, der die Wirkung des bisher gültigen Befehls aufhebt.
- Satzweise
  Satzweise wirksame Befehle gelten nur für den Satz, in dem sie programmiert werden.

## Programmende

Der letzte Satz in den Abarbeitungsreihenfolgen enthält ein spezielles Wort für das Programmende: `M2`, `M17` bzw. `M30`.

### 2.2.2.2 Satzregeln

#### Satzanfang

NC-Sätze können am Satzanfang durch Satznummern gekennzeichnet werden. Diese bestehen aus dem Zeichen "N" und einer positiven ganzen Zahl, z. B.:
`N40 ...`

Die Reihenfolge der Satznummern ist beliebig, aufsteigende Satznummern sind empfehlenswert.

**Hinweis**

Satznummern müssen innerhalb eines Programms eindeutig sein, um beim Satzsuchlauf ein eindeutiges Ergebnis zu erzielen.

#### Satzende

Ein Satz endet mit dem Zeichen LF (LINE FEED = neue Zeile).

**Hinweis**

Das Zeichen LF muss nicht geschrieben werden. Es wird automatisch durch die Zeilenschaltung erzeugt.

#### Satzlänge

Ein Satz kann maximal **512 Zeichen** enthalten (inklusive Kommentar und Satzende-Zeichen LF).

**Hinweis**

Im Allgemeinen werden in der aktuellen Satzanzeige am Bildschirm drei Sätze mit jeweils maximal 66 Zeichen angezeigt. Kommentare werden ebenfalls angezeigt. Meldungen werden im eigenen Meldefenster angezeigt.

#### Reihenfolge der Anweisungen

Um den Satzaufbau übersichtlich zu gestalten, sollten die Anweisungen in einem Satz in folgender Reihenfolge angeordnet werden:
`N... G... X... Y... Z... F... S... T... D... M... H...`

| Adresse | Bedeutung |
|---|---|
| `N` | Adresse der Satznummer |
| `G` | Wegbedingung |
| `X,Y,Z` | Weginformation |
| `F` | Vorschub |

| | |
|---|---|
| S | Drehzahl |
| T | Werkzeug |
| D | Werkzeugkorrekturnummer |
| M | Zusatzfunktion |
| H | Hilfsfunktion |

**Hinweis**

Einige Adressen können innerhalb eines Satzes auch mehrfach verwendet werden, z. B.:

G..., M..., H...

### 2.2.2.3 Wertzuweisungen

Den Adressen können Werte zugewiesen werden. Dabei gelten folgende Regeln:

- Ein "="-Zeichen zwischen der Adresse und dem Wert muss geschrieben werden, wenn:
  - die Adresse aus mehr als einem Buchstaben besteht.
  - der Wert aus mehr als einer Konstanten besteht.

  Das "="-Zeichen kann entfallen, wenn die Adresse ein einzelner Buchstabe ist und der Wert aus nur einer Konstanten besteht.
- Vorzeichen sind erlaubt.
- Trennzeichen nach dem Adressbuchstaben sind zulässig.

Beispiele:

| | |
|---|---|
| X10 | Wertzuweisung (10) an die Adresse X, "=" nicht erforderlich |
| X1=10 | Wertzuweisung (10) an eine Adresse (X) mit numerischer Erweiterung (1), "=" erforderlich |
| X=10*(5+SIN(37.5)) | Wertzuweisung über einen numerischen Ausdruck, "=" erforderlich |

**Hinweis**

Nach einer numerischen Erweiterung muss immer eines der Sonderzeichen "=", "(", "[", ")", "]", "," oder ein Operator folgen, um die Adresse mit numerischer Erweiterung von einem Adressbuchstaben mit Wert zu unterscheiden.

### 2.2.2.4 Kommentare

Um die Verständlichkeit eines NC-Programms zu erhöhen, können die NC-Sätze mit Kommentaren versehen werden.

Ein Kommentar steht am Ende eines Satzes und wird durch Strichpunkt (";") vom Programmteil des NC-Satzes abgetrennt.

Beispiel 1:

| Programmcode | Kommentar |
|---|---|
| N10 G1 F100 X10 Y20 | ; Kommentar zur Erläuterung des NC-Satzes |

Beispiel 2:

| Programmcode | Kommentar |
|---|---|
| N10 | ; Firma G&S, Auftrag Nr. 12A71 |
| N20 | ; Programm erstellt von H. Müller, Abt. TV 4, am 21.11.94 |
| N50 | ; Teil Nr. 12, Gehäuse für Tauchpumpe Typ TP23A |

**Hinweis**

Kommentare werden abgespeichert und erscheinen beim Programmlauf in der aktuellen Satzanzeige.

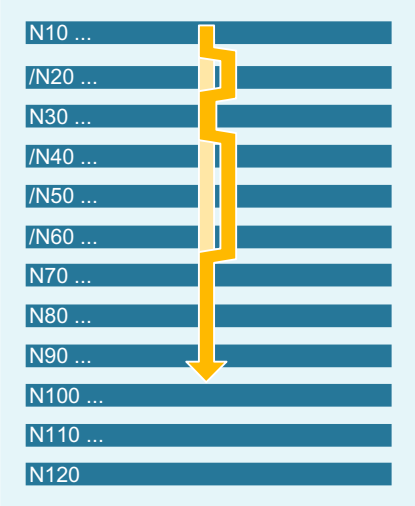
### 2.2.2.5 Ausblenden von Sätzen

NC-Sätze, die nicht bei jedem Programmlauf ausgeführt werden sollen, können für die Abarbeitung ausgeblendet werden. Anwendung findet diese Funktion z. B. beim Testen bzw. Einfahren neuer Programme.

### Ausblendsätze

Die Sätze, die ausgeblendet werden sollen, werden mit dem Zeichen "/" (Schrägstrich) vor der Satznummer gekennzeichnet. Es können auch mehrere Sätze in Folge ausgeblendet werden. Die Anweisungen in den ausgeblendeten Sätzen werden nicht ausgeführt, das Programm wird mit dem jeweils nächsten nicht ausgeblendeten Satz fortgeführt.

Beispiel:



| Programmcode | Kommentar |
|---|---|
| N10 … | ; wird abgearbeitet |
| /N20 … | ; ausgeblendet |
| N30 … | ; wird abgearbeitet |
| /N40 … | ; ausgeblendet |
| /N50 … | ; ausgeblendet |
| /N60 … | ; ausgeblendet |
| N70 … | ; wird abgearbeitet |
| ... | |

## Ausblendebenen

Sätzen können Ausblendebenen (max. 10) zugeordnet werden, die über die Bedienoberfläche oder das PLC-Anwenderprogramm aktivierbar sind.

Die Zuordnung erfolgt im NC-Programm durch Voranstellen eines Schrägstrichs, gefolgt von der Nummer der Ausblendebene. Pro Satz kann nur eine Ausblendebene angegeben werden.

Beispiel:

| Programmcode | Kommentar |
|---|---|
| / ... | ; Satz wird ausgeblendet (1. Ausblendebene) |
| /0 ... | ; Satz wird ausgeblendet (1. Ausblendebene) |
| /1 N010... | ; Satz wird ausgeblendet (2. Ausblendebene) |
| /2 N020... | ; Satz wird ausgeblendet (3. Ausblendebene) |
| ... | |
| /7 N100... | ; Satz wird ausgeblendet (8. Ausblendebene) |
| /8 N080... | ; Satz wird ausgeblendet (9. Ausblendebene) |
| /9 N090... | ; Satz wird ausgeblendet (10. Ausblendebene) |

### Hinweis

Änderungen der auszublendenden Ebenen sind nur im STOP/Reset-Zustand der Steuerung möglich.

### Hinweis

Wie viele Ausblendebenen nutzbar sind, ist abhängig von einem Anzeige-Maschinendatum.

### Hinweis

Das Ausblenden von Sätzen ist auch bei einem Satzsuchlauf aktiv.

### Hinweis

Veränderbare Programmabläufe können auch durch den Einsatz von System- und Anwendervariablen für bedingte Sprünge erzeugt werden.

# 2.3 Anlegen eines NC-Programms

## 2.3.1 Grundsätzliches Vorgehen

Beim Erstellen eines NC-Programms ist die Programmierung, also die Umsetzung der einzelnen Arbeitsschritte in die NC-Sprache, meist nur ein kleiner Teil der Programmier-Arbeit.

Vor der eigentlichen Programmierung sollte die Planung und Vorbereitung der Arbeitsschritte im Vordergrund stehen. Je genauer Sie sich vorab überlegen, wie das NC-Programm eingeteilt und aufgebaut sein soll, umso schneller und einfacher wird die eigentliche Programmierung von der Hand gehen und umso übersichtlicher und weniger fehleranfällig wird das fertige NC-Programm sein. Übersichtliche Programme erweisen sich außerdem besonders dann als vorteilhaft, wenn später Änderungen vorgenommen werden sollen.

Da nicht jedes Teil identisch aussieht, ist es nicht sinnvoll, jedes Programm genau nach derselben Methode zu erstellen. Für die meisten Fälle wird sich die folgende Vorgehensweise aber als zweckmäßig erweisen.

### Vorgehensweise

1. **Werkstückzeichnung vorbereiten**
   - Werkstücknullpunkt festlegen
   - Koordinatensystem einzeichnen
   - Eventuell fehlende Koordinaten berechnen
2. **Bearbeitungsablauf festlegen**
   - Welche Werkzeuge werden wann und zur Bearbeitung welcher Kontur eingesetzt?
   - In welcher Reihenfolge werden die Einzelelemente des Werkstücks gefertigt?
   - Welche Einzelelemente wiederholen sich (evtl. auch gedreht) und sollten in einem Unterprogramm abgelegt werden?
   - Gibt es in anderen Teileprogrammen bzw. Unterprogrammen Teilkonturen, die für das aktuelle Werkstück wiederverwendet werden können?
   - Wo ist Nullpunktverschieben, Drehen, Spiegeln, Skalieren zweckmäßig oder notwendig (Frame-Konzept)?

3. **Arbeitsplan aufstellen**
   Schrittweise alle Bearbeitungsvorgänge der Maschine festlegen, z. B.:
   - Eilgangbewegungen zum Positionieren
   - Werkzeugwechsel
   - Bearbeitungsebene festlegen
   - Freifahren zum Nachmessen
   - Spindel, Kühlmittel ein-/ausschalten
   - Werkzeugdaten aufrufen
   - Zustellen
   - Bahnkorrektur
   - Anfahren an die Kontur
   - Wegfahren von der Kontur
   - etc.
4. **Arbeitsschritte in die Programmiersprache übersetzen**
   - Jeden Einzelschritt als NC-Satz (bzw. NC-Sätze) aufschreiben.
5. **Alle Einzelschritte zu einem Programm zusammenfassen**

## 2.3.2 Verfügbare Zeichen

Für die Erstellung von NC-Programmen stehen folgende Zeichen zur Verfügung:

- Großbuchstaben:
  A, B, C, D, E, F, G, H, I, J, K, L, M, N,(O),P, Q, R, S, T, U, V, W, X, Y, Z
- Kleinbuchstaben:
  a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z
- Ziffern:
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Sonderzeichen:
  Siehe nachfolgende Tabelle!

| Sonderzeichen | Bedeutung |
|---|---|
| % | Programmanfangszeichen (nur für Programmerstellung am externen PC) |
| ( | Klammerung von Parametern oder in Ausdrücken |
| ) | Klammerung von Parametern oder in Ausdrücken |
| [ | Klammerung von Adressen oder Feldindizes |
| ] | Klammerung von Adressen oder Feldindizes |
| < | kleiner |
| > | größer |
| : | Hauptsatz, Labelabschluss, Kettungsoperator |
| = | Zuweisung, Teil von Gleichheit |

| Sonderzeichen | Bedeutung |
|---|---|
| / | Division, Satzunterdrückung |
| * | Multiplikation |
| + | Addition |
| - | Subtraktion, negatives Vorzeichen |
| " | Anführungszeichen, Kennung für Zeichenkette |
| ' | Hochkomma, Kennung für spezielle Zahlenwerte: hexadezimal, binär |
| $ | systemeigene Variablenkennung |
| s_ | Unterstrich, zu Buchstaben gehörig |
| ? | reserviert |
| ! | reserviert |
| . | Dezimalpunkt |
| , | Komma, Trennzeichen von Parametern |
| ; | Kommentarbeginn |
| & | Formatierungszeichen, gleiche Wirkung wie Leerzeichen |
| LF | Satzende |
| Tabulator | Trennzeichen |
| Leerzeichen | Trennzeichen (Blank) |

**Hinweis**

Buchstabe "O" nicht mit der Zahl "0" verwechseln!

**Hinweis**

Klein- und Großbuchstaben werden nicht unterschieden (Ausnahme: Werkzeugaufruf).

**Hinweis**

Nicht darstellbare Sonderzeichen werden wie Leerzeichen behandelt.

## 2.3.3 Programmkopf

Die NC-Sätze, die den eigentlichen Bewegungssätzen zur Herstellung der Werkstückkontur vorangestellt sind, werden als Programmkopf bezeichnet.

Der Programmkopf enthält Informationen / Anweisungen bezüglich:

- Werkzeugwechsel
- Werkzeugkorrekturen
- Spindelbewegung
- Vorschubregelung
- Geometrieeinstellungen (Nullpunktverschiebung, Wahl der Arbeitsebene)

### Programmkopf beim Drehen

Das folgende Beispiel zeigt, wie der Programmkopf eines NC-Programms zum Drehen typischerweise aufgebaut ist:

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 G0 G153 X200 Z500 T0 D0` | `; Werkzeugträger zurückziehen, bevor Werkzeugrevolver geschwenkt wird.` |
| `N20 T5` | `; Werkzeug 5 einschwenken.` |
| `N30 D1` | `; Schneidendatensatz des Werkzeugs aktivieren.` |
| `N40 G96 S300 LIMS=3000 M4 M8` | `; Konstante Schnittgeschwindigkeit (Vc) = 300 m/min, Drehzahlbegrenzung = 3000 U/min, Drehrichtung links, Kühlung ein.` |
| `N50 DIAMON` | `; X-Achse wird im Durchmesser programmiert.` |
| `N60 G54 G18 G0 X82 Z0.2` | `; Nullpunktverschiebung und Arbeitsebene aufrufen, Startposition anfahren.` |
| `...` | |

### Programmkopf beim Fräsen

Das folgende Beispiel zeigt, wie der Programmkopf eines NC-Programms zum Fräsen typischerweise aufgebaut ist:

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 T="SF12"` | `; alternativ: T123` |
| `N20 M6` | `; Werkzeugwechsel auslösen` |
| `N30 D1` | `; Schneidendatensatz des Werkzeugs aktivieren` |
| `N40 G54 G17` | `; Nullpunktverschiebung und Arbeitsebene` |
| `N50 G0 X0 Y0 Z2 S2000 M3 M8` | `; Anfahrbewegung zum Werkstück, Spindel und Kühlmittel ein` |
| `...` | |

Wenn mit Werkzeugorientierung / Koordinatentransformationen gearbeitet wird, sollten am Programmanfang evtl. noch aktive Transformationen abgelöscht werden:

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 CYCLE800()` | `; Rücksetzten der geschwenkten Ebene` |
| `N20 TRAFOOF` | `; Rücksetzen von TRAORI, TRANSMIT, TRACYL, ...` |
| `...` | |

## 2.3.4 Programmbeispiele

### 2.3.4.1 Beispiel 1: Erste Programmierschritte

Programmbeispiel 1 soll dazu dienen, erste Programmierschritte durchzuführen und zu testen.

### Vorgehensweise

1. Teileprogramm neu anlegen (Namen)
2. Teileprogramm editieren
3. Teileprogramm auswählen
4. Einzelsatz aktivieren
5. Teileprogramm starten

**Hinweis**

Damit das Programm auf der Maschine ablaufen kann, müssen Maschinendaten entsprechend gesetzt sein (→ siehe Angaben des Maschinenherstellers!).

**Hinweis**

Beim Testen eines Programms können Alarme auftreten, die erst zurückgesetzt werden müssen.

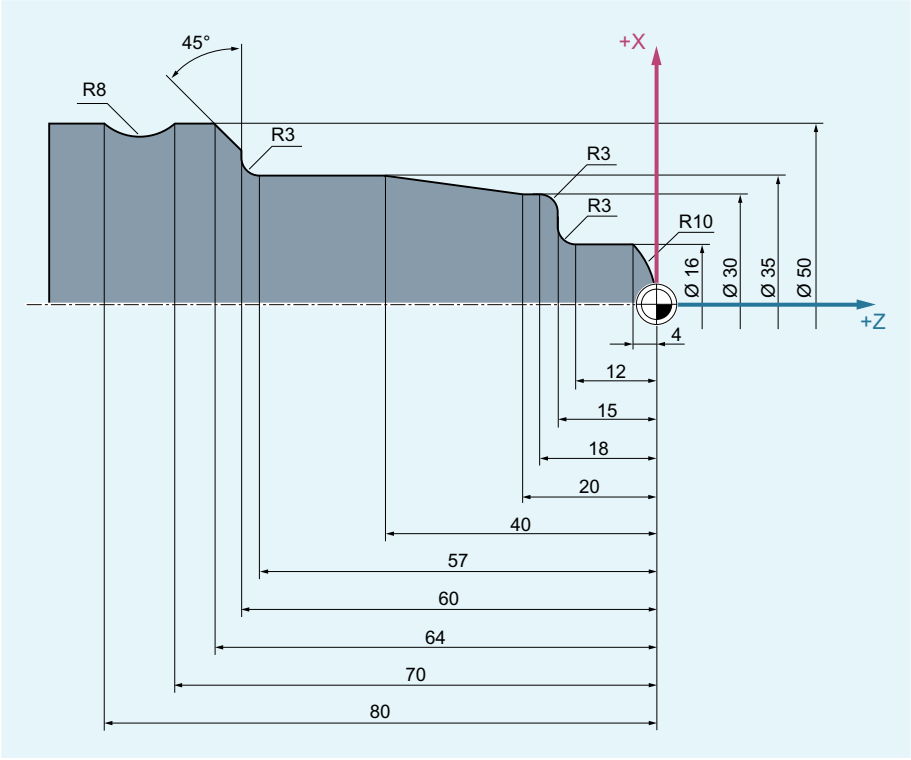
### NC-Programm

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 MSG("DAS IST MEIN NC-PROGRAMM")` | `; Meldung "DAS IST MEIN NC-PROGRAMM" in Alarmzeile ausgeben` |
| `N20 F200 S900 T1 D2 M3` | `; Vorschub, Spindel, Werkzeug, Werkzeugkorrektur, Spindel rechts` |
| `N30 G0 X100 Y100` | `; Position im Eilgang anfahren` |
| `N40 G1 X150` | `; Rechteck mit Vorschub, Gerade in X` |
| `N50 Y120` | `; Gerade in Y` |
| `N60 X100` | `; Gerade in X` |
| `N70 Y100` | `; Gerade in Y` |
| `N80 G0 X0 Y0` | `; Rückfahren im Eilgang` |
| `N100 M30` | `; Satzende` |

## 2.3.4.2 Beispiel 2: NC-Programm zum Drehen

Programmbeispiel 2 ist für die Bearbeitung eines Werkstücks auf einer Drehmaschine vorgesehen. Es beinhaltet Radiusprogrammierung und Werkzeugradiuskorrektur.

## Maßzeichnung des Werkstücks




## NC-Programm

| **Programmcode** | **Kommentar** |
|---|---|
| N5 G0 G53 X280 Z380 D0 | ; Startpunkt |
| N10 TRANS X0 Z250 | ; Nullpunktverschiebung |
| N15 LIMS=4000 | ; Drehzahlbegrenzung (G96) |
| N20 G96 S250 M3 | ; konstante Schnittgeschwindigkeit anwählen |
| N25 G90 T1 D1 M8 | ; Werkzeuganwahl und Korrektur anwählen |
| N30 G0 G42 X-1.5 Z1 | ; Werkzeug anstellen mit Werkzeugradiuskorrektur |
| N35 G1 X0 Z0 F0.25 | |
| N40 G3 X16 Z-4 I0 K-10 | ; Radius 10 drehen |
| N45 G1 Z-12 | |
| N50 G2 X22 Z-15 CR=3 | ; Radius 3 drehen |
| N55 G1 X24 | |
| N60 G3 X30 Z-18 I0 K-3 | ; Radius 3 drehen |
| N65 G1 Z-20 | |
| N70 X35 Z-40 | |
| N75 Z-57 | |
| N80 G2 X41 Z-60 CR=3 | ; Radius 3 drehen |
| N85 G1 X46 | |
| N90 X52 Z-63 | |

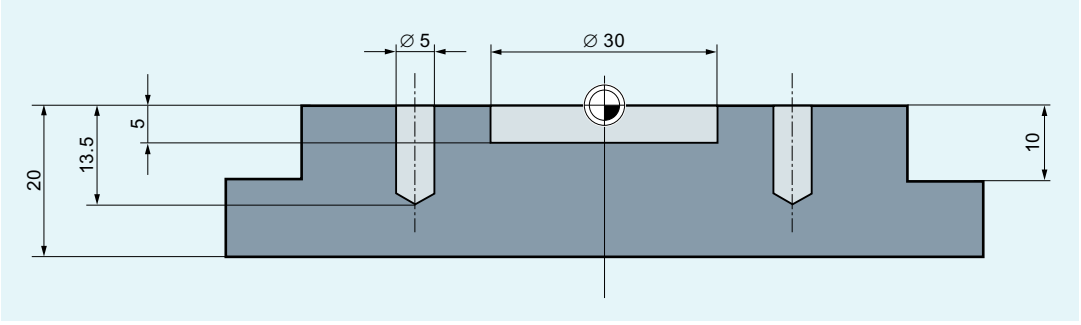
| **Programmcode** | **Kommentar** |
|---|---|
| N95 G0 G40 G97 X100 Z50 M9 | ; Werkzeugradiuskorrektur abwählen und Werkzeugwechselpunkt anfahren |
| N100 T2 D2 | ; Werkzeug aufrufen und Korrektur anwählen |
| N105 G96 S210 M3 | ; konstante Schnittgeschwindigkeit anwählen |
| N110 G0 G42 X50 Z-60 M8 | ; Werkzeug anstellen mit Werkzeugradiuskorrektur |
| N115 G1 Z-70 F0.12 | ; Durchmesser 50 drehen |
| N120 G2 X50 Z-80 I6.245 K-5 | ; Radius 8 drehen |
| N125 G0 G40 X100 Z50 M9 | ; Werkzeug abheben und Werkzeugradiuskorrektur abwählen |
| N130 G0 G53 X280 Z380 D0 M5 | ; Werkzeugwechselpunkt verfahren |
| N135 M30 | ; Programm-Ende |

### 2.3.4.3 Beispiel 3: NC-Programm zum Fräsen

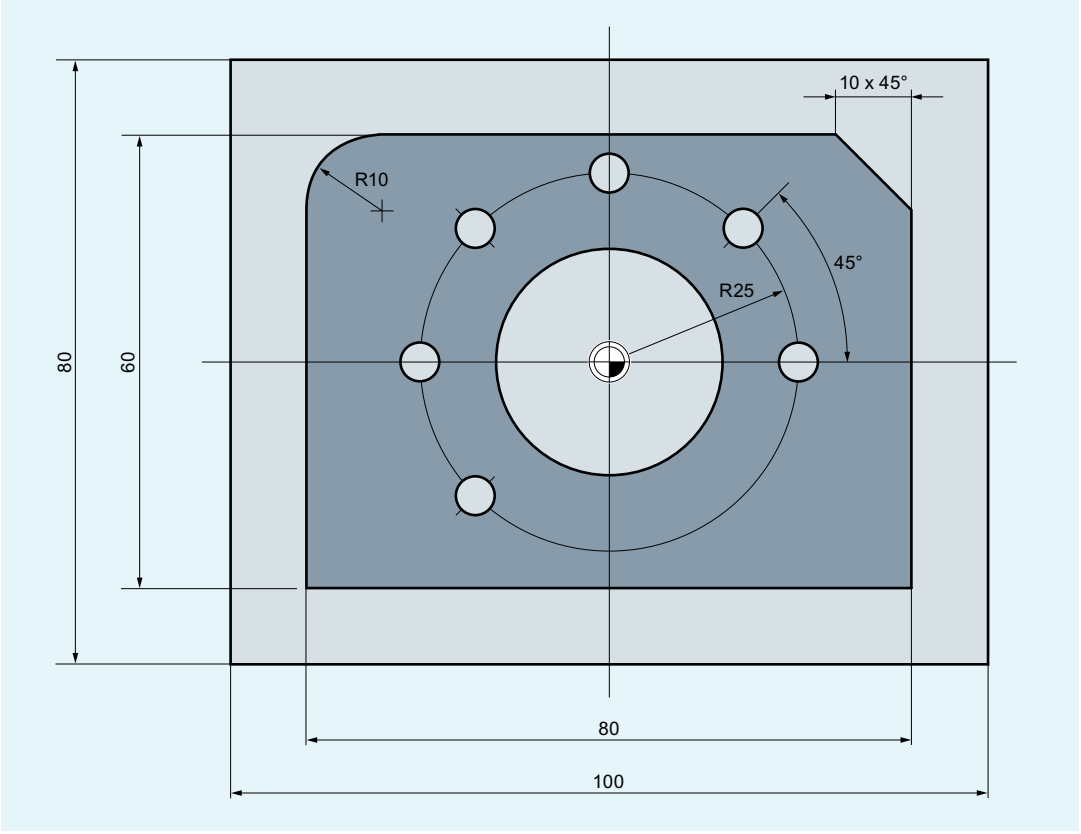
Programmbeispiel 3 ist für die Bearbeitung eines Werkstücks auf einer Vertikalfräsmaschine vorgesehen. Es beinhaltet Oberflächen- und Seitenfräsen sowie Bohren.

## Maßzeichnung des Werkstücks

### Seitenansicht



### Aufsicht




## NC-Programm

| **Programmcode** | **Kommentar** |
|---|---|
| N10 T="PF60" | ; Voranwahl des Werkzeugs mit dem Namen PF60. |
| N20 M6 | ; Werkzeug in die Spindel einwechseln. |
| N30 S2000 M3 M8 | ; Drehzahl, Drehrichtung, Kühlung ein. |
| N40 G90 G64 G54 G17 G0 X-72 Y-72 | ; Grundeinstellungen der Geometrie und Startpunkt anfahren. |
| N50 G0 Z2 | ; Z-Achse auf Sicherheitsabstand. |
| N60 G450 CFTCP | ; Verhalten bei aktivem G41/G42. |
| N70 G1 Z-10 F3000 | ; Fräser auf Eingriffstiefe mit Vorschub=3000mm/min. |
| N80 G1 G41 X-40 | ; Einschalten der Fräserradiuskorrektur. |
| N90 G1 X-40 Y30 RND=10 F1200 | ; Fahren an der Kontur mit Vorschub=1200mm/min. |
| N100 G1 X40 Y30 CHR=10 | |
| N110 G1 X40 Y-30 | |
| N120 G1 X-41 Y-30 | |
| N130 G1 G40 Y-72 F3000 | ; Abwahl der Fräserradiuskorrektur. |

| **Programmcode** | **Kommentar** |
|---|---|
| N140 G0 Z200 M5 M9 | ; Ausheben des Fräsers, Spindel + Kühlung aus. |
| N150 T="SF10" | ; Voranwahl des Werkzeugs mit dem Namen SF10. |
| N160 M6 | ; Werkzeug in die Spindel einwechseln. |
| N170 S2800 M3 M8 | ; Drehzahl, Drehrichtung, Kühlung ein. |
| N180 G90 G64 G54 G17 G0 X0 Y0 | ; Grundeinstellungen für Geometrie und Startpunkt anfahren. |
| N190 G0 Z2 | |
| N200 POCKET4(2,0,1,-5,15,0,0,0,0,0,800,1300,0,21,5,,,2,0.5) | ; Aufruf des Taschenfräszyklus. |
| N210 G0 Z200 M5 M9 | ; Ausheben des Fräsers, Spindel + Kühlung aus. |
| N220 T="ZB6" | ; Zentrierbohrer 6mm aufrufen. |
| N230 M6 | |
| N240 S5000 M3 M8 | |
| N250 G90 G60 G54 G17 X25 Y0 | ; Genauhalt G60 wegen genauem Positionieren. |
| N260 G0 Z2 | |
| N270 MCALL CYCLE82(2,0,1,-2.6,,0) | ; Modaler Aufruf des Bohrzyklus. |
| N280  POSITION: | ; Sprungmarke zur Wiederholung. |
| N290  HOLES2(0,0,25,0,45,6) | ; Positionsmuster für Bohrbild. |
| N300  ENDLABEL: | ; Endkennung für Wiederholung. |
| N310 MCALL | ; Rücksetzen des modalen Aufrufs. |
| N320 G0 Z200 M5 M9 | |
| N330 T="SPB5" | ; Spiralbohrer D5mm aufrufen. |
| N340 M6 | |
| N350 S2600 M3 M8 | |
| N360 G90 G60 G54 G17 X25 Y0 | |
| N370 MCALL CYCLE82(2,0,1,-13.5,,0) | ; Modaler Aufruf des Bohrzyklus. |
| N380  REPEAT POSITION | ; Wiederholung der Positionsbeschreibung vom Zentrieren. |
| N390 MCALL | ; Rücksetzen des Bohrzyklus. |
| N400 G0 Z200 M5 M9 | |
| N410 M30 | ; Programmende. |

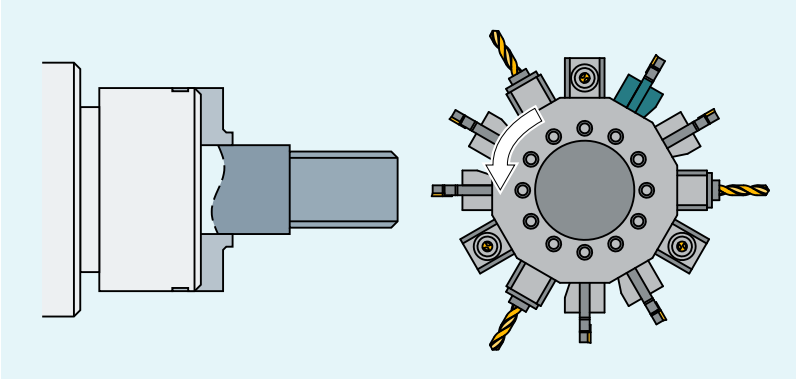
# 2.4 Werkzeugwechsel

## Art des Werkzeugwechsels

**Hinweis**

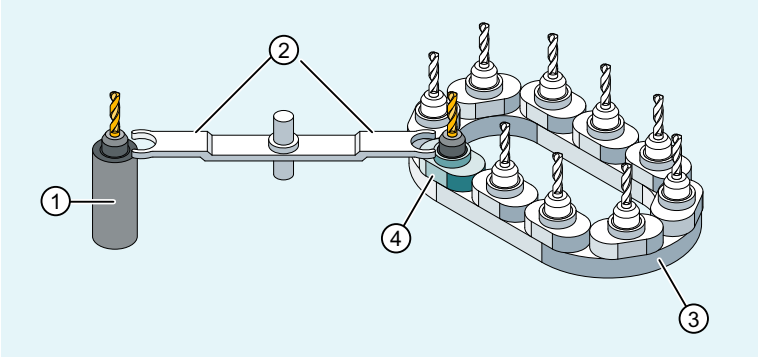
Die Art des Werkzeugwechsels wird vom Maschinenhersteller bei der Inbetriebnahme festgelegt.

### Werkzeugwechsel bei Drehmaschinen mit Revolvermagazin



Bei Revolvermagazinen an Drehmaschinen wird der Werkzeugwechsel, also das Suchen und Wechseln des Werkzeugs, nur mit dem T-Befehl aufgerufen.

### Werkzeugwechsel bei Werkzeugmaschinen mit Ketten-, Scheiben- oder Flächenmagazinen




① Spindel
② Greifer
③ Magazin (hier: Kettenmagazin)
④ Wechselstelle für Spindel

Bei Ketten-, Scheiben- und Flächenmagazinen findet ein Werkzeugwechselvorgang normalerweise in zwei Schritten statt:

1. Mit dem T-Befehl wird das Werkzeug im Magazin gesucht.
2. Anschließend erfolgt mit dem M-Befehl das Einwechseln in die Spindel.

**Programmierung des Werkzeugwechsel bei aktiver / nicht aktiver Werkzeugverwaltung**

Die Programmierung des Werkzeugwechsels bei aktiver Werkzeugverwaltung unterscheidet sich von der Programmierung des Werkzeugwechsels bei nicht aktiver Werkzeugverwaltung. Es sind daher beide Varianten beschrieben:

→ Werkzeugwechsel bei aktiver Werkzeugverwaltung (Seite 61)

→ Werkzeugwechsel bei nicht aktiver Werkzeugverwaltung (Seite 65)

### Programmierung der Arbeitsebene

Mit dem Werkzeugwechsel muss die entsprechende Arbeitsebene (Seite 33) programmiert werden (Grundstellung: G18). Damit wird sichergestellt, dass die Werkzeuglängenkorrektur der richtigen Achse zugeordnet ist.

### Aktivierung der Werkzeugkorrektur

Der Werkzeugwechsel aktiviert die unter einer D-Nummer (Seite 81) gespeicherten Werkzeugkorrekturwerte.

## 2.4.1 Werkzeugwechsel bei aktiver Werkzeugverwaltung

### Werkzeugverwaltung

Die Funktion "Werkzeugverwaltung" stellt sicher, dass an der Maschine zu jeder Zeit das richtige Werkzeug am richtigen Platz ist und die einem Werkzeug zugeordneten Daten dem aktuellen Stand entsprechen. Außerdem ermöglicht sie ein schnelles Einwechseln eines Werkzeugs, vermeidet Ausschuss durch Überwachung der Werkzeugeinsatzzeit sowie der Maschinenstillstandszeit durch Berücksichtigung von Ersatzwerkzeugen.

### Werkzeugnamen

An einer Werkzeugmaschine mit aktiver Werkzeugverwaltung müssen die Werkzeuge zur eindeutigen Identifikation mit Namen und Nummern versehen werden (z. B. "Bohrer", "3").

Der Werkzeug-Aufruf kann dann über den Werkzeugnamen erfolgen, z. B.:
`T="Bohrer"`

**Hinweis**

Der Werkzeugnamen darf keine Sonderzeichen enthalten.

### 2.4.1.1 Werkzeugwechsel bei aktiver WZV mit der Funktion T=Platz

Mit Programmierung der Funktion T=Platz erfolgt ein direkter Werkzeugwechsel.

## Anwendung

Bei Drehmaschinen mit Revolvermagazin.

## Syntax

**Werkzeug anwählen**

```
T=<Nr>
T=<Name>
T<n>=<Nr>
T<n>=<Name>
```

**Werkzeug abwählen**

```
T0
```

## Bedeutung

| | | |
|---|---|---|
| `T=`: | Befehl zum Werkzeugwechsel und Aktivierung der Werkzeugkorrektur<br>Als Angaben sind möglich: | |
| | `<Nr>`: | Nummer des Magazinplatzes |
| | `<Name>`: | Name des Werkzeugs<br>**Hinweis:**<br>Bei der Programmierung eines Werkzeugnamens muss auf die korrekte Schreibweise (Groß-/Kleinschreibung) geachtet werden. |
| `<n>`: | Spindelnummer als Adresserweiterung<br>**Hinweis:**<br>Die Möglichkeit, eine Spindelnummer als Adresserweiterung zu programmieren, ist von der Projektierung der Maschine abhängig (→ siehe Angaben des Maschinenherstellers). | |
| `T0`: | Werkzeugabwahl (Magazinplatz nicht besetzt) | |

**Hinweis**

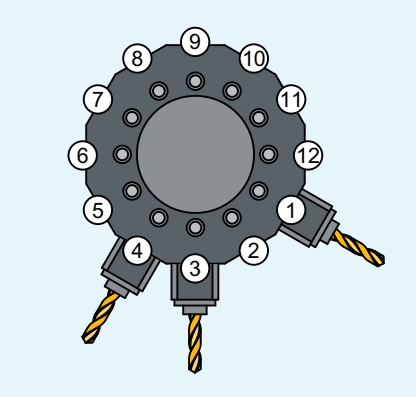
Ist in einem Werkzeugmagazin der angewählte Magazinplatz nicht besetzt, wirkt der Werkzeugbefehl wie T0. Die Anwahl des nicht besetzten Magazinplatzes kann zum Positionieren des leeren Platzes benutzt werden.

## Beispiel

Ein Revolvermagazin hat die Plätze 1 bis 12 mit folgender Werkzeugbesetzung:

| Platz | Werkzeug | Interne T-Nummer | Zustand |
|---|---|---|---|
| 1 | Bohrer, Duplo-Nr. = 1 | T15 | gesperrt |
| 2 | nicht besetzt | | |

| Platz | Werkzeug | Interne T-Nummer | Zustand |
|---|---|---|---|
| 3 | Bohrer, Duplo-Nr. = 2 | T10 | freigegeben |
| 4 | Bohrer, Duplo-Nr. = 3 | T1 | aktiv |
| 5 ... 12 | nicht besetzt | | |




① ... ⑫ Magazinplatznummern

Im NC-Programm ist folgender Werkzeug-Aufruf programmiert:

```
N10 T=1
```

Der Aufruf wird wie folgt verarbeitet:

1. Der Magazinplatz 1 wird betrachtet und dabei der Bezeichner des Werkzeugs ermittelt.
2. Die Werkzeugverwaltung erkennt, dass dieses Werkzeug gesperrt und somit nicht einsatzfähig ist.
3. Eine WZ-Suche nach T="Bohrer" wird entsprechend der eingestellten Suchstrategie gestartet:
   "Suche das aktive WZ, sonst nimm das mit der nächst größeren Duplo-Nr."
4. Als einsatzfähiges Werkzeug wird gefunden:
   "Bohrer" Duplo-Nr. 3 (auf Magazinplatz 4)
   **Damit ist die Werkzeug-Anwahl abgeschlossen und es wird der Werkzeugwechsel angestoßen**.

**Hinweis**

Mit der Suchstrategie "Nimm das erste verfügbare Werkzeug aus der Gruppe" ist die Reihenfolge der Werkzeuge nicht definiert. In diesem Fall wird T10 **oder** T1 ausgewählt.

Mit der Suchstrategie "Nimm das erste Werkzeug mit dem Status 'aktiv' aus der Gruppe" wird T1 eingewechselt.

### 2.4.1.2 Werkzeugwechsel bei aktiver WZV mit M6

Mit Programmierung des T-Befehls wird das Werkzeug angewählt. Aktiv wird das Werkzeug erst mit M6 (inklusive Werkzeugkorrektur).

## Anwendung

Bei Werkzeugmaschinen mit Ketten-, Scheiben- oder Flächenmagazinen.

## Syntax

**Werkzeug anwählen**
```
T=<Nr>
T=<Name>
T<n>=<Nr>
T<n>=<Name>
```

**Werkzeugwechsel**
```
M6
```

**Werkzeug abwählen**
```
T0
```

## Bedeutung

| | | |
|---|---|---|
| T=: | Adresse zur Werkzeuganwahl<br>Als Angaben sind möglich: | |
| | <Nr>: | Nummer des Magazinplatzes |
| | <Name>: | Name des Werkzeugs<br>**Hinweis:**<br>Bei der Programmierung eines Werkzeugnamens muss auf die korrekte Schreibweise (Groß-/Kleinschreibung) geachtet werden. |
| <n>: | Spindelnummer als Adresserweiterung<br>**Hinweis:**<br>Die Möglichkeit, eine Spindelnummer als Adresserweiterung zu programmieren, ist von der Projektierung der Maschine abhängig (→ siehe Angaben des Maschinenherstellers). | |
| M6: | M-Funktion für den Werkzeugwechsel (gemäß DIN 66025)<br>Mit M6 werden das angewählte Werkzeug (T…) und die Werkzeugkorrektur (D...) aktiv. | |
| T0: | Werkzeugabwahl (Magazinplatz nicht besetzt) | |

## Beispiel

| **Programmcode** | **Kommentar** |
|---|---|
| N10 T=1 M6 | ; Einwechseln des Werkzeugs von Magazinplatz 1. |
| N20 D1 | ; Anwahl Werkzeuglängenkorrektur. |
| N30 G1 X10 ... | ; Arbeiten mit Werkzeug T=1. |
| ... | |

| **Programmcode** | **Kommentar** |
|---|---|
| N70 T="Bohrer" | ; Vorwahl von Werkzeug mit Namen "Bohrer". |
| N80 ... | ; Arbeiten mit Werkzeug T=1. |
| ... | |
| N100 M6 | ; Einwechseln des Bohrers. |
| N140 D1 G1 X10 ... | ; Arbeiten mit Bohrer. |
| ... | |

## 2.4.2 Werkzeugwechsel bei nicht aktiver Werkzeugverwaltung

### 2.4.2.1 Werkzeugwechsel bei nicht aktiver WZV mit T-Nummer

Mit Programmierung der T-Nummer erfolgt ein direkter Werkzeugwechsel.

#### Anwendung

Bei Drehmaschinen mit Revolvermagazin.

#### Syntax

**Werkzeug anwählen**

```
T<Nr>
T=<Nr>
T<n>=<Nr>
```

**Werkzeug abwählen**

```
T0
T0=<Nr>
```

#### Bedeutung

| | | |
|---|---|---|
| T: | Adresse zur Werkzeuganwahl inklusive Werkzeugwechsel und Aktivierung der Werkzeugkorrektur | |
| <n>: | Spindelnummer als Adresserweiterung<br>**Hinweis:**<br>Die Möglichkeit, eine Spindelnummer als Adresserweiterung zu programmieren, ist von der Projektierung der Maschine abhängig (→ siehe Angaben des Maschinenherstellers). | |
| <Nr>: | Nummer des Werkzeugs | |
| | Wertebereich: | 0 ... 32000 |
| T0: | Abwählen des aktiven Werkzeugs | |

### Beispiel

| Programmcode | Kommentar |
|---|---|
| N10 T1 D1 | ; Einwechseln von Werkzeug T1 und Aktivieren der Werkzeugkorrektur D1. |
| ... | |
| N70 T0 | ; Werkzeug T1 abwählen. |
| ... | |

## 2.4.2.2 Werkzeugwechsel bei nicht aktiver WZV mit M6

Mit der Programmierung des T-Befehls wird das Werkzeug angewählt. Aktiv wird das Werkzeug erst mit M6 (inklusive Werkzeugkorrektur).

### Anwendung

Bei Werkzeugmaschinen mit Ketten-, Scheiben- oder Flächenmagazinen.

### Syntax

**Werkzeug anwählen**
```
T<Nr>
T=<Nr>
T<n>=<Nr>
```

**Werkzeugwechsel**
```
M6
```

**Werkzeug abwählen**
```
T0
T0=<Nr>
```

### Bedeutung

| | | |
|---|---|---|
| T: | Adresse zur Werkzeuganwahl | |
| <n>: | Spindelnummer als Adresserweiterung<br>**Hinweis:**<br>Die Möglichkeit, eine Spindelnummer als Adresserweiterung zu programmieren, ist von der Projektierung der Maschine abhängig (→ siehe Angaben des Maschinenherstellers). | |
| <Nr>: | Nummer des Werkzeugs | |
| | Wertebereich: | 0 ... 32000 |
| M6: | M-Funktion für den Werkzeugwechsel (gemäß DIN 66025)<br>Mit M6 werden das angewählte Werkzeug (T…) und die Werkzeugkorrektur (D...) aktiv. | |
| T0: | Abwählen des aktiven Werkzeugs | |

### Beispiel

```
Programmcode                    Kommentar
N10 T1 M6                       ; Einwechseln von Werkzeug T1.
N20 D1                          ; Anwahl Werkzeuglängenkorrektur.
N30 G1 X10 ...                  ; Arbeiten mit T1.
...
N70 T5                          ; Vorwahl von Werkzeug T5.
N80 ...                         ; Arbeiten mit T1.
...
N100 M6                         ; Einwechseln von Werkzeug T5.
N110 D1 G1 X10 ...              ; Arbeiten mit Werkzeug T5
...
```

## 2.4.3 Verhalten bei fehlerhafter T-Programmierung

Das Verhalten bei einer fehlerhaften T-Programmierung ist abhängig von der Projektierung der Maschine:

| MD22562 TOOL_CHANGE_ERROR_MODE | | |
|---|---|---|
| **Bit** | **Wert** | **Bedeutung** |
| 7 | 0 | Grundstellung!<br>Bei der T-Programmierung wird sofort geprüft, ob die T-Nummer dem NC bekannt ist. Wenn dies nicht der Fall, wird ein Alarm abgesetzt. |
| | 1 | Die programmierte T-Nummer wird erst geprüft, wenn die D-Anwahl erfolgt ist. Wenn die T-Nummer dem NC nicht bekannt ist, dann wird bei D-Anwahl ein Alarm gesetzt.<br>Dieses Verhalten ist dann gewünscht, wenn die T-Programmierung z. B. auch eine Positionierung bewirken soll und dafür die Werkzeug-Daten nicht vorhanden sein müssen (Revolver-Magazin). |

# 2.5 Werkzeugkorrekturen

## 2.5.1 Programmierte Kontur und Werkzeugbahn

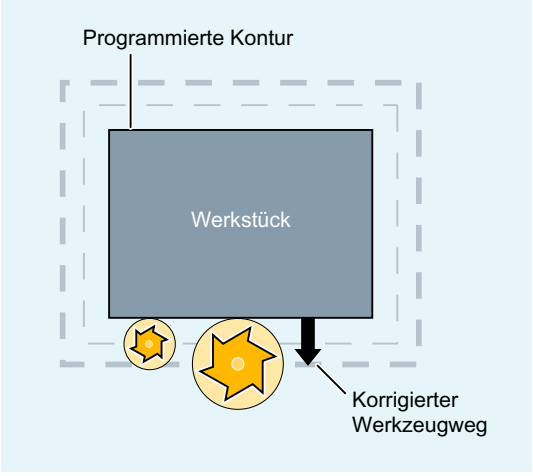
Werkstückmaße werden direkt programmiert (z. B. nach Fertigungszeichnung). Werkzeugdaten wie Fräserdurchmesser, Schneidenlage der Drehmeißel (linker / rechter Drehmeißel) und Werkzeuglängen müssen daher bei der Programmerstellung nicht berücksichtigt werden.

### Die Steuerung korrigiert den Verfahrweg

Bei der Fertigung eines Werkstücks werden die Werkzeugwege abhängig von der jeweiligen Werkzeuggeometrie so gesteuert, dass mit jedem eingesetzten Werkzeug die programmierte Kontur hergestellt werden kann.

Damit die Steuerung die Werkzeugwege berechnen kann, müssen die Werkzeugdaten im Werkzeugkorrekturspeicher der Steuerung eingetragen sein. Über das NC-Programm werden lediglich das benötigte Werkzeug (T...) und der benötigte Korrekturdatensatz (D...) aufgerufen.

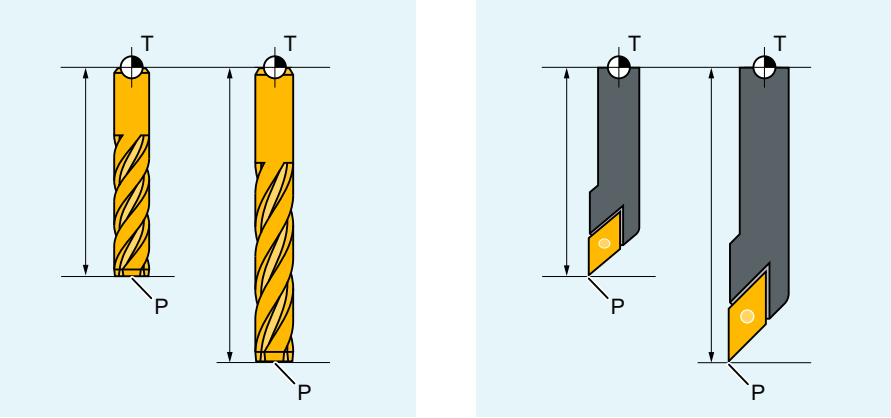
Die Steuerung holt sich während der Programmverarbeitung die benötigten Korrekturdaten aus dem Werkzeugkorrekturspeicher und korrigiert für unterschiedliche Werkzeuge individuell die Werkzeugbahn:




## 2.5.2 Werkzeuglängenkorrektur

Mit der Werkzeuglängenkorrektur werden die Längenunterschiede zwischen den eingesetzten Werkzeugen ausgeglichen.

Als Werkzeuglänge gilt der Abstand zwischen Werkzeugträgerbezugspunkt und Werkzeugspitze:




T Werkzeugträgerbezugspunkt

P Werkzeugspitze

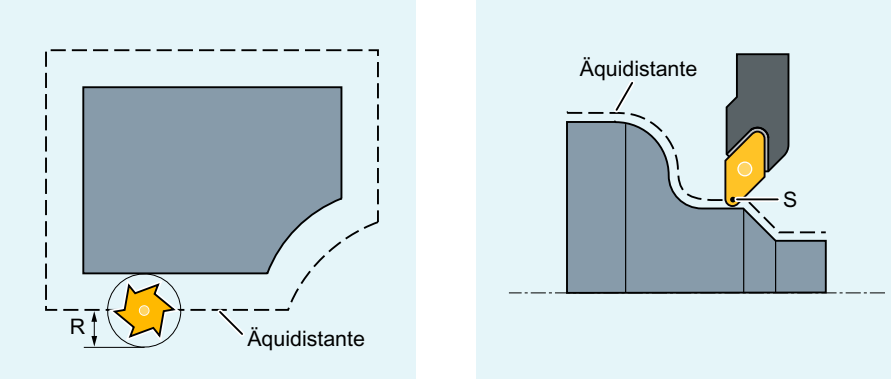
Diese Länge wird vermessen und zusammen mit vorgebbaren Verschleißwerten in den Werkzeug-Korrekturspeicher der Steuerung eingegeben. Hieraus errechnet die Steuerung die Verfahrbewegungen in Zustellrichtung.

**Hinweis**

Der Korrekturwert der Werkzeuglänge ist abhängig von der räumlichen Orientierung des Werkzeugs.

## 2.5.3 Werkzeugradiuskorrektur

Kontur und Werkzeugbahn sind nicht identisch. Der Fräser- bzw. Schneidenmittelpunkt muss entsprechend des Werkzeugradius auf einer Äquidistanten zur Kontur (Werkzeugmittelpunktsbahn) verfahren werden. Dazu wird von der Steuerung während der Abarbeitung des Programms die programmierte Werkzeugmittelpunktsbahn anhand des Werkzeugradius des aktiven Werkzeugs (Werkzeug-Korrekturspeicher) so verschoben, dass die Werkzeugschneide exakt an der programmierten Kontur verfahren wird.




R Werkzeugradius

S Schneidenmittelpunkt

Die Werkzeugradiuskorrektur ist ausführlich beschrieben in Kapitel "Werkzeugradiuskorrekturen (Seite 254)".

### Siehe auch

2 1/2 D-Werkzeugkorrektur (CUT2D, CUT2DD, CUT2DF, CUT2DFD) (Seite 289)

## 2.5.4 Werkzeug-Korrekturspeicher

Im Werkzeug-Korrekturspeicher der Steuerung müssen für jede Werkzeug-Schneide folgende Daten vorhanden sein:

- Werkzeugtyp
- Schneidenlage
- Geometrische Werkzeug-Größen (Länge, Radius)

Diese Daten werden als Werkzeug-Parameter (max. 25) eingetragen. Welche Parameter für ein Werkzeug benötigt werden, ist abhängig vom Werkzeugtyp. Nicht benötigte Werkzeug-Parameter sind mit dem Wert "Null" zu belegen (entspricht der Vorbelegung vom System).

**Hinweis**

Einmal in den Korrekturspeicher eingetragene Werte werden bei jedem Werkzeug-Aufruf mitverrechnet.

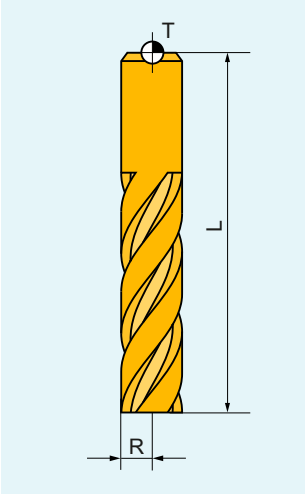
### Werkzeugtyp

Der Werkzeugtyp (Bohrer, Fräser oder Drehwerkzeuge) bestimmt, welche Geometrieangaben erforderlich sind und wie diese verrechnet werden.

### Schneidenlage

Die Schneidenlage beschreibt die Lage der Werkzeugspitze in Bezug auf den Schneidenmittelpunkt. Die Schneidenlage wird zusammen mit dem Schneidenradius zur Berechnung der Werkzeugradiuskorrektur bei Drehwerkzeugen (Werkzeugtyp 5xx) (Seite 77) benötigt.

### Geometrische Werkzeug-Größen (Länge, Radius)




| | |
|---|---|
| T | Werkzeugträgerbezugspunkt |
| R | Werkzeugradius |
| L | Werkzeuglänge |

Die geometrischen Werkzeug-Größen bestehen aus mehreren Komponenten (Geometrie, Verschleiß). Die Komponenten verrechnet die Steuerung zu einer resultierenden Größe (z. B. Gesamtlänge 1, Gesamtradius). Das jeweilige Gesamtmaß kommt bei Aktivierung des Korrekturspeichers zur Wirkung.

Wie diese Werte in den Achsen verrechnet werden, bestimmen der Werkzeugtyp und die aktuelle Ebene (G17/G18/G19).

## 2.5.5 Werkzeugtypen

### 2.5.5.1 Werkzeugtypnummer und Werkzeuggruppen

Jedem Werkzeugtyp ist eine eindeutige 3-stellige Nummer zugeordnet. Über die erste Ziffer (Hunderterstelle) erfolgt die Zuordnung des Werkzeugs zu einer der folgenden Technologien bzw. Werkzeuggruppen:

| Werkzeugtyp | Werkzeuggruppe |
|---|---|
| 1xy | Fräser (Seite 72) |
| 2xy | Bohrer (Seite 74) |
| 3xy | reserviert |
| 4xy | Schleifwerkzeuge (Seite 75) |
| 5xy | Drehwerkzeuge (Seite 77) |
| 6xy | reserviert |
| 7xy | Sonderwerkzeuge (Seite 79) |

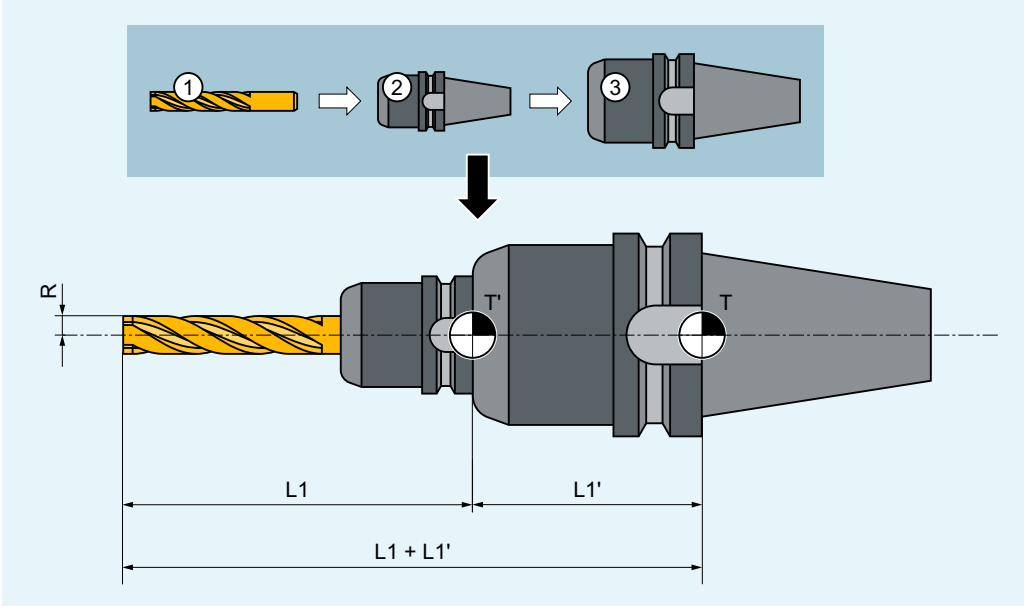
### 2.5.5.2 Fräswerkzeuge

Innerhalb der Werkzeuggruppe "Fräswerkzeuge" gibt es folgende Werkzeugtypen:

| | |
|---|---|
| 100 | Fräswerkzeug nach CLDATA (Cutter Location Data) |
| 110 | Kugelkopffräser |
| 111 | Zylindrischer Gesenkfräser |
| 120 | Schaftfräser ohne Eckenverrundung |
| 121 | Schaftfräser mit Eckenverrundung |
| 130 | Winkelkopffräser ohne Eckenverrundung |
| 131 | Winkelkopffräser mit Eckenverrundung |
| 140 | Planfräser |
| 145 | Gewindefräser |
| 150 | Scheibenfräser |
| 151 | Säge |
| 155 | Kegelstumpffräser ohne Eckenverrundung |
| 156 | Kegelstumpffräser mit Eckenverrundung |
| 157 | Kegeliger Gesenkfräser |
| 160 | Bohrgewindefräser |

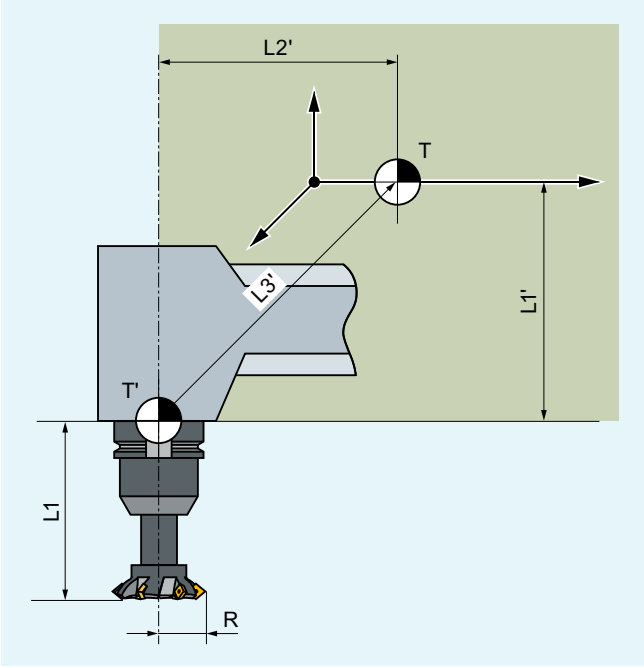
## Werkzeugparameter

Die folgenden Abbildungen geben einen Überblick, welche Werkzeugparameter bei Fräswerkzeugen in den Korrekturspeicher eingetragen werden:




| | |
|---|---|
| ① | Werkzeug |
| ② | Werkzeugaufnahme |
| ③ | Werkzeugadapter |
| T | Adapterbezugspunkt (bei eingestecktem Werkzeug = Werkzeugträgerbezugspunkt) |
| T' | Werkzeughalterbezugspunkt |
| L1 | Geometrie - Länge 1 |
| L1' | Adaptermaß - Länge 1 |
| L1 + L1' | Gesamtlänge L1 |
| R | Radius |

| Werkzeugparameter | Bedeutung |
|---|---|
| $TC_DP1 | Werkzeugtyp 1xy |
| $TC_DP3 | Geometrie - Länge 1 |
| $TC_DP6 | Geometrie - Radius |
| $TC_DP21 | Adaptermaß - Länge 1 |
| • Verschleißwerte entsprechend Erfordernis.<br>• Übrige Werte sind auf 0 zu setzen. | |




| | |
|---|---|
| T | Werkzeugträgerbezugspunkt |
| T' | Werkzeughalterbezugspunkt |
| L1 | Geometrie - Länge 1 |
| R | Werkzeugradius |
| L1' | Basismaß - Länge 1 |
| L2' | Basismaß - Länge 2 |
| L3' | Basismaß - Länge 3 |

| Werkzeugparameter | Bedeutung |
|---|---|
| $TC_DP1 | Werkzeugtyp |
| $TC_DP3 | Geometrie - Länge 1 |
| $TC_DP6 | Geometrie - Radius |
| $TC_DP21 | Basismaß - Länge 1 |
| $TC_DP22 | Basismaß - Länge 2 |
| $TC_DP23 | Basismaß - Länge 3 |
| • Verschleißwerte entsprechend Erfordernis.<br>• Übrige Werte sind auf 0 zu setzen. | |

### 2.5.5.3 Bohrer

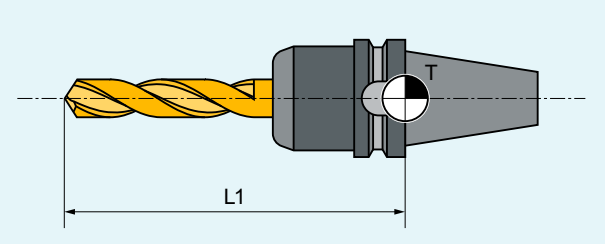
Innerhalb der Werkzeuggruppe "Bohrer" gibt es folgende Werkzeugtypen:

| Nr. | Werkzeugtyp |
|---|---|
| 200 | Spiralbohrer |
| 205 | Vollbohrer |
| 210 | Bohrstange |

| Nr. | Werkzeugtyp |
|---|---|
| 220 | Zentrierbohrer |
| 230 | Spitzsenker |
| 231 | Flachsenker |
| 240 | Gewindebohrer Regelgewinde |
| 241 | Gewindebohrer Feingewinde |
| 242 | Gewindebohrer Withworthgewinde |
| 250 | Reibahle |

### Werkzeugparameter

Die folgende Abbildung gibt einen Überblick, welche Werkzeugparameter bei Bohrern in den Korrekturspeicher eingetragen werden:




T Werkzeugträgerbezugspunkt

L1 Länge 1

| Werkzeugparameter | Bedeutung |
|---|---|
| $TC_DP1 | Werkzeugtyp |
| $TC_DP3 | Geometrie - Länge 1 |
| • Verschleißwerte entsprechend Erfordernis.<br>• Übrige Werte sind auf 0 zu setzen. | |

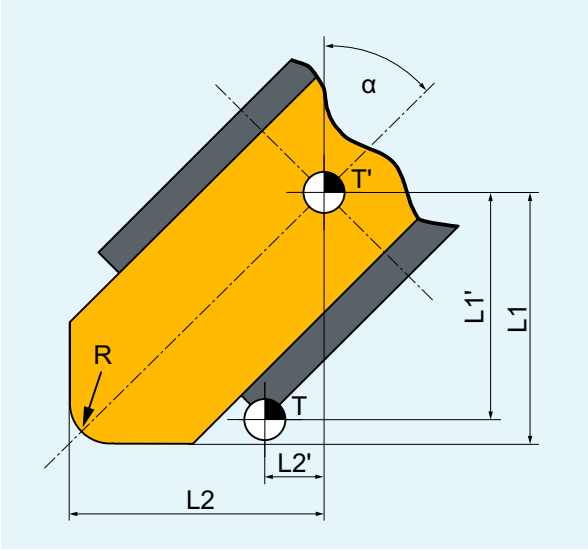
## 2.5.5.4 Schleifwerkzeuge

Innerhalb der Werkzeuggruppe "Schleifwerkzeuge" gibt es folgende Werkzeugtypen:

| | |
|---|---|
| 400 | Umfangsschleifscheibe |
| 401 | Umfangsschleifscheibe mit Überwachung |
| 402 | Umfangsschleifscheibe ohne Überwachung ohne Basismaß (WZV) |
| 403 | Umfangsschleifscheibe mit Überwachung ohne Basismaß für Schleifscheibenumfangsgeschwindigkeit SUG |
| 410 | Planscheibe |
| 411 | Planscheibe (WZV) mit Überwachung |
| 412 | Planscheibe (WZV) ohne Überwachung |
| 413 | Planscheibe mit Überwachung ohne Basismaß für Schleifscheibenumfangsgeschwindigkeit SUG |
| 490 | Abrichter |

## Werkzeugparameter

Die folgende Abbildung gibt einen Überblick, welche Werkzeugparameter bei Schleifwerkzeugen in den Korrekturspeicher eingetragen werden:




| | |
|---|---|
| T | Werkzeugträgerbezugspunkt |
| T' | Werkzeughalterbezugspunkt |
| L1 | Geometrie - Länge 1 |
| L1' | Basismaß - Länge 1 |
| L2 | Geometrie - Länge 2 |
| L2' | Basismaß - Länge 2 |
| R | Radius |
| α | Winkel der schrägen Scheibe |

| Schneidenspezifische Parameter | Bedeutung |
|---|---|
| $TC_DP1 | Werkzeugtyp 4xy |
| $TC_DP2 | Schneidenlage |
| $TC_DP3 | Geometrie Länge 1 |
| $TC_DP4 | Geometrie Länge 2 |
| $TC_DP6 | Radius |
| $TC_DP21 | Basismaß Länge 1 |
| $TC_DP22 | Basismaß Länge 2 |
| • Verschleißwerte entsprechend Erfordernis.<br>• Übrige Werte sind auf 0 zu setzen. | |

| Werkzeugspezifische Parameter | Bedeutung |
|---|---|
| $TC_TPG1 | Spindelnummer |
| $TC_TPG2 | Verkettungsvorschrift [1)] |
| $TC_TPG3 | Minimaler Scheibenradius |

| Werkzeugspezifische Parameter | Bedeutung |
|---|---|
| $TC_TPG4 | Minimale Scheibenbreite |
| $TC_TPG5 | Aktuelle Scheibenbreite |
| $TC_TPG6 | Maximale Drehzahl |
| $TC_TPG7 | Maximale Umfangsgeschwindigkeit |
| $TC_TPG8 | Winkel der schrägen Scheibe |
| $TC_TPG9 | Parameter-Nummer für Radiusberechnung |
| $TC_TPG_DRSPATH | Verzeichnispfad auf das Abrichtprogramm |
| $TC_TPG_DRSPROG | Abrichtprogrammname |

[1] Die Längenkorrekturen Geometrie, Verschleiß und Basismaß können jeweils für die linke und rechte Scheibenkorrektur verkettet werden. D. h., werden die Längenkorrekturen für die linke Schneide geändert, so werden die Werte automatisch auch für die rechte Schneide eingetragen und umgekehrt.

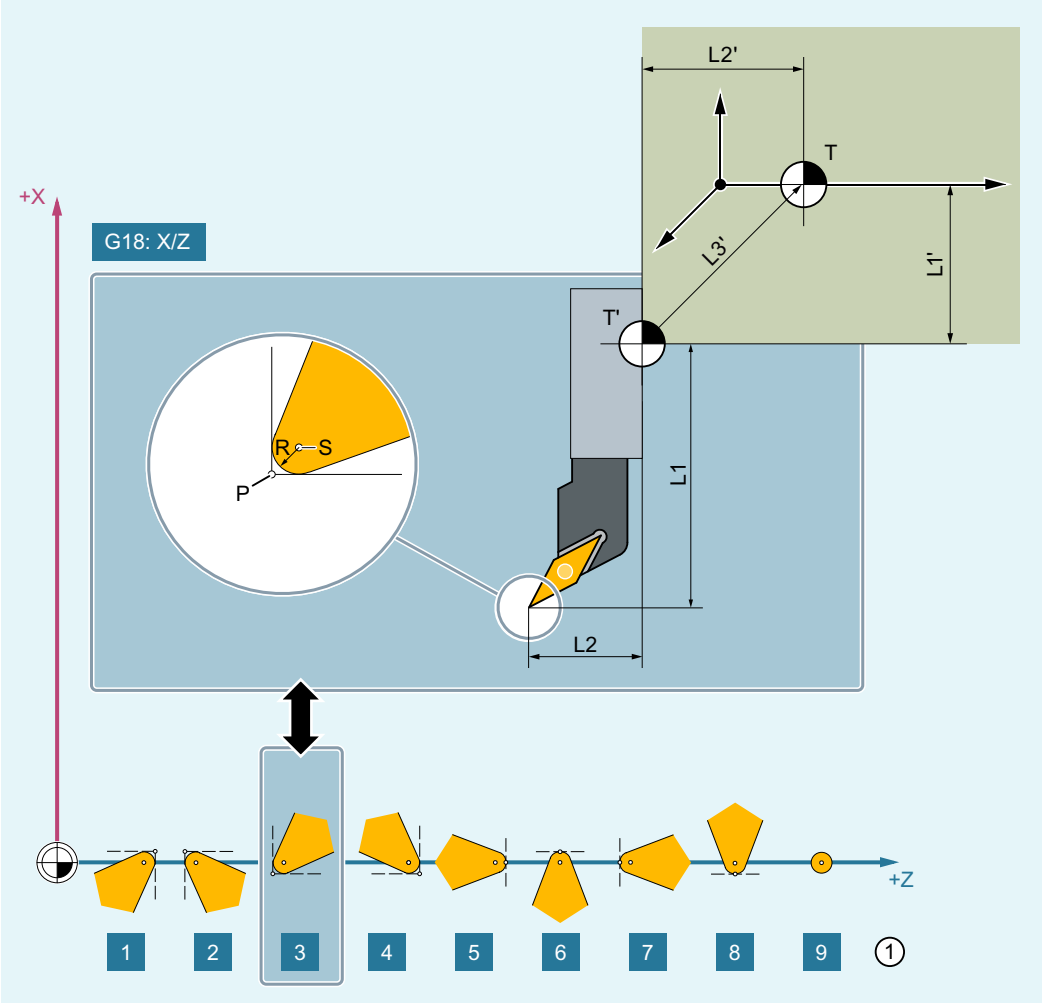
### 2.5.5.5 Drehwerkzeuge

Innerhalb der Werkzeuggruppe "Drehwerkzeuge" gibt es folgende Werkzeugtypen:

| | |
|---|---|
| 500 | Schruppstahl |
| 510 | Schlichtstahl |
| 520 | Einstechstahl |
| 530 | Abstechstahl |
| 540 | Gewindestahl |
| 550 | Pilzstahl / Formstahl (WZV) |
| 560 | Drehbohrer (ECOCUT) |
| 580 | Messtaster mit Parameter Schneidenlage |

## Werkzeugparameter

Die folgende Abbildung gibt einen Überblick, welche Werkzeugparameter bei Drehwerkzeugen in den Korrekturspeicher eingetragen werden:




| | |
|---|---|
| ① | Schneidenlage (1 - 9) bei Bearbeitung hinter der Drehmitte |
| P | Werkzeugspitze |
| S | Schneidenmittelpunkt |
| R | Schneidenradius |
| T | Werkzeugträgerbezugspunkt |
| T' | Werkzeughalterbezugspunkt |
| L1 | Geometrie - Länge 1 |
| L2 | Geometrie - Länge 2 |
| L1' | Basismaß - Länge 1 |
| L2' | Basismaß - Länge 2 |
| L3' | Basismaß - Länge 3 |

| Werkzeugparameter | Bedeutung |
|---|---|
| $TC_DP1 | Werkzeugtyp |
| $TC_DP2 | Schneidenlage |
| $TC_DP3 | Geometrie - Länge 1 |
| $TC_DP4 | Geometrie - Länge 2 |
| $TC_DP6 | Geometrie - Radius (Schneidenradius) |
| $TC_DP21 | Basismaß - Länge 1 |
| $TC_DP22 | Basismaß - Länge 2 |
| $TC_DP23 | Basismaß - Länge 3 |
| • Verschleißwerte entsprechend Erfordernis.<br>• Übrige Werte sind auf 0 zu setzen. | |

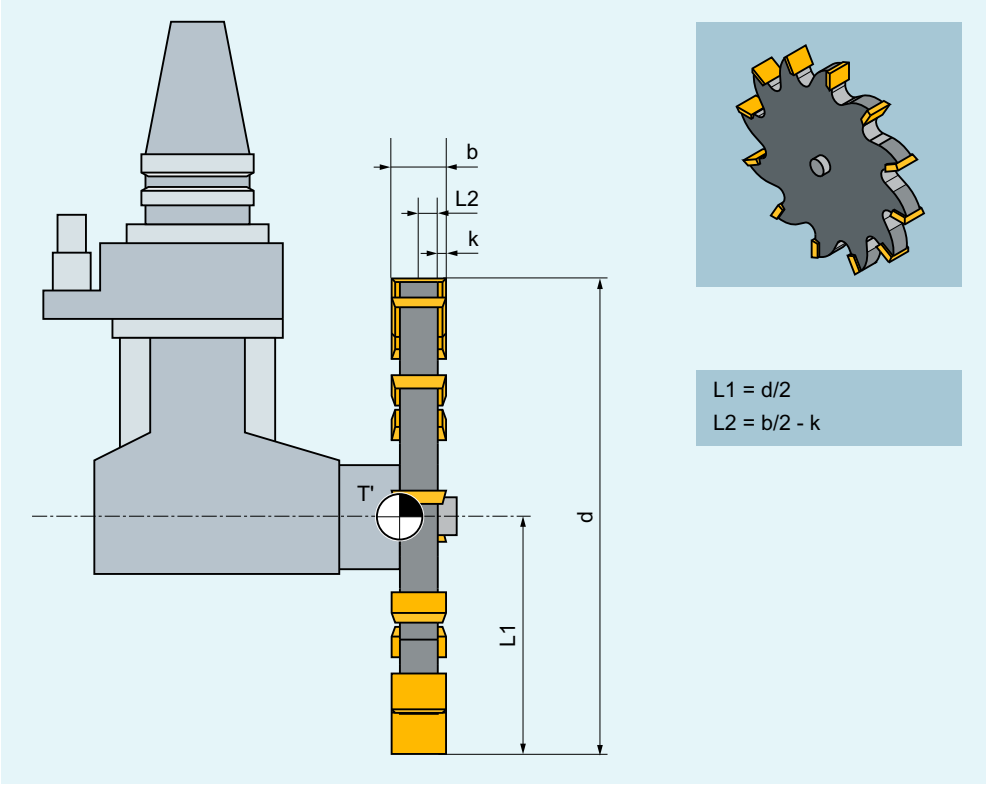
### 2.5.5.6 Sonderwerkzeuge

Innerhalb der Werkzeuggruppe "Sonderwerkzeuge" gibt es folgende Werkzeugtypen:

| | |
|---|---|
| 700 | Nutsäge |
| 710 | 3D-Messtaster |
| 711 | Kantentaster |
| 712 | Monotaster |
| 713 | L-Taster |
| 714 | Sterntaster |
| 725 | Kalibrierwerkzeug |
| 730 | Anschlag |
| 731 | Pinole |
| 732 | Lünette |

## Werkzeugparameter

Die folgende Abbildung gibt einen Überblick, welche Werkzeugparameter beim Werkzeugtyp "Nutsäge" in den Korrekturspeicher eingetragen werden:




T' Werkzeughalterbezugspunkt
L1 Geometrie - Länge 1
L2 Geometrie - Länge 2
d Durchmesser
b Nutbreite
k Überstand

| Werkzeugparameter | Bedeutung |
|---|---|
| $TC_DP1 | Werkzeugtyp |
| $TC_DP3 | Geometrie - Länge 1 |
| $TC_DP4 | Geometrie - Länge 2 |
| $TC_DP6 | Durchmesser |
| $TC_DP7 | Nutbreite |
| $TC_DP8 | Überstand |
| $TC_DP21 | Basismaß Länge 1 |
| $TC_DP22 | Basismaß Länge 2 |
| $TC_DP23 | Basismaß Länge 3 |
| • Verschleißwerte entsprechend Erfordernis.<br>• Übrige Werte sind auf 0 zu setzen. | |

## 2.5.6 Werkzeugkorrekturen aktivieren/deaktivieren (D, D0)

Den 1 bis 8 (bei aktiver WZV 12) Schneiden eines Werkzeugs können verschiedene Werkzeugkorrekturdatensätze zugeordnet werden (z. B. unterschiedliche Korrekturwerte für die linke und rechte Schneide bei einem Einstechmeißel).

Die Aktivierung der Korrekturdaten (u. a. die Daten für die Werkzeuglängenkorrektur) einer speziellen Schneide erfolgt durch Aufruf der D-Nummer. Bei Programmierung von D0 sind die Korrekturen für das Werkzeug unwirksam.

Eine Werkzeugradiuskorrektur muss zusätzlich durch G41/G42 eingeschaltet werden.

**Hinweis**

Werkzeuglängenkorrekturen wirken, wenn die D-Nummer programmiert ist. Wird keine D-Nummer programmiert, ist bei einem Werkzeugwechsel die über Maschinendatum definierte Standardeinstellung aktiv (→ siehe Angaben des Maschinenherstellers).

### Syntax

```
D<Nr>
X... Y... Z...
G41/G42 X... Y... Z...
G40
D0
```

## Bedeutung

| | | |
|---|---|---|
| D: | Adresse zur Aktivierung eines Korrekturdatensatzes für das aktive Werkzeug<br>Die Werkzeuglängenkorrektur wird mit dem ersten programmierten Verfahren der zugehörigen Längenkorrekturachse herausgefahren.<br>**Achtung:**<br>Eine Werkzeuglängenkorrektur wirkt auch ohne D-Programmierung, wenn für den Werkzeugwechsel die automatische Aktivierung einer Werkzeugschneide projektiert ist (→ siehe Angaben des Maschinenherstellers). | |
| <Nr>: | Über die D-Nummer wird der zu aktivierende Werkzeugkorrekturdatensatz angegeben.<br>Die Art der D-Programmierung ist abhängig von der Projektierung der Maschine (→ siehe Angaben des Maschinenherstellers).<br>Es gibt folgende Möglichkeiten:<br>• D-Nummer = Schneidennummer<br>Zu jedem Werkzeug T<Nr> (ohne WZV) bzw. T="Name" (mit WZV) existieren D-Nummern von 1 bis max. 12. Diese D-Nummern sind direkt den Schneiden von Werkzeugen zugeordnet. Zu jeder D-Nummer (= Schneidennummer) gehört ein Korrekturdatensatz ($TC_DPx[<t>,<d>]).<br>• Freie Wahl von D-Nummern<br>Die D-Nummern können den Schneidennummern eines Werkzeugs frei zugeordnet werden. Die Obergrenze der verwendbaren D-Nummern ist durch ein Maschinendatum festgelegt. | |
| | Wertebereich: | 0 - 32000 |
| D0: | Deaktivierung des Korrekturdatensatzes für das aktive Werkzeug | |
| G41: | Befehl zum Einschalten der Werkzeugradiuskorrektur mit Bearbeitungsrichtung **links** von der Kontur | |
| G42: | Befehl zum Einschalten der Werkzeugradiuskorrektur mit Bearbeitungsrichtung **rechts** von der Kontur | |
| G40: | Befehl zum Ausschalten der Werkzeugradiuskorrektur | |

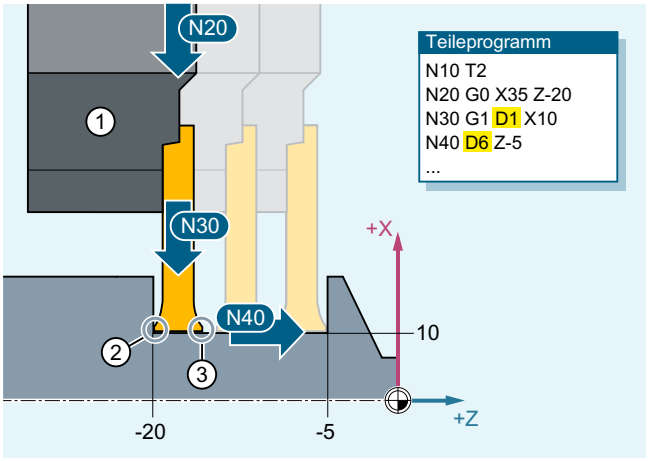
**Weitere Informationen** zu G40/G41/G42 siehe Kapitel "Werkzeugradiuskorrekturen (Seite 254)".

## Beispiele

### Beispiel 1: Werkzeugwechsel mit T-Befehl (Drehen)

| Programmcode | Kommentar |
|---|---|
| N10 T1 D1 | ; Werkzeug T1 einwechseln und den Werkzeugkorrekturdatensatz D1 von T1 aktivieren. |
| N11 G0 X... Z... | ; Die Längenkorrekturen werden herausgefahren. |
| N50 T4 D2 | ; Werkzeug T4 einwechseln und den Werkzeugkorrekturdatensatz D2 von T4 aktivieren. |
| ... | |
| N70 G0 Z... D1 | ; Andere Schneide D1 für das Werkzeug T4 aktivieren. |

**Beispiel 2: Unterschiedliche Korrekturwerte für die linke und rechte Schneide bei einem Einstechmeißel**




① Einstechdrehmeißel (T2)
② Schneide D1
③ Schneide D6

## Siehe auch

Werkzeugradiuskorrektur (Seite 69)

## Weitere Informationen

### Änderung der Werkzeugkorrekturdaten

In der Standardeinstellung werden Änderungen der Werkzeugkorrekturdaten erst nach erneuter T- oder D-Programmierung wirksam.

Über das folgende Maschinendatum kann festgelegt werden, dass eingegebene Werkzeugkorrekturdaten sofort wirksam gesetzt werden:

MD9440 $MM_ACTIVATE_SEL_USER

**⚠ WARNUNG**

**Kollisionsgefahr**

Wenn MD9440 gesetzt ist, dann werden Werkzeugkorrekturen, die sich aus Änderungen von Werkzeugkorrekturdaten während des Teileprogramm-Stopps ergeben, mit dem Fortsetzen des Teileprogramms herausgefahren.

## 2.5.7 Werkzeugkorrekturen unterdrücken (SUPD)

Mit dem Befehl SUPD können die Werkzeugkorrekturen satzweise unterdrückt werden.

Vorteil gegenüber der Werkzeugkorrekturabwahl mit D0 ist, dass bei SUPD die Werkzeugkorrekturinformation erhalten bleibt und dadurch die erneute Aktivierung der Werkzeugkorrekturdaten durch Programmierung der D-Nummer entfällt.

Anwendung findet SUPD ausschließlich zur Unterdrückung der Werkzeuglängenkorrektur, z. B. in Teileprogrammsätzen, in denen der Nullpunkt, aber nicht die Werkzeuglänge verrechnet werden soll. Eine Anwendung bei aktivem G41/G42 zur zusätzlichen Unterdrückung der Werkzeugradiuskorrekturen ist nicht zu empfehlen.

## Syntax

```
D<Nr.>
X... Y... Z...
X... Y... Z... SUPD
...
D0
```

## Bedeutung

| | | |
|---|---|---|
| SUPD: | G-Befehl zur Deaktivierung des Korrekturdatensatzes für das aktive Werkzeug im aktiven Satz | |
| | Wirksamkeit: | satzweise |

## Randbedingungen

- SUPD sollte nur in Linearsätzen verwendet werden.
- SUPD kann nicht in Synchronaktionen verwendet werden.
- SUPD kann nicht zusammen mit folgenden Funktionen verwendet werden:
  - 3D-Werkzeugradiuskorrektur für das 3D-Stirnfräsen (CUT3DFxx)
  - Kurventabellen (CTAB)
- Die Anwendung von SUPD bei aktiver Werkzeugradiuskorrektur (G41/G42) ist möglich, wird aber nicht empfohlen.
  Soll die Funktion dennoch verwendet werden, muss das folgene Settingdatum auf 0 gesetzt werden:
  SD42480 $SC_STOP_CUTCOM_STOPRE = 0
  Dadurch wird verhindert, dass es bei aktivem G41/G42 zu einer Programmunterbrechung kommt.

## Beispiel

Im Unterprogramm SUB_SUP soll beim Verfahren die Werkzeuglänge unterdrückt werden.

### Teileprogramm

| **Programmcode** | **Kommentar** |
|---|---|
| ... | |

| Programmcode | Kommentar |
|---|---|
| N300 $P_UIFR[1]=CTRANS(X,1000,Y,400,Z,-120) | |
| N310 T="BALL_D3" | |
| N320 M6 | |
| N330 TRAFOOF | |
| N340 G54 G0 Z49 D1 | |
| N350 G0 X1100 Y500 C0 A0 | |
| N360 **SUB_SUP** | ; Aufruf des Unterprogramms. |
| N370 G0 X1200 | |
| N380 M30 | |

### Unterprogramm mit D0

| Programmcode | Kommentar |
|---|---|
| N10 PROC **SUB_SUP** | |
| N20 DEF INT NUMBER | |
| N30 NUMBER=$P_TOOL | |
| N40 G0 Z49 **D0** | ; Abwahl der Werkzeugkorrekturen. |
| N50 D=NUMBER | ; Erneute Anwahl der Werkzeugkorrekturen. |
| N60 RET | |

### Unterprogramm mit SUPD

| Programmcode | Kommentar |
|---|---|
| N10 PROC **SUB_SUP** | |
| N40 G0 Z49 **SUPD** | ; Unterdrückung der Werkzeugkorrekturen im aktiven Satz. |
| N60 RET | |

## 2.5.8 Programmierbarer Werkzeugkorrektur-Offset (TOFFL, TOFF, TOFFR, TOFFLR)

Über die Adressen TOFFx hat der Anwender die Möglichkeit, die effektive Werkzeuglänge und den effektiven Werkzeugradius im NC-Programm zu modifizieren, ohne die im Korrekturspeicher abgelegten Werkzeugkorrekturdaten zu verändern.

Mit dem Programmende werden diese programmierten Offsets wieder gelöscht.

## Syntax

### Werkzeuglängen-Offset

```
TOFFL=<Value>
TOFFL[1]=<Value> TOFFL[2]=<Value> TOFFL[3]=<Value>
TOFF[<GeoAx>]=<Value>
```

Die Werkzeuglänge kann in allen drei Komponenten gleichzeitig verändert werden. Es dürfen in einem Satz aber nicht gleichzeitig Befehle der Gruppe TOFFL/TOFFL[1..3] einerseits und

der Gruppe TOFF[<GeoAx>] andererseits verwendet werden. Ebenso dürfen in einem Satz nicht gleichzeitig TOFFL und TOFFL[1] geschrieben werden.

Werden in einem Satz nicht alle drei Werkzeuglängenkomponenten programmiert, so bleiben die nicht programmierten Komponenten unverändert. Dadurch ist es möglich, Korrekturen für mehrere Komponenten satzweise aufzubauen. Dies gilt jedoch nur, solange die Werkzeugkomponenten entweder nur mit TOFFL oder nur mit TOFF modifiziert werden. Ein Wechsel der Programmiervariante TOFFL nach TOFF oder umgekehrt löscht zunächst alle evtl. zuvor programmierten Werkzeuglängen-Offsets (siehe Beispiel 3).

#### Werkzeugradius-Offset

```
TOFFR=<Value>
```

#### Simultaner Werkzeuglängen- und Werkzeugradius-Offset

```
TOFFLR=<Value>
```

## Bedeutung

| | | |
|---|---|---|
| TOFFL: | Korrektur der effektiven Werkzeuglänge<br>TOFFL kann mit oder ohne Index programmiert werden: | |
| | TOFFL=... | Der programmierte Offset-Wert wirkt in der Richtung, in der auch die im Korrekturspeicher abgelegte Werkzeuglängenkomponente **L1** wirkt.<br>Die Anweisungen TOFFL und TOFFL[1] sind in ihrer Wirkung identisch. |
| | TOFFL[1]=...<br>TOFFL[2]=...<br>TOFFL[3]=... | Der programmierte Offset-Wert wirkt in der Richtung, in der auch die im Korrekturspeicher abgelegte Werkzeuglängenkomponente **L1**, **L2** bzw. **L3** wirkt. |
| | **Hinweis:**<br>Wie die Werkzeuglängen-Korrekturwerte in den Achsen verrechnet werden, bestimmen der Werkzeugtyp und die aktuelle Arbeitsebene (G17/G18/G19). | |
| TOFF: | Korrektur der Werkzeuglänge in der Komponente parallel zu der angegebenen Geometrieachse<br>TOFF wirkt in Richtung der Werkzeuglängenkomponente, die bei nicht gedrehtem Werkzeug (orientierbarer Werkzeugträger bzw. Orientierungstransformation) parallel zu der im Index angegebenen Geometrieachse wirkt.<br>**Hinweis:**<br>Ein Frame beeinflusst die Zuordnung der programmierten Werte zu den Werkzeuglängenkomponenten nicht. D. h., für die Zuordnung der Werkzeuglängenkomponenten zu den Geometrieachsen wird nicht das Werkstückkoordinatensystem (WKS), sondern das Werkzeugkoordinatensystem in Werkzeuggrundstellung zu Grunde gelegt. | |
| <GeoAx>: | Bezeichner der Geometrieachse | |
| TOFFR: | Korrektur des effektiven Werkzeugradius<br>TOFFR verändert den effektiven Werkzeugradius **bei aktiver Werkzeugradiuskorrektur** um den programmierten Offset-Wert. | |

| | | |
|---|---|---|
| TOFFLR: | Korrektur der effektiven Werkzeuglänge in der Komponente L1 **und** des effektiven Werkzeugradius<br>**Hinweis:**<br>Bei Werkzeugen mit Eckenverrundung (Typen 111, 121, 131 und 156) korrigiert TOFFLR auch den Eckenradius. | |
| <Value>: | Offset-Wert | |
| | Typ: | REAL |

## Beispiele

### Beispiel 1: Positiver Werkzeuglängen-Offset

Das aktive Werkzeug sei ein Bohrer mit der Länge L1 = 100 mm.

Die aktive Ebene sei G17. D. h., der Bohrer zeigt in Z-Richtung.

Die effektive Bohrerlänge soll um 1 mm verlängert werden. Für die Programmierung dieses Werkzeuglängen-Offsets stehen folgende Varianten zur Verfügung:

- `TOFFL=1`
- `TOFFL[1]=1`
- `TOFF[Z]=1`

### Beispiel 2: Negativer Werkzeuglängen-Offset

Das aktive Werkzeug sei ein Bohrer mit der Länge L1 = 100 mm.

Die aktive Ebene sei G18. D. h., der Bohrer zeigt in Y-Richtung.

Die effektive Bohrerlänge soll um 1 mm verkürzt werden. Für die Programmierung dieses Werkzeuglängen-Offsets stehen folgende Varianten zur Verfügung:

- `TOFFL=-1`
- `TOFFL[1]=-1`
- `TOFF[Y]=1`

### Beispiel 3: Wechsel der Programmiervariante von TOFFL nach TOFF

Das aktive Werkzeug sei ein Fräswerkzeug. Die aktive Ebene sei G17.

```
Programmcode                        Kommentar
N10 TOFFL[1]=3 TOFFL[3]=5           ; Wirksame Offsets: L1=3, L2=0, L3=5
N20 TOFFL[2]=4                      ; Wirksame Offsets: L1=3, L2=4, L3=5
N30 TOFF[Z]=1.3                     ; Wirksame Offsets: L1=0, L2=0, L3=1.3
```

### Beispiel 4: Zuordnung der Offset-Werte nach Ebenenwechsel

```
Programmcode              Kommentar
N10 $TC_DP1[1,1]=120
N20 $TC_DP3[1,1]=100      ; Werkzeuglänge L1=100mm
N30 T1 D1 G17
N40 TOFF[Z]=1.0           ; Offset in Z-Richtung (entspricht L1 bei G17).
```

| Programmcode | Kommentar |
|---|---|
| N50 G0 X0 Y0 Z0 | ; Maschinenachsposition X0 Y0 Z101 |
| N60 G18 G0 X0 Y0 Z0 | ; Maschinenachsposition X0 Y100 Z1 |
| N70 G17 | |
| N80 **TOFFL=1.0** | ; Offset in L1-Richtung (entspricht Z bei G17). |
| N90 G0 X0 Y0 Z0 | ; Maschinenachsposition X0 Y0 Z101. |
| N100 G18 G0 X0 Y0 Z0 | ; Maschinenachsposition X0 Y101 Z0. |

In diesem Beispiel bleibt beim Wechsel nach G18 im Satz N60 der Offset von 1 mm in der Z–Achse erhalten, die effektive Werkzeuglänge in der Y–Achse ist die unveränderte Werkzeuglänge von 100 mm.

Im Satz N100 wirkt der Offset beim Wechsel nach G18 dagegen in der Y–Achse, weil er bei der Programmierung der Werkzeuglänge L1 zugeordnet wurde, und diese Längenkomponente bei G18 in der Y-Achse wirkt.

### Beispiel 5: Simultaner Werkzeuglängen- und Werkzeugradius-Offset

a, Schaftfräser **ohne** Eckenverrundung (WZ-Typ 120):

| Programmcode | Kommentar |
|---|---|
| ... | |
| TOFFLR=0.1 | ; Wirksame Offsets: |
| | ; Offset Werkzeuglänge (L1) = 5 |
| | ; Offset Werkzeugradius = 5 |
| ... | |

b, Schaftfräser **mit** Eckenverrundung (WZ-Typ 121):

| Programmcode | Kommentar |
|---|---|
| ... | |
| TOFFLR=0.1 | ; Wirksame Offsets: |
| | ; Offset Werkzeuglänge (L1) = 0.1 |
| | ; Offset Werkzeugradius = 0.1 |
| | ; Offset Eckenradius = 0.1 |
| ... | |

## Weitere Informationen

### Werkzeuglängen-Offsets

Programmierte Werkzeuglängen-Offsets werden abhängig von der Art der Programmierung entweder den im Korrekturspeicher abgelegten Werkzeuglängenkomponenten L1, L2 und L3 (TOFFL) oder den Geometrieachsen (TOFF) zugeordnet. Entsprechend werden die programmierten Offsets bei einem Ebenenwechsel (G17/G18/G19 ↔ G17/G18/G19) behandelt:

- Sind die Offset-Werte den Werkzeuglängenkomponenten zugeordnet, werden die Richtungen, in welchen die programmierten Offsets wirken, entsprechend getauscht.
- Sind die Offset-Werte den Geometrieachsen zugeordnet, beeinflusst ein Ebenenwechsel die Zuordnung in Bezug auf die Koordinatenachsen nicht.

Bei der Zuordnung der programmierten Offset-Werte zu den Werkzeuglängenkomponenten werden folgende Settingdaten ausgewertet:

SD42940 $SC_TOOL_LENGTH_CONST (Wechsel der Werkzeuglängenkomponenten bei Ebenenwechsel)

SD42950 $SC_TOOL_LENGTH_TYPE (Zuordnung der Werkzeuglängenkompensation unabhängig vom Werkzeugtyp)

Haben diese Settingdaten gültige Werte ungleich 0, dann haben diese Vorrang vor dem Inhalt der G-Gruppe 6 (Ebenenanwahl G17/G18/G19) bzw. dem in den Werkzeugdaten enthaltenen Werkzeugtyp ($TC_DP1[<T–Nr.>, <D–Nr.>]). D. h., diese Settingdaten beeinflussen die Bewertung der Offsets in gleicher Weise wie die Werkzeuglängenkomponenten L1 bis L3.

### Werkzeugradius-Offset

Die Adresse TOFFR hat fast die gleiche Wirkung wie die Adresse OFFN (Seite 254). Ein Unterschied ergibt sich lediglich bei aktiver Mantelkurventransformation (TRACYL) und aktiver Nutwandkorrektur. In diesem Fall wirkt OFFN mit negativem Vorzeichen auf den Werkzeugradius, TOFFR dagegen mit positivem Vorzeichen.

OFFN und TOFFR können gleichzeitig wirksam sein. Sie wirken dann in der Regel additiv (außer bei Nutwandkorrektur).

### Werkzeugwechsel

Alle Offset-Werte bleiben bei einem Werkzeugwechsel (Schneidenwechsel) erhalten. D. h., sie werden bei dem neuen Werkzeug (der neuen Schneide) ebenfalls wirksam.

### Systemvariablen zum Lesen der aktuellen Offset-Werte

Die aktuell wirksamen Offsets können mit den folgenden Systemvariablen gelesen werden:

| Systemvariable | Bedeutung |
|---|---|
| $P_TOFFL [<n>]<br>mit 0 ≤ n ≤ 3 | Liest den aktuellen Offset-Wert von TOFFL (bei n = 0) bzw. TOFFL[1...3] (bei n = 1, 2, 3) im Vorlaufkontext. |
| $P_TOFF [<GeoAx>] | Liest den aktuellen Offset-Wert von TOFF[<GeoAx>] im Vorlaufkontext. |
| $P_TOFFR | Liest den aktuellen Offset-Wert von TOFFR im Vorlaufkontext. |
| $P_TOFFCR | Liest den aktuellen Offset-Wert des Eckenradius im Vorlaufkontext. |
| $AC_TOFFL[<n>]<br>mit 0 ≤ n ≤ 3 | Liest den aktuellen Offset-Wert von TOFFL (bei n = 0) bzw. TOFFL[1...3] (bei n = 1, 2, 3) im Hauptlaufkontext (Synchronaktionen). |
| $AC_TOFF[<GeoAx>] | Liest den aktuellen Offset-Wert von TOFF[<GeoAx>] im Hauptlaufkontext (Synchronaktionen). |
| $AC_TOFFR | Liest den aktuellen Offset-Wert von TOFFR im Hauptlaufkontext (Synchronaktionen). |
| $AC_TOFFCR | Liest den aktuellen Offset-Wert des Eckenradius im Hauptlaufkontext (Synchronaktionen). |

### Hinweis

Die Systemvariablen $AC_TOFFL, $AC_TOFF, AC_TOFFR und AC_TOFFCR lösen beim Lesen aus dem Vorlaufkontext (NC-Programm) einen automatischen Vorlaufstopp aus.

### Anwendungen

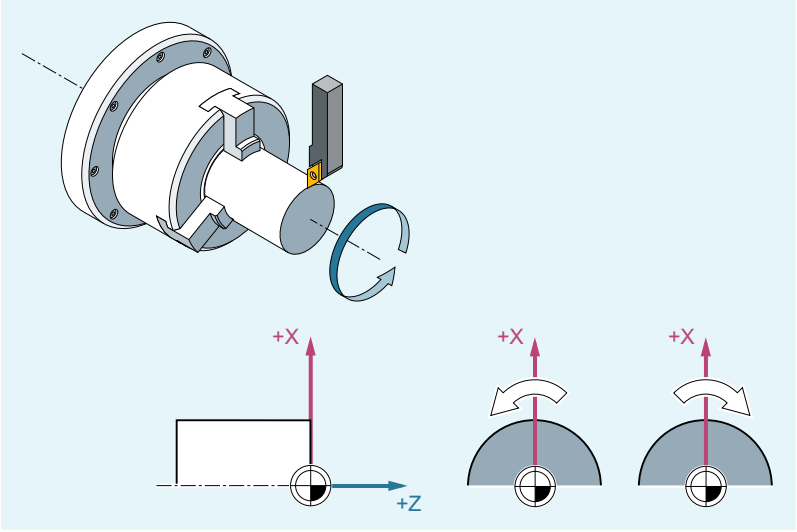
Die Funktion "Programmierbarer Werkzeugkorrektur-Offset" ist speziell für Kugelfräser und Fräser mit Eckenradien interessant, da diese im CAM-System oft auf Kugelmitte statt auf Kugelspitze berechnet werden. Bei der Messung des Werkzeugs wird aber in der Regel die Werkzeugspitze vermessen und als Werkzeuglänge im Korrekturspeicher hinterlegt.

Für die 3D-Werkzeugradiuskorrektur bei Verwendung eines Kugelfräsers ist es sinnvoll, gleichzeitig Werkzeuglänge und -radius um den gleichen Wert zu korrigieren. Dafür steht dem Anwender die Adresse TOFFLR zur Verfügung.

# 2.6 Spindelbewegung

## 2.6.1 Spindeldrehzahl (S), Spindeldrehrichtung (M3, M4, M5)

Die Angaben Spindeldrehzahl und -drehrichtung versetzen die Spindel in eine Drehbewegung und schaffen die Voraussetzung für die spanabhebende Bearbeitung.




Neben der Hauptspindel können weitere Spindeln vorhanden sein (z. B. bei Drehmaschinen die Gegenspindel oder ein angetriebenes Werkzeug). In der Regel wird die Hauptspindel per Maschinendatum als Masterspindel deklariert. Diese Zuweisung kann per NC-Befehl geändert werden.

### Syntax

`S... / S<n>=...`

`M3 / M<n>=3`

`M4 / M<n>=4`

`M5 / M<n>=5`

| | |
|---|---|
| `SETMS(<n>)` | |
| `...` | |
| `SETMS` | |

### Bedeutung

| | |
|---|---|
| `S...`: | Spindeldrehzahl in Umdrehungen/min für die Masterspindel |
| `S<n>=...`: | Spindeldrehzahl in Umdrehungen/min für Spindel <n> |

| | |
|---|---|
| | **Hinweis:**<br>Die mit S0=… angegebene Drehzahl gilt für die Masterspindel. |
| M3: | Spindeldrehrichtung rechts für Masterspindel |
| M<n>=3: | Spindeldrehrichtung rechts für Spindel <n> |
| M4: | Spindeldrehrichtung links für Masterspindel |
| M<n>=4: | Spindeldrehrichtung links für Spindel <n> |
| M5: | Spindel-Halt für Masterspindel |
| M<n>=5: | Spindel-Halt für Spindel <n> |
| SETMS(<n>): | Spindel <n> soll als Masterspindel gelten |
| SETMS: | SETMS ohne Spindelangabe schaltet auf die projektierte Masterspindel zurück |

**Hinweis**

Pro NC-Satz dürfen maximal 3 S-Werte programmiert werden, z. B.:

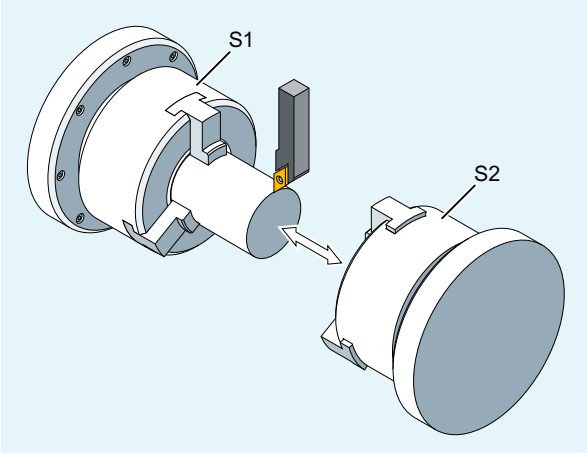
```
S... S2=... S3=...
```

**Hinweis**

SETMS muss in einem eigenen Satz stehen.

## Beispiel

S1 ist Masterspindel, S2 ist zweite Arbeitsspindel. Das Drehteil soll von 2 Seiten bearbeitet werden. Hierfür ist eine Aufteilung der Arbeitsschritte notwendig. Nach dem Abstechen nimmt die Synchroneinrichtung (S2) das Werkstück für die abstichseitige Bearbeitung auf. Hierzu wird diese Spindel S2 als Masterspindel definiert, für sie gilt dann G95.




| **Programmcode** | **Kommentar** |
|---|---|
| N10 S300 M3 | ; Drehzahl und Drehrichtung für Antriebsspindel = voreingestellte Masterspindel. |
| ... | ; Bearbeitung der rechten Werkstückseite. |
| N100 SETMS(2) | ; S2 ist jetzt Masterspindel. |
| N110 S400 G95 F… | ; Drehzahl für neue Masterspindel. |

| Programmcode | Kommentar |
|---|---|
| ... | ; Bearbeitung der linken Werkstückseite. |
| N160 SETMS | ; Zurückschalten auf Masterspindel S1. |

## Weitere Informationen

### Interpretation des S-Werts bei der Masterspindel

Ist in der G-Gruppe 1 (modal wirksame Bewegungsbefehle) die Funktion G331 oder G332 aktiv, wird der programmierte S-Wert immer als Drehzahl in Umdrehungen/min interpretiert. Andernfalls ist die Interpretation des S-Werts abhängig von der G-Gruppe 15 (Vorschubtyp): Bei aktivem G96, G961 oder G962 wird der S-Wert als konstante Schnittgeschwindigkeit in m/min interpretiert, in allen anderen Fällen als Drehzahl in Umdrehungen/min.

Bei einem Wechsel von G96/G961/G962 auf G331/G332 wird der Wert der konstanten Schnittgeschwindigkeit auf Null gesetzt, bei einem Wechsel von G331/G332 auf eine Funktion innerhalb der G-Gruppe 1 ungleich G331/G332 wird der Drehzahlwert auf Null gesetzt. Die betreffenden S-Werte müssen bei Bedarf neu programmiert werden.

### Voreingestellte M-Befehle M3, M4, M5

In einem Satz mit Achsbefehlen werden die Funktionen M3, M4, M5 eingeschaltet **bevor** die Achsbewegungen starten (Grundeinstellung der Steuerung).

Beispiel:

| Programmcode | Kommentar |
|---|---|
| N10 G1 F500 X70 Y20 S270 M3 | ; Die Spindel läuft auf 270 U/min hoch, dann werden die Bewegungen in X und Y ausgeführt. |
| N100 G0 Z150 M5 | ; Spindel-Halt vor der Rückzugsbewegung in Z. |

### Hinweis

Über Maschinendatum ist einstellbar, ob die Achsbewegungen erst nach Spindelhochlauf auf Solldrehzahl bzw. Spindelstopp ausgeführt werden oder ob sofort nach den programmierten Schaltvorgängen verfahren wird.

### Arbeiten mit mehreren Spindeln

In einem Kanal können gleichzeitig 5 Spindeln vorhanden sein (Masterspindel plus 4 zusätzliche Spindeln).

Eine Spindel wird per Maschinendatum als **Masterspindel** definiert. Für diese Spindel gelten spezielle Funktionen wie z. B. Gewindeschneiden, Gewindebohren, Umdrehungsvorschub, Verweilzeit. Für die übrigen Spindeln (z. B. eine zweite Arbeitsspindel und ein angetriebenes Werkzeug) müssen bei Drehzahl und Drehrichtung/Spindelstopp die entsprechenden Nummern angegeben werden.

Beispiel:

| Programmcode | Kommentar |
|---|---|
| N10 S300 M3 S2=780 M2=4 | ; Masterspindel: 300 U/min, Rechtslauf<br>2.Spindel: 780 U/min, Linkslauf |

### Programmierbares Umschalten der Masterspindel

Über den Befehl `SETMS(<n>)` kann im NC-Programm jede Spindel als Masterspindel definiert werden. `SETMS` muss in einem eigenen Satz stehen.

Beispiel:

| Programmcode | Kommentar |
|---|---|
| `N10 SETMS(2)` | `; Spindel 2 ist jetzt Masterspindel.` |

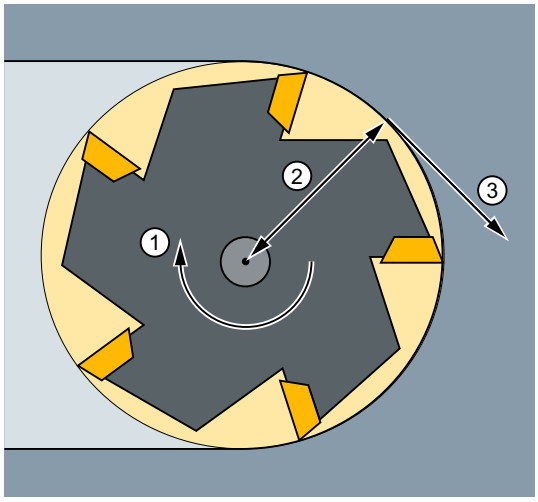
**Hinweis**

Für die neu deklarierte Masterspindel gelten jetzt die mit `S...` angegebene Drehzahl sowie die mit `M3`, `M4`, `M5` programmierten Funktionen.

Mit `SETMS` ohne Spindelangabe wird auf die im Maschinendatum festgelegte Masterspindel zurückgeschaltet.

## 2.6.2 Werkzeug-Schnittgeschwindigkeit (SVC)

Alternativ zur Spindeldrehzahl kann bei Fräsbearbeitungen auch die in der Praxis gebräuchlichere Werkzeug-Schnittgeschwindigkeit programmiert werden.




① Spindeldrehzahl
② Werkzeugradius
③ Werkzeug-Schnittgeschwindigkeit

Über den Radius des aktiven Werkzeugs berechnet die Steuerung aus der programmierten Werkzeug-Schnittgeschwindigkeit die wirksame Spindeldrehzahl:

$S = (SVC * 1000) / (R_{WKZ} * 2\pi)$

mit: S: Spindeldrehzahl in U/min
SVC: Werkzeug-Schnittgeschwindigkeit in m/min bzw. ft/min
$R_{WKZ}$: Radius des aktiven Werkzeugs in mm

Der Werkzeugtyp ($TC_DP1) des aktiven Werkzeugs wird nicht berücksichtigt.

Die programmierte Werkzeug-Schnittgeschwindigkeit ist unabhängig vom Bahnvorschub F sowie der G-Funktionsgruppe 15 (Vorschubtyp). Drehrichtung und Spindelstart erfolgt über M3 bzw. M4, Spindel-Stopp über M5.

Eine Änderung der Werkzeugradiusdaten im Korrekturspeicher wird mit der nächsten Werkzeugkorrekturanwahl bzw. der nächsten Aktualisierung der aktiven Korrekturdaten wirksam.

Werkzeugwechsel und An-/Abwahl eines Werkzeugkorrekturdatensatzes führen zu einer Neuberechnung der wirksamen Spindeldrehzahl.

## Voraussetzungen

Die Programmierung der Werkzeug-Schnittgeschwindigkeit erfordert:

- die geometrischen Verhältnisse eines rotierenden Werkzeugs (Fräs- oder Bohrwerkzeug)
- einen aktiven Werkzeugkorrekturdatensatz

## Syntax

```
T... D... SVC[<n>]=<Value>
...
S... M3/M4
```

## Bedeutung

| | | |
|---|---|---|
| SVC: | Schlüsselwort zur Programmierung der Werkzeug-Schnittgeschwindigkeit | |
| [<n>]: | Nummer der Spindel<br>Mit dieser Adresserweiterung wird angegeben, für welche Spindel die programmierte Schnittgeschwindigkeit wirksam sein soll. Ohne Adresserweiterung bezieht sich die Angabe immer auf die aktuelle Masterspindel.<br>**Hinweis:**<br>Für jede Spindel kann eine eigene Schnittgeschwindigkeit vorgegeben werden.<br>**Hinweis:**<br>Die Programmierung von SVC ohne Adresserweiterung setzt voraus, dass die Masterspindel das aktive Werkzeug besitzt. Bei Wechsel der Masterspindel muss der Anwender ein entsprechendes Werkzeug anwählen. | |
| <Value>: | Wert der Werkzeug-Schnittgeschwindigkeit | |
| | Einheit: | m/min (bei G71/G710) bzw. ft/min (bei G70/G700) |
| T... D...: | Im Satz mit SVC muss der Werkzeugradius bekannt sein. D. h., ein entsprechendes Werkzeug inklusive Werkzeugkorrekturdatensatz muss aktiv bzw. im Satz angewählt sein. Die Reihenfolge von SVC und T/D-Anwahl bei Programmierung im gleichen Satz ist beliebig. | |
| S... M3/M4: | Die Programmierung der Spindeldrehzahl führt zur Abwahl der Werkzeug-Schnittgeschwindigkeit.<br>**Hinweis:**<br>Ein Wechsel zwischen SVC- und S-Programmierung ist beliebig möglich, auch bei drehender Spindel. Der jeweils nicht aktive Wert wird gelöscht. | |

**Hinweis**

Die Programmierung von SVC ist nicht möglich bei folgenden aktiven Spindel-Vorschubbewegungen:

- Konstante Schnittgeschwindigkeit: G96/G961/G962 S... (Seite 100)
- Konstante Scheibenumfangsgeschindigkeit: SUG (Seite 105)
- Spindel positionieren: SPOS/SPOSA/M19 (Seite 121)
- Masterspindel in den Achsbetrieb umschalten: M70 (Seite 121)

Umgekehrt führt die Programmierung eines dieser Funktionen zur Abwahl von SVC (Werkzeug-Schnittgeschwindigkeit).

**Hinweis**

**Maximale Werkzeugdrehzahl**

Über die Systemvariable $TC_TP_MAX_VELO[<T-Nummer>] kann eine maximale Werkzeugdrehzahl (Spindeldrehzahl) vorgegeben werden.
Wenn keine Drehzahlgrenze definiert ist, dann findet keine Überwachung statt.

**Hinweis**

Die z. B. per CAD-Systemen generierten Werkzeugbahnen von "Normwerkzeugen", welche bereits den Werkzeugradius berücksichtigen und nur die Differenz zum Normwerkzeug im Schneidenradius enthalten, werden im Zusammenhang mit der SVC-Programmierung nicht unterstützt.

## Beispiele

Für alle Beispiele soll gelten: Werkzeughalter = Spindel (für Fräsen Standard)

### Beispiel 1: Fräser mit Radius 6 mm

| Programmcode | Kommentar |
|---|---|
| `N10 G0 X10 T1 D1` | `; Anwahl Fräswerkzeug mit z.B. $TC_DP6[1,1] = 6 (WZ-Radius = 6 mm)` |
| `N20 SVC=100 M3` | `; Schnittgeschwindigkeit = 100 m/min`<br>`⇒ Resultierende Spindeldrehzahl:`<br>`S = (100 m/min * 1000) / (6,0 mm * 2 * 3,14) = 2653,93 U/min` |
| `N30 G1 X50 G95 FZ=0.03` | `; SVC und Zahnvorschub` |
| `...` | |

### Beispiel 2: WZ–Anwahl und SVC im gleichen Satz

| Programmcode | Kommentar |
|---|---|
| `N10 G0 X20` | |
| `N20 T1 D1 SVC=100` | `; Werkzeug- und Korrekturdatensatz-Anwahl zusammen mit SVC im Satz (Reihenfolge beliebig).` |

| Programmcode | Kommentar |
|---|---|
| N30 X30 M3 | ; Spindelstart mit Drehrichtung rechts, Schnittgeschwindigkeit 100 m/min |
| N40 G1 X20 F0.3 G95 | ; SVC und Umdrehungsvorschub |

### Beispiel 3: Schnittgeschwindigkeiten für zwei Spindeln vorgeben

| Programmcode | Kommentar |
|---|---|
| N10 SVC[3]=100 M6 T1 D1 | |
| N20 SVC[5]=200 | ; Der Werkzeugradius der aktiven Werkzeugkorrektur ist für beide Spindeln gleich, die wirksame Drehzahl ist für Spindel 3 und Spindel 5 unterschiedlich. |

### Beispiel 4:

Annahmen:

Master bzgl. Werkzeugwechsel wird durch die Toolholder bestimmt:

MD20124 $MC_TOOL_MANAGEMENT_TOOLHOLDER > 1

Beim Werkzeugwechsel wird die alte Werkzeugkorrektur beibehalten und erst mit der Programmierung von D wird eine Werkzeugkorrektur des neuen Werkzeugs aktiv:

MD20270 $MC_CUTTING_EDGE_DEFAULT = - 2

| Programmcode | Kommentar |
|---|---|
| N10 $TC_MPP1[9998,1]=2 | ; Magazinplatz ist Werkzeughalter |
| N11 $TC_MPP5[9998,1]=1 | ; Magazinplatz ist Werkzeughalter 1 |
| N12 $TC_MPP_SP[9998,1]=3 | ; Werkzeughalter 1 ist der Spindel 3 zugeordnet |
| N20 $TC_MPP1[9998,2]=2 | ; Magazinplatz ist Werkzeughalter |
| N21 $TC_MPP5[9998,2]=4 | ; Magazinplatz ist Werkzeughalter 4 |
| N22 $TC_MPP_SP[9998,2]=6 | ; Werkzeughalter 4 ist der Spindel 6 zugeordnet |
| N30 $TC_TP2[2]="WZ2" | |
| N31 $TC_DP6[2,1]=5.0 | ; Radius = 5,0 mm von T2, Korrektur D1 |
| N40 $TC_TP2[8]="WZ8" | |
| N41 $TC_DP6[8,1]=9.0 | ; Radius = 9,0 mm von T8, Korrektur D1 |
| N42 $TC_DP6[8,4]=7.0 | ; Radius = 7,0 mm von T8, Korrektur D4 |
| ... | |
| N100 SETMTH(1) | ; Masterwerkzeughalternummer setzen |
| N110 T="WZ2" M6 D1 | ; Werkzeug T2 wird eingewechselt und Korrektur D1 aktiviert. |
| N120 G1 G94 F1000 M3=3 SVC=100 | ; S3 = (100 m/min * 1000) / (5,0 mm * 2 * 3,14) = 3184,71 U/min |
| N130 SETMTH(4) | ; Masterwerkzeughalternummer setzen |
| N140 T="WZ8" | ; Entspricht T8="WZ8" |
| N150 M6 | ; Entspricht M4=6 |
| | Werkzeug "WZ8" kommt auf Mastertoolholder, aber wegen MD20270=-2 bleibt alte Werkzeugkorrektur aktiv. |
| N160 SVC=50 | ; S3 = (50 m/min * 1000) / (5,0 mm * 2 * 3,14) = 1592,36 U/min |
| | Korrektur von Werkzeughalter 1 ist noch aktiv und dieser ist Spindel 3 zugeordnet. |

| Programmcode | Kommentar |
|---|---|
| N170 D4 | ; Korrektur D4 von neuem Werkzeug "WZ8" wird aktiv (auf Werkzeughalter 4). |
| N180 SVC=300 | ; S6 = (300 m/min * 1000) / (7,0 mm * 2 * 3,14) = 6824,39 U/min |
| | Spindel 6 ist Werkzeughalter 4 zugeordnet. |

**Beispiel 5:**

Annahmen:

Spindeln sind zugleich Werkzeughalter:

MD20124 $MC_TOOL_MANAGEMENT_TOOLHOLDER = 0

Bei Werkzeugwechsel wird automatisch Werkzeugkorrekturdatensatz D4 angewählt:

MD20270 $MC_CUTTING_EDGE_DEFAULT = 4

| Programmcode | Kommentar |
|---|---|
| N10 $TC_MPP1[9998,1]=2 | ; Magazinplatz ist Werkzeughalter |
| N11 $TC_MPP5[9998,1]=1 | ; Magazinplatz ist Werkzeughalter 1 = Spindel 1 |
| N20 $TC_MPP1[9998,2]=2 | ; Magazinplatz ist Werkzeughalter |
| N21 $TC_MPP5[9998,2]=3 | ; Magazinplatz ist Werkzeughalter 3 = Spindel 3 |
| N30 $TC_TP2[2]="WZ2" | |
| N31 $TC_DP6[2,1]=5.0 | ; Radius = 5,0 mm von T2, Korrektur D1 |
| N40 $TC_TP2[8]="WZ8" | |
| N41 $TC_DP6[8,1]=9.0 | ; Radius = 9,0 mm von T8, Korrektur D1 |
| N42 $TC_DP6[8,4]=7.0 | ; Radius = 7,0 mm von T8, Korrektur D4 |
| ... | |
| N100 SETMS(1) | ; Spindel 1 = Masterspindel |
| N110 T="WZ2" M6 D1 | ; Werkzeug T2 wird eingewechselt und Korrektur D1 aktiviert. |
| N120 G1 G94 F1000 M3 SVC=100 | ; S1 = (100 m/min * 1000) / (5,0 mm * 2 * 3,14) = 3184,71 U/min |
| N200 SETMS(3) | ; Spindel 3 = Masterspindel |
| N210 M4 SVC=150 | ; S3 = (150 m/min * 1000) / (5,0 mm * 2 * 3,14) = 4777,07 U/min |
| | Bezieht sich auf die Werkzeugkorrektur D1 von T="WZ2", S1 dreht mit alter Drehzahl weiter. |
| N220 T="WZ8" | ; Entspricht T8="WZ8" |
| N230 M4 SVC=200 | ; S3 = (200 m/min * 1000) / (5,0 mm * 2 * 3,14) = 6369,43 U/min |
| | Bezieht sich auf die Werkzeugkorrektur D1 von T="WZ2". |
| N240 M6 | ; Entspricht M3=6 |
| | Werkzeug "WZ8" kommt auf Masterspindel, die Werkzeugkorrektur D4 des neuen Werkzeugs wird aktiv. |
| N250 SVC=50 | ; S3 = (50 m/min * 1000) / (7,0 mm * 2 * 3,14) = 1137,40 U/min |
| | Korrektur D4 auf Masterspindel ist aktiv. |
| N260 D1 | ; Korrektur D1 von neuem Werkzeug "WZ8" aktiv. |

| **Programmcode** | **Kommentar** |
|---|---|
| N270 SVC[1]=300 | ; S1 = (300 m/min * 1000) / (9,0 mm * 2 * 3,14) = 5307,86 U/min |
| | S3 = (50 m/min * 1000) / (9,0 mm * 2 * 3,14) = 884,64 U/min |
| ... | |

## Weitere Informationen

### Werkzeugradius

Folgende Werkzeugkorrekturdaten (des aktiven Werkzeugs) tragen zum Werkzeugradius bei:

- $TC_DP6 (Radius- Geometrie)
- $TC_DP15 (Radius - Verschleiß)
- $TC_SCPx6 (Korrektur zu $TC_DP6)
- $TC_ECPx6 (Korrektur zu $TC_DP6)

Nicht berücksichtigt werden:

- Online-Radiuskorrekturen
- Aufmaß zur programmierten Kontur (OFFN)

### Werkzeugradiuskorrektur (G41/G42)

Werkzeugradiuskorrektur (G41/G42) und SVC beziehen sich beide auf den Werkzeugradius, sind aber funktionell entkoppelt und unabhängig voneinander.

### Gewindebohren ohne Ausgleichsfutter (G331, G332)

Die Programmierung von SVC ist auch in Verbindung mit G331 bzw. G332 möglich.

### Synchronaktionen

Die Vorgabe von SVC aus Synchronaktionen ist nicht möglich.

### Schnittgeschwindigkeit und Spindeldrehzahl-Programmierungsvariante lesen

Die Schnittgeschwindigkeit einer Spindel und die Drehzahl-Programmierungsvariante (Spindeldrehzahl S oder Werkzeug-Schnittgeschwindigkeit SVC) können über Systemvariablen gelesen werden:

- Mit Vorlauf-Stopp im Teileprogramm über die Systemvariablen:

| | |
|---|---|
| $AC_SVC[<n>] | Schnittgeschwindigkeit, die bei der Aufbereitung des aktuellen Hauptlaufsatzes für die Spindel mit Nummer <n> wirksam war. |
| $AC_S_TYPE[<n>] | Spindeldrehzahl-Programmierungsvariante, die bei der Aufbereitung des aktuellen Hauptlaufsatzes für die Spindel mit Nummer <n> wirksam war. |

| **Wert:** | **Bedeutung:** |
|---|---|
| 1 | Spindeldrehzahl S in U/min |
| 2 | Werkzeug-Schnittgeschwindigkeit SVC in m/min bzw. ft/min |

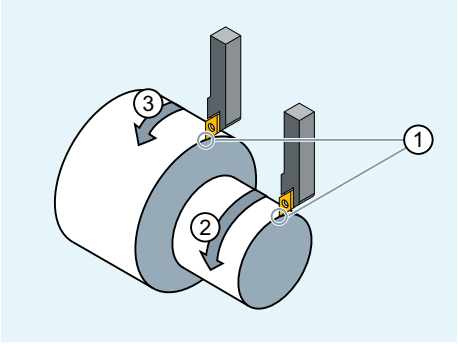
- Ohne Vorlauf-Stopp im Teileprogramm über die Systemvariablen:

| | |
|---|---|
| $P_SVC[<n>] | Programmierte Schnittgeschwindigkeit für Spindel <n> |
| $P_S_TYPE[<n>] | Programmierte Spindeldrehzahl-Programmierungsvariante für Spindel <n> |

| Wert: | Bedeutung: |
|---|---|
| 1 | Spindeldrehzahl S in U/min |
| 2 | Werkzeug-Schnittgeschwindigkeit SVC in m/min bzw. ft/min |

## 2.6.3 Konstante Schnittgeschwindigkeit (G96/G961/G962, G97/G971/G972, G973, LIMS, SCC)

Bei aktiver Funktion "Konstante Schnittgeschwindigkeit" wird, abhängig vom jeweiligen Werkstückdurchmesser, die Spindeldrehzahl so verändert, dass die Schnittgeschwindigkeit S in m/min bzw. ft/min an der Werkzeugschneide konstant bleibt.




① Schnittgeschwindigkeit konstant
② Spindeldrehzahl erhöht
③ Spindeldrehzahl verringert

Daraus ergeben sich folgende Vorteile:

- Gleichmäßige Drehbilder und damit eine hohe Oberflächenqualität
- Werkzeugschonende Bearbeitung

### Syntax

**Konstante Schnittgeschwindigkeit für die Masterspindel ein-/ausschalten:**

```
G96/G961/G962 S...
...
G97/G971/G972/G973
```

**Drehzahlbegrenzung für die Masterspindel:**
`LIMS=<Wert>`
`LIMS[<Spindel>]=<Wert>`

**Andere Bezugsachse für G96/G961/G962:**
`SCC[<Achse>]`

**Hinweis**

SCC[<Achse>] kann getrennt oder zusammen mit G96/G961/G962 programmiert werden.

## Bedeutung

| | | |
|---|---|---|
| `G96`: | Umdrehungsvorschub (wie bei G95 (Seite 108)) und konstante Schnittgeschwindigkeit<br>Mit G96 wird automatisch G95 eingeschaltet. Falls G95 vorher noch nicht eingeschaltet war, muss bei Aufruf von G96 ein neuer Vorschubwert F... angegeben werden. | |
| `G961`: | Linearvorschub (wie bei G94 (Seite 108)) und konstante Schnittgeschwindigkeit | |
| `G962`: | Linearvorschub oder Umdrehungsvorschub und konstante Schnittgeschwindigkeit | |
| `S...`: | Zusammen mit G96, G961 bzw. G962 wird S... nicht als Spindeldrehzahl sondern als Schnittgeschwindigkeit interpretiert. Die Schnittgeschwindigkeit wirkt immer auf die Masterspindel. | |
| | Einheit: | m/min (bei G71/G710) bzw. ft/min (bei G70/G700) |
| | Wertebereich: | 0,1 m/min ... 9999 9999,9 m/min |
| `G97`: | Umdrehungsvorschub und konstante Spindeldrehzahl (konstante Schnittgeschwindigkeit AUS) | |
| `G971`: | Linearvorschub und konstante Spindeldrehzahl (konstante Schnittgeschwindigkeit AUS) | |
| `G972`: | Linearvorschub oder Umdrehungsvorschub und konstante Spindeldrehzahl (konstante Schnittgeschwindigkeit AUS) | |
| `G973`: | Umdrehungsvorschub ohne Spindeldrehzahlbegrenzung und konstante Spindeldrehzahl (G97 ohne LIMS für ISO-Modus) | |
| | **Hinweis:**<br>Nach G97 (bzw. G971 ... G973) wird S... wieder als Spindeldrehzahl in Umdrehungen/min interpretiert. Falls keine neue Spindeldrehzahl angegeben ist, wird die zuletzt durch G96 (bzw. G961 oder G962) eingestellte Drehzahl beibehalten. | |
| `LIMS`: | Drehzahlbegrenzung für die Masterspindel (nur wirksam bei aktivem G96/G961/G97)<br>Bei Maschinen mit umschaltbaren Masterspindeln können in einem Satz für bis zu 4 Spindeln Begrenzungen mit unterschiedlichen Werten programmiert werden. | |
| | `<Spindel>`: | Nummer der Spindel |
| | `<Wert>`: | Spindeldrehzahlobergrenze in Umdrehungen/min |
| `SCC`: | Bei aktiver Funktion G96/G961/G962 kann mit SCC[<Achse>] eine beliebige Geometrieachse als Bezugsachse zugeordnet werden. | |

**Hinweis**

Bei Erstanwahl von G96/G961/G962 muss eine konstante Schnittgeschwindigkeit S... eingegeben werden, bei Wiederanwahl von G96/G961/G962 ist die Angabe optional.

**Hinweis**

Die mit LIMS programmierte Drehzahlbegrenzung darf die mit G26 programmierte oder über Settingdaten festgelegte Grenzdrehzahl nicht überschreiten.

**Hinweis**

Die Bezugsachse für G96/G961/G962 muss zum Programmierzeitpunkt von SCC[<Achse>] eine im Kanal bekannte Geometrieachse sein. Die Programmierung von SCC[<Achse>] ist auch bei aktivem G96/G961/G962 möglich.

## Beispiele

### Beispiel 1: Konstante Schnittgeschwindigkeit mit Drehzahlbegrenzung einschalten

```
Programmcode                                Kommentar
N10 SETMS(3)
N20 G96 S100 LIMS=2500                      ; Konstante Schnittgeschwindigkeit =
                                            100 m/min, Max. Drehzahl = 2500 U/min
...
N60 G96 G90 X0 Z10 F8 S100 LIMS=444         ; Max. Drehzahl = 444 U/min
```

### Beispiel 2: Drehzahlbegrenzung für 4 Spindeln vorgeben

Die Drehzahlbegrenzungen werden für die Spindel 1 (Masterspindel) und die Spindeln 2, 3 und 4 festgelegt:

```
Programmcode
N10 LIMS=300 LIMS[2]=450 LIMS[3]=800
LIMS[4]=1500
...
```

### Beispiel 3: Zuordnung einer Y-Achse bei einer Planbearbeitung mit X-Achse

```
Programmcode                Kommentar
N10 G18 LIMS=3000 T1 D1     ; Drehzahlbegrenzung auf 3000 U/min
N20 G0 X100 Z200
N30 Z100
N40 G96 S20 M3              ; Konstante Schnittgeschwindigkeit = 20 m/min, ist
                            abhängig von der X-Achse.
N50 G0 X80
N60 G1 F1.2 X34             ; Planbearbeitung in X mit 1,2 mm/Umdrehung.
N70 G0 G94 X100
N80 Z80
N100 T2 D1
N110 G96 S40 SCC[Y]         ; Y-Achse wird G96 zugeordnet und G96 aktiviert (ist
                            in einem Satz möglich). Konstante Schnittgeschwindig-
                            keit = 40 m/min, ist abhängig von der Y-Achse.
...
N140 Y30
```

| Programmcode | Kommentar |
|---|---|
| N150 G01 F1.2 Y=27 | ; Einstechen in Y, Vorschub F = 1,2 mm/Umdrehungen. |
| N160 G97 | ; Konstante Schnittgeschwindigkeit aus. |
| N170 G0 Y100 | |

## Weitere Informationen

### Berechnung der Spindeldrehzahl

Basis für die Berechnung der Spindeldrehzahl aus der programmierten Schnittgeschwindigkeit ist die ENS-Position der Planachse (Radius).

**Hinweis**

Frames zwischen WKS und ENS (z. B. programmierbare Frames wie SCALE, TRANS oder ROT) werden bei der Berechnung der Spindeldrehzahl berücksichtigt und können eine Drehzahländerung bewirken (z. B. wenn sich bei SCALE der wirksame Durchmesser ändert).

### Drehzahlbegrenzung LIMS

Falls ein Werkstück mit großen Durchmesserunterschieden zu bearbeiten ist, empfiehlt sich die Angabe einer Spindeldrehzahlbegrenzung mit LIMS (maximale Spindeldrehzahl). Hierdurch lassen sich bei kleinen Durchmessern unzulässig hohe Drehzahlen ausschließen. LIMS ist nur wirksam bei aktivem G96, G961 und G97. Bei G971 wirkt LIMS nicht. Beim Einwechseln des Satzes in den Hauptlauf werden alle programmierten Werte in die Settingdaten übernommen.

**Hinweis**

Die im Teileprogramm mit LIMS geänderten Drehzahlgrenzen werden in die Settingdaten übernommen und bleiben somit auch über das Programmende hinaus gespeichert.

Sollen die mit LIMS geänderten Drehzahlgrenzen nach Programmende dagegen nicht mehr gelten, muss folgende Definition in den GUD-Baustein des Maschinenherstellers eingefügt werden:

REDEF $SA_SPIND_MAX_VELO_LIMS PRLOC

### Konstante Schnittgeschwindigkeit ausschalten (G97/G971/G972/G973)

Nach G97 (bzw. G971 ... G973) wird S... wieder als Spindeldrehzahl in Umdrehungen/min interpretiert. Falls keine neue Spindeldrehzahl angegeben ist, wird die zuletzt durch G96 (bzw. G961 oder G962) eingestellte Drehzahl beibehalten.

Die Funktion G96/G961 kann auch mit G94 oder G95 ausgeschaltet werden. In diesem Fall gilt die zuletzt programmierte Drehzahl S... für den weiteren Bearbeitungsablauf.

G97 kann ohne vorheriges G96 programmiert werden. Die Funktion wirkt dann wie G95, zusätzlich kann LIMS programmiert werden.

Mit G973 kann die konstante Schnittgeschwindigkeit ausgeschaltet werden, ohne dass eine Spindeldrehzahlbegrenzung aktiviert wird.

**Hinweis**

Die Planachse muss über Maschinendatum definiert sein.

### Fahren im Eilgang G0

Beim Fahren im Eilgang G0 werden keine Drehzahländerungen vorgenommen.

Ausnahme:

Wird die Kontur im Eilgang angefahren und der nächste NC-Satz enthält einen Bahnbefehl G1/G2/G3/…, dann stellt sich bereits im Anfahrsatz G0 die Drehzahl für den nächsten Bahnbefehl ein.

### Andere Bezugsachse für G96/G961/G962

Bei aktiver Funktion G96/G961/G962 kann mit SCC[<Achse>] eine beliebige Geometrieachse als Bezugsachse zugeordnet werden. Ändert sich die Bezugsachse und damit die Bezugsposition der Werkzeugspitze (TCP - Tool Center Point) für die konstante Schnittgeschwindigkeit, wird die resultierende Spindeldrehzahl über die eingestellte Brems- bzw. Beschleunigungsrampe angefahren.

### Achstausch der zugeordneten Kanalachse

Die Eigenschaft Bezugsachse für G96/G961/G962 ist immer einer Geometrieachse zugeordnet. Bei Achstausch der zugeordneten Kanalachse bleibt die Eigenschaft Bezugsachse für G96/G961/G962 im alten Kanal.

Ein Geometrieachstausch beeinflusst die Zuordnung Geometrieachse zur konstanten Schnittgeschwindigkeit nicht. Verändert ein Geometrieachstausch die TCP-Bezugsposition für G96/G961/G962, so fährt die Spindel über Rampe die neue Drehzahl an.

Wird durch Geometrieachstausch keine neue Kanalachse zugeordnet (z. B. GEOAX(0,X)), so wird die Spindeldrehzahl entsprechend G97 eingefroren.

Beispiele für Geometrieachstausch mit Zuordnungen der Bezugsachse:

| **Programmcode** | **Kommentar** |
|---|---|
| N05 G95 F0.1 | |
| N10 GEOAX(1,X1) | ; Kanalachse X1 wird zur ersten Geoachse. |
| N20 SCC[X] | ; Erste Geoachse (X) wird zur Bezugsachse |
| | ; für G96/G961/G962. |
| N30 GEOAX(1,X2) | ; Kanalachse X2 wird zur ersten Geoachse. |
| N40 G96 M3 S20 | ; Bezugsachse für G96 ist Kanalachse X2. |

| **Programmcode** | **Kommentar** |
|---|---|
| N05 G95 F0.1 | |
| N10 GEOAX(1,X1) | ; Kanalachse X1 wird zur ersten Geoachse. |
| N20 SCC[X1] | ; X1 und implizit die erste Geoachse (X) wird |
| | ;zur Bezugsachse für G96/G961/G962. |

| Programmcode | Kommentar |
|---|---|
| N30 GEOAX(1,X2) | ; Kanalachse X2 wird zur ersten Geoachse. |
| N40 G96 M3 S20 | ; Bezugsachse für G96 ist X2 bzw. X, kein Alarm. |

| Programmcode | Kommentar |
|---|---|
| N05 G95 F0.1 | |
| N10 GEOAX(1,X2) | ; Kanalachse X2 wird zur ersten Geoachse. |
| N20 SCC[X1] | ; X1 ist keine Geoachse, Alarm. |

| Programmcode | Kommentar |
|---|---|
| N05 G0 Z50 | |
| N10 X35 Y30 | |
| N15 SCC[X] | ; Bezugsachse für G96/G961/G962 ist X. |
| N20 G96 M3 S20 | ; Konstante Schnittgeschwindigkeit mit 10 mm/min ein. |
| N25 G1 F1.5 X20 | ; Planbearbeitung in X mit 1,5 mm/Umdrehung. |
| N30 G0 Z51 | |
| N35 SCC[Y] | ; Bezugsachse für G96 ist Y, |
| | ; Reduzierung Spindeldrehzahl (Y30). |
| N40 G1 F1.2 Y25 | ; Planbearbeitung in Y mit 1,2 mm/Umdrehung. |

## 2.6.4 Konstante Scheibenumfangsgeschwindigkeit ein-/ausschalten (GWPSON, GWPSOF)

Mit den vordefinierten Prozeduren GWPSON(...) und GWPSOF(...) wird die konstante Scheibenumfangsgeschwindigkeit (SUG) für Schleifwerkzeuge (WZ-Typ: 400 bis 499) ein- bzw. ausgeschaltet.

### Syntax

```
GWPSON(<TNr>)
S<n>=...
...
GWPSOF(<TNr>)
```

### Bedeutung

| | |
|---|---|
| GWPSON(...): | Konstante Scheibenumfangsgeschwindigkeit einschalten |
| GWPSOF(...): | Konstante Scheibenumfangsgeschwindigkeit ausschalten |
| <TNr>: | T-Nummer<br>**Hinweis:**<br>Nur erforderlich, wenn die konstante Scheibenumfangsgeschwindigkeit statt für das aktive, im Einsatz befindliche Werkzeug für eine nicht aktive Schleifscheibe ein- bzw. ausgeschaltet werden soll. |

| | |
|---|---|
| S<n>=…: | Scheibenumfangsgeschwindigkeit in m/s oder ft/s für Spindel <n> |
| S0=... bzw. S... : | Scheibenumfangsgeschwindigkeit für die Masterspindel |

### Status abfragen

Mit der folgenden Systemvariablen kann vom Teileprogramm aus abgefragt werden, ob die konstante Scheibenumfangsgeschwindigkeit für eine bestimmte Spindel aktiv ist:

$P_GWPS[<n>] ; mit <n> = Spindelnummer

| Wert | Bedeutung |
|---|---|
| 0 (= FALSE) | SUG ist **aus**geschaltet. |
| 1 (= TRUE) | SUG ist **ein**geschaltet. |

## 2.6.5 Programmierbare Spindeldrehzahlbegrenzung (G25, G26)

Die in Maschinen- und Settingdaten festgelegten minimalen und maximalen Spindeldrehzahlen können per Teileprogrammbefehl verändert werden.

Programmierte Spindeldrehzahlbegrenzungen sind für alle Spindeln des Kanals möglich.

### Syntax

```
G25 S… S1=… S2=…
G26 S… S1=… S2=…
```

### Bedeutung

| | |
|---|---|
| G25: | **Untere** Spindeldrehzahlbegrenzung |
| G26: | **Obere** Spindeldrehzahlbegrenzung |
| S... S1=… S2=… : | Minimale bzw. maximale Spindeldrehzahl(en)<br>**Hinweis:**<br>Pro Satz dürfen maximal drei Spindeldrehzahlbegrenzungen programmiert werden.<br>Wertebereich: 0.1 ... 9999 9999.9 U/min |

**Hinweis**

Eine mit G25 oder G26 programmierte Spindeldrehzahlbegrenzung überschreibt die Grenzdrehzahlen in den Settingdaten und bleibt somit auch über das Programmende hinaus gespeichert.

Sollen die mit G25/G26 geänderten Drehzahlgrenzen nach Programmende dagegen nicht mehr gelten, müssen folgende Definitionen in den GUD-Baustein des Maschinenherstellers eingefügt werden:

REDEF $SA_SPIND_MIN_VELO_G25 PRLOC

REDEF $SA_SPIND_MAX_VELO_G26 PRLOC

## Beispiel

| Programmcode | Kommentar |
|---|---|
| `N10 G26 S1400 S2=350 S3=600` | `; Obere Grenzdrehzahl für Masterspindel, Spindel 2 und Spindel 3` |

# 2.7 Vorschubregelung

## 2.7.1 Vorschub (G93, G94, G95, F, FGROUP, FL, FGREF)

Mit diesen Befehlen werden im NC-Programm die Vorschubgeschwindigkeiten für alle an der Bearbeitungsfolge beteiligten Achsen eingestellt.

### Syntax

```
G93
G94
G95
F<Wert>
FGROUP(<Achse_1>,<Achse_2>,...)
FGREF[<Rundachse>]=<Bezugsradius>
FL[<Achse>]=<Wert>
```

### Bedeutung

| | |
|---|---|
| G93: | Typ des Bahnvorschubs: Zeitreziproker Vorschub [1/min] |
| G94: | Typ des Bahnvorschubs: Linearvorschub [mm/min], [inch/min] oder [Grad/min] |
| G95: | Typ des Bahnvorschubs: Umdrehungsvorschub [mm/Umdrehung] bzw. [inch/Umdrehung]<br>Der Umdrehungsvorschub kann wählbar abgeleitet werden von einer Masterspindel, einer beliebigen anderen Spindel oder Rundachse. |
| F<Wert> | Bahnvorschub für alle oder die mit FGROUP ausgewählten Bahnachsen. |
| FGROUP: | Festlegung der Bahnachsen, auf die sich der unter F programmierte Bahnvorschub bezieht. |
| FGREF: | Mit FGREF wird für jede der unter FGROUP angegebenen Rundachsen der effektive Radius (<Bezugsradius>) programmiert |
| FL: | Grenzgeschwindigkeit für Synchron-/Bahnachsen<br>Es gilt die mit G94 eingestellte Einheit.<br>Pro Achse (Kanalachse, Geometrieachse oder Orientierungsachse) kann ein FL-Wert programmiert werden. |
| <Achse>: | Name einer Kanalachse, Typ: AXIS |

### Beispiele

#### Beispiel 1: Wirkungsweise von FGROUP

Das folgende Beispiel soll die Wirkung von FGROUP auf den Bahnweg und Bahnvorschub verdeutlichen. Die Variable $AC_TIME enthält die Zeit vom Satzanfang in Sekunden. Sie ist nur in Synchronaktionen verwendbar.

| Programmcode | Kommentar |
|---|---|
| N100 G0 X0 A0 | |
| N110 FGROUP(X,A) | |

| **Programmcode** | **Kommentar** |
|---|---|
| N120 G91 G1 G710 F100 | ; Vorschub= 100mm/min bzw. 100Grad/min |
| N130 DO $R1=$AC_TIME | |
| N140 X10 | ; Vorschub= 100mm/min, Bahnweg= 10mm, R1= ca.6s |
| N150 DO $R2=$AC_TIME | |
| N160 X10 A10 | ; Vorschub= 100mm/min, Bahnweg= 14.14mm, R2= ca.8s |
| N170 DO $R3=$AC_TIME | |
| N180 A10 | ; Vorschub= 100Grad/min, Bahnweg= 10Grad, R3= ca.6s |
| N190 DO $R4=$AC_TIME | |
| N200 X0.001 A10 | ; Vorschub= 100mm/min, Bahnweg= 10mm, R4= ca.6s |
| N210 G700 F100 | ; Vorschub= 2540mm/min bzw. 100Grad/min |
| N220 DO $R5=$AC_TIME | |
| N230 X10 | ; Vorschub= 2540mm/min, Bahnweg= 254mm, R5= ca.6s |
| N240 DO $R6=$AC_TIME | |
| N250 X10 A10 | ; Vorschub= 2540mm/min, Bahnweg= 254,2mm, R6= ca.6s |
| N260 DO $R7=$AC_TIME | |
| N270 A10 | ; Vorschub= 100Grad/min, Bahnweg= 10Grad, R7= ca.6s |
| N280 DO $R8=$AC_TIME | |
| N290 X0.001 A10 | ; Vorschub= 2540mm/min, Bahnweg= 10mm, R8= ca.0.288s |
| N300 FGREF[A]=360/(2*$PI) | ; 1 Grad = 1 inch über den effektiven Radius einstellen. |
| N310 DO $R9=$AC_TIME | |
| N320 X0.001 A10 | ; Vorschub= 2540mm/min, Bahnweg= 254mm, R9= ca.6s |
| N330 M30 | |

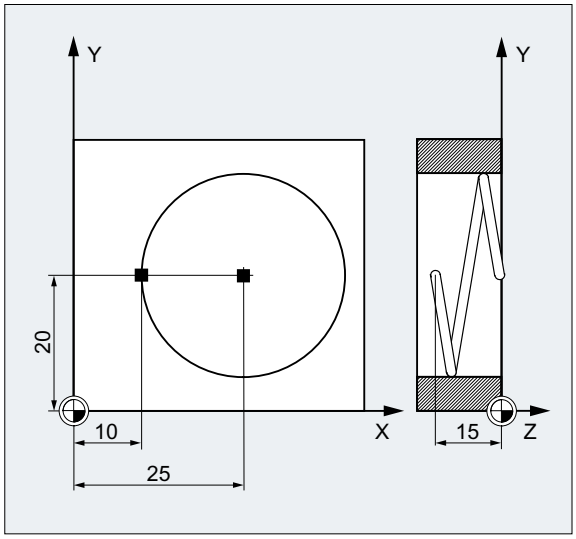
### Beispiel 2: Synchronachsen mit Grenzgeschwindigkeit FL verfahren

Die Bahngeschwindigkeit der Bahnachsen wird reduziert, falls die Synchronachse Z die Grenzgeschwindigkeit erreicht.

| **Programmcode** |
|---|
| N10 G0 X0 Y0 |
| N20 FGROUP(X) |
| N30 G1 X1000 Y1000 G94 F1000 FL[Y]=500 |
| N40 Z-50 |

### Beispiel 3: Schraubenlinieninterpolation

Die Bahnachsen X und Y fahren mit programmiertem Vorschub, die Zustellachse Z ist Synchronachse.

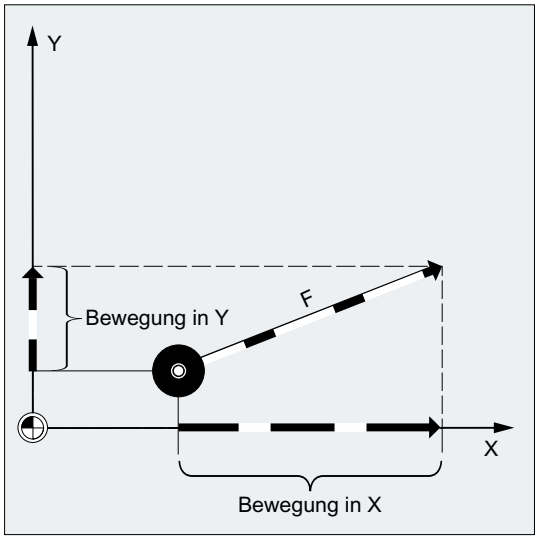



| Programmcode | Kommentar |
|---|---|
| N10 G17 G94 G1 Z0 F500 | ; Zustellen des Werkzeugs. |
| N20 X10 Y20 | ; Anfahren der Startposition. |
| N25 FGROUP(X,Y) | ; Achsen X/Y sind Bahnachsen, Z ist Synchronachse. |
| N30 G2 X10 Y20 Z-15 I15 J0 F1000 FL[Z]=200 | ; Auf der Kreisbahn gilt Vorschub 1000 mm/min, in Z-Richtung wird synchron verfahren. |
| ... | |
| N100 FL[Z]=$MA_AX_VELO_LIMIT[0,Z] | ; Durch Lesen der Geschwindigkeit aus dem MD wird die Grenzgeschwindigkeit abgewählt, der Wert aus dem MD gelesen. |
| N110 M30 | ; Programmende. |

## Weitere Informationen

### Vorschubgeschwindigkeit für Bahnachsen (F)

Im Regelfall setzt sich der Bahnvorschub aus den einzelnen Geschwindigkeitskomponenten aller an der Bewegung beteiligten Geometrieachsen zusammen und bezieht sich auf den Fräsermittelpunkt bzw. auf die Werkzeugspitze des Drehstahls.




Die Vorschubgeschwindigkeit wird unter der Adresse `F` angegeben. Je nach Voreinstellung in den Maschinendaten gelten die über G-Befehle festgelegten Maßeinheiten in mm oder inch.

Pro NC-Satz darf ein `F`-Wert programmiert werden. Die Einheit der Vorschubgeschwindigkeit wird über einen der G-Befehle `G93`/`G94`/`G95` festgelegt. Der Vorschub `F` wirkt nur auf Bahnachsen und gilt solange, bis ein neuer Vorschubwert programmiert wird. Nach der Adresse `F` sind Trennzeichen zulässig.

Beispiele:

`F100` oder `F 100`

```
F.5
```

```
F=2*FEED
```

### Vorschubart (G93/G94/G95)

Die G-Befehle `G93`, `G94` und `G95` sind modal wirksam. Wird zwischen `G93`, `G94` und `G95` umgeschaltet, so ist der Bahnvorschubwert erneut zu programmieren. Für die Bearbeitung mit Rundachsen kann der Vorschub auch in Grad/min angegeben werden.

### Zeitreziproker Vorschub (G93)

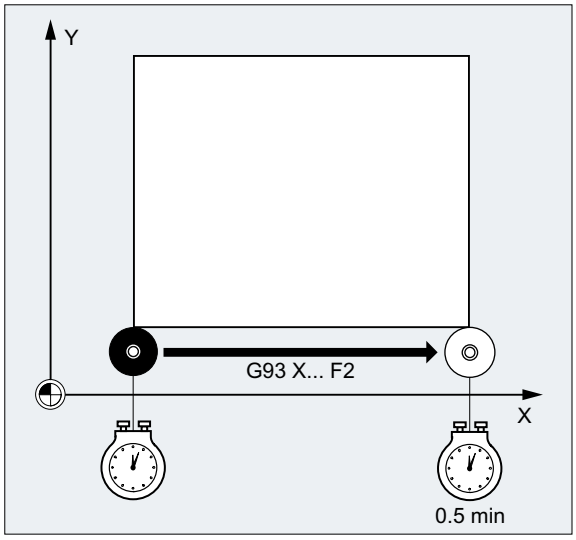
Der zeitreziproke Vorschub gibt die Zeitdauer für das Abfahren eines Satzes an.

Einheit: 1/min

Beispiel:

```
N10 G93 G01 X100 F2
```

Bedeutet: der programmierte Bahnweg wird in 0,5 min abgefahren.




**Hinweis**

Falls die Bahnlängen von Satz zu Satz sehr unterschiedlich sind, sollte bei `G93` in jedem Satz ein neuer `F`-Wert bestimmt werden. Für die Bearbeitung mit Rundachsen kann der Vorschub auch in Grad/min angegeben werden.

### Vorschub für Synchronachsen

Der unter der Adresse `F` programmierte Vorschub gilt für alle im Satz programmierten Bahnachsen, jedoch nicht für Synchronachsen. Die Synchronachsen werden so gesteuert, dass sie für ihren Weg die gleiche Zeit benötigen wie die Bahnachsen und alle Achsen ihren Endpunkt zur gleichen Zeit erreichen.

### Grenzgeschwindigkeit für Synchronachsen (FL)

Mit dem Befehl `FL` kann für Synchronachsen eine Grenzgeschwindigkeit programmiert werden. Falls kein `FL` programmiert wird, gilt die Eilganggeschwindigkeit. Abgewählt wird `FL` durch MD-Zuweisung (MD36200 $MA_AX_VELO_LIMIT).

### Bahnachse als Synchronachse verfahren (FGROUP)

Mit `FGROUP` wird festgelegt, ob eine Bahnachse mit Bahnvorschub oder als Synchronachse verfahren werden soll. Bei der Schraubenlinieninterpolation kann z. B. festgelegt werden, dass nur zwei Geometrieachsen X und Y mit programmiertem Vorschub verfahren werden sollen. Die Zustellachse Z wäre dann Synchronachse.

Beispiel: `FGROUP(X,Y)`

### FGROUP ändern

Eine Änderung der mit FGROUP getroffenen Einstellung ist möglich:

1. durch erneute Programmierung von FGROUP: z. B. FGROUP(X,Y,Z)
2. durch Programmierung von FGROUP ohne Achsangabe: FGROUP()
   Nach FGROUP() gilt der im Maschinendatum eingestellte Grundzustand. Geometrieachsen fahren jetzt wieder im Bahnachsverbund.

**Hinweis**

Achsbezeichner bei FGROUP müssen Kanalachsnamen sein.

### Maßeinheiten für den Vorschub F

Mit den G-Befehlen G700 und G710 wird zusätzlich zu den geometrischen Angaben auch das Maßsystem für die Vorschübe F definiert, d. h.:

- bei G700: [inch/min]
- bei G710: [mm/min]

**Hinweis**

Durch G70/G71 werden Vorschubangaben **nicht** beeinflusst.

### Maßeinheit für Synchronachsen mit Grenzgeschwindigkeit FL

Die für F per G-Befehl G700/G710 eingestellte Maßeinheit gilt auch für FL.

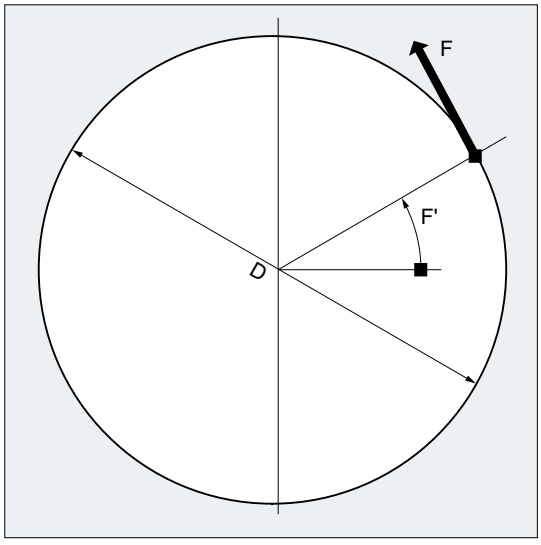
### Maßeinheit für Rund- und Linearachsen

Für Linear- und Rundachsen, die über FGROUP miteinander verbunden sind und gemeinsam eine Bahn fahren, gilt der Vorschub in der Maßeinheit der Linearachsen. Je nach Voreinstellung mit G94/G95 in mm/min oder inch/min bzw. mm/Umdrehung oder inch/Umdrehung.

Die Tangentialgeschwindigkeit der Rundachse in mm/min oder inch/min errechnet sich nach der Formel:

F[mm/min] = F'[Grad/min] * π * D[mm] / 360[Grad]

mit: F: Tangentialgeschwindigkeit
F': Winkelgeschwindigkeit
π: Kreiskonstante
D: Durchmesser




### Rundachsen mit Bahngeschwindigkeit F verfahren (FGREF)

Für Bearbeitungsvorgänge, bei denen das Werkzeug oder das Werkstück oder beide von einer Rundachse bewegt werden, soll der wirksame Bearbeitungsvorschub in gewohnter Weise als Bahnvorschub über den F-Wert programmiert werden können. Dazu muss für jede der beteiligten Rundachsen ein effektiver Radius (Bezugsradius) angegeben werden.

Die Einheit des Bezugsradius ist abhängig von der `G70`/`G71`/`G700`/`G710`-Einstellung.

Um zur Berechnung des Bahnvorschubs beizutragen, müssen alle mitwirkenden Achsen in den `FGROUP`-Befehl aufgenommen werden.

Um kompatibel zum Verhalten ohne `FGREF`-Programmierung zu bleiben, wird nach dem System-Hochlauf und bei RESET die Bewertung 1 Grad = 1 mm wirksam. Dies entspricht einem Bezugsradius von FGREF = 360 mm / (2π) = 57.296 mm.

**Hinweis**

Diese Voreinstellung ist unabhängig vom aktiven Grundsystem (MD10240 $MN_SCALING_SYSTEM_IS_METRIC) und von der aktuell wirksamen `G70`/`G71`/`G700`/`G710`-Einstellung.

Besonderheiten:

**Programmcode**

```
N100 FGROUP(X,Y,Z,A)
N110 G1 G91 A10 F100
N120 G1 G91 A10 X0.0001 F100
```

Bei dieser Programmierung wird der programmierte F-Wert in N110 als Rundachsvorschub in Grad/min bewertet, während die Vorschubbewertung in N120 abhängig von der aktuell wirksamen G70/G71/G700/G710-Einstellung entweder 100 inch/min oder 100 mm/min ist.

**ACHTUNG**

**Vorschubunterschied**

Die FGREF-Bewertung wirkt auch, wenn nur Rundachsen im Satz programmiert sind. Die gewohnte F-Wert-Interpretation als Grad/min gilt in diesem Fall nur, wenn der Radiusbezug entsprechend der FGREF-Voreinstellung ist:

- bei G71/G710: FGREF[A]=57.296
- bei G70/G700: FGREF[A]=57.296/25.4

### Bezugsradius lesen

Der Wert des Bezugsradius einer Rundachse kann über Systemvariablen gelesen werden:

- In Synchronaktionen oder mit Vorlauf-Stopp im Teileprogramm über die Systemvariable:

  $AA_FGREF[<Achse>] Aktueller Hauptlauf-Wert

- Ohne Vorlauf-Stopp im Teileprogramm über die Systemvariable:

  $PA_FGREF[<Achse>] Programmierter Wert

Sind keine Werte programmiert, liest man in beiden Variablen für Rundachsen die Voreinstellung 360 mm / (2π) = 57.296 mm (entspricht 1 mm pro Grad).

Für Linearachsen liest man in beiden Variablen immer den Wert 1 mm.

### Geschwindigkeitsbestimmende Bahnachsen lesen

Die an der Bahninterpolation beteiligten Achsen können über Systemvariablen gelesen werden:

- In Synchronaktionen oder mit Vorlauf-Stopp im Teileprogramm über die Systemvariablen:

  $AA_FGROUP[<Achse>] Liefert den Wert "1", wenn die angegebene Achse per Grundeinstellung oder durch FGROUP-Programmierung einen Einfluss auf die Bahngeschwindigkeit im aktuellen Hauptlaufsatz hat. Wenn nicht, dann liefert die Variable den Wert "0".

  $AC_FGROUP_MASK Liefert einen Bitschlüssel der mit FGROUP programmierten Kanalachsen, die zur Bahngeschwindigkeit beitragen sollen.

- Ohne Vorlauf-Stopp im Teileprogramm über die Systemvariablen:

| | |
|---|---|
| $PA_FGROUP[<Achse>] | Liefert den Wert "1", wenn die angegebene Achse per Grundeinstellung oder durch FGROUP-Programmierung einen Einfluss auf die Bahngeschwindigkeit hat. Wenn nicht, dann liefert die Variable den Wert "0". |
| $P_FGROUP_MASK | Liefert einen Bitschlüssel der mit FGROUP programmierten Kanalachsen, die zur Bahngeschwindigkeit beitragen sollen. |

**Bahnbezugsfaktoren für Orientierungsachsen mit FGREF**

Bei Orientierungsachsen ist die Wirkungsweise der FGREF[]-Faktoren davon abhängig, ob die Änderung der Orientierung des Werkzeugs entweder durch Rundachs- oder Vektorinterpolation erfolgt.

Bei **Rundachsinterpolation** werden die jeweiligen FGREF-Faktoren der Orientierungsachsen wie bei Rundachsen einzeln als Bezugsradius für die Wege der Achsen eingerechnet.

Bei **Vektorinterpolation** wird ein effektiver FGREF-Faktor wirksam, der als geometrischer Mittelwert aus den einzelnen FGREF-Faktoren bestimmt wird:

FGREF[effektiv] = n-teWurzel aus [(FGREF[A] * FGREF[B]...)]

mit: A: Achsbezeichner der 1. Orientierungsachse
B: Achsbezeichner der 2. Orientierungsachse
C: Achsbezeichner der 3. Orientierungsachse
n: Anzahl der Orientierungsachsen

Beispiel:

Für eine Standard-5-Achs-Transformation gibt es zwei Orientierungsachsen und somit berechnet sich der effektive Faktor als Wurzel aus dem Produkt der beiden axialen Faktoren:

FGREF[effektiv] = Quadratwurzel aus [(FGREF[A] * FGREF[B])]

**Hinweis**

Mit dem effektiven Faktor für Orientierungsachsen FGREF kann somit ein Bezugspunkt auf dem Werkzeug festgelegt werden, auf den sich der programmierte Bahnvorschub bezieht.

## 2.7.2 Positionierachsen verfahren (POS, POSA, POSP, FA, WAITP, WAITMC)

Positionierachsen werden unabhängig von Bahnachsen mit eigenem achsspezifischem Vorschub verfahren. Es gelten keine Interpolationsbefehle. Mit den Befehlen POS/POSA/POSP werden die Positionierachsen verfahren und gleichzeitig die Bewegungsabläufe koordiniert

Typische Beispiele für Positionierachsen sind:

- Palettenzuführeinrichtungen
- Messstationen

Mit WAITP kann im NC-Programm die Stelle gekennzeichnet werden, an der so lange gewartet werden soll, bis eine in einem früheren NC-Satz unter POSA programmierte Achse ihren Endpunkt erreicht hat.

Mit WAITMC wird bei Eintreffen der angegebenen Wartemarke augenblicklich der nächste NC-Satz eingewechselt.

## Syntax

```
POS[<Achse>]=<Position>
POSA[<Achse>]=<Position>
POSP[<Achse>]=(<Endposition>,<Teillänge>,<Modus>)
FA[<Achse>]=<Wert>
WAITP(<Achse>) ; Programmierung im eigenen NC-Satz!
WAITMC(<Wartemarke>)
```

## Bedeutung

| | | | |
|---|---|---|---|
| POS / POSA: | Positionierachse auf die angegebene Position verfahren<br>POS und POSA haben die gleiche Funktionalität, unterscheiden sich aber im Satzwechselverhalten:<br>• Mit POS wird der NC-Satz erst weitergeschaltet, wenn die anzufahrende Position erreicht ist.<br>• Mit POSA wird der NC-Satz weitergeschaltet, auch wenn die anzufahrende Position nicht erreicht ist. | | |
| | <Achse>: | Name der zu verfahrenden Achse (Kanal- oder Geometrieachsbezeichner) | |
| | <Position>: | Anzufahrende Achsposition | |
| | | Typ: | REAL |
| POSP: | Positionierachse in Teilstücken auf die angegebene Endposition verfahren | | |
| | <Endposition>: | Anzufahrende Achsendposition | |
| | <Teillänge>: | Länge eines Teilstücks | |
| | <Modus>: | Anfahrmodus | |
| | | = 0: | Für die beiden letzten Teilstücke erfolgt eine Aufteilung des verbleibenden Wegs bis zur Endposition auf zwei gleich große Reststücke (Vorbesetzung). |
| | | = 1: | Die Teillänge wird so angepasst, dass die Summe aller errechneten Teillängen genau den Weg zur Endposition ergibt. |
| | **Hinweis:**<br>POSP wird speziell für die Programmierung von Pendelbewegungen (Seite 968) eingesetzt. | | |

| | | | |
|---|---|---|---|
| FA: | Vorschub für die angegebene Positionierachse | | |
| | <Achse>: | Name der zu verfahrenden Achse (Kanal- oder Geometrieachsbezeichner) | |
| | <Wert>: | Vorschubgeschwindigkeit | |
| | | Einheit: | mm/min bzw. inch/min oder Grad/min |
| | **Hinweis:**<br>Pro NC-Satz können maximal 5 FA-Werte programmiert werden. | | |
| WAITP: | Warten auf das Verfahrende einer Positionierachse<br>Mit der Abarbeitung nachfolgender Sätze wird solange gewartet, bis die angegebene und in einem früheren NC-Satz mit POSA programmierte Positionierachse ihre Endposition erreicht hat (mit Genauhalt fein). | | |
| | <Achse>: | Name der Achse (Kanal- oder Geometrieachsbezeichner), für die der WAITP-Befehl gelten soll | |
| | **Hinweis:**<br>Mit WAITP kann eine Achse als Pendelachse oder für das Verfahren als konkurrierende Positionsachse (durch PLC) freigegeben werden. | | |
| WAITMC: | Warten auf das Eintreffen der angegebenen Wartemarke<br>Bei Eintreffen der Wartemarke wird sofort der nächste NC-Satz eingewechselt. | | |
| | <Wartemarke>: | Nummer der Wartemarke | |

**⚠ VORSICHT**

**Fahren mit POSA**

Wird in einem nachfolgenden Satz ein Befehl gelesen, der implizit Vorlaufstopp erzeugt, wird der nachfolgende Satz erst dann ausgeführt, wenn alle vorher aufbereiteten und gespeicherten Sätze vollständig abgearbeitet sind. Der vorherige Satz wird im Genauhalt (wie bei G9) angehalten.

## Beispiele

### Beispiel 1: Fahren mit POSA und Zugriff auf Zustandsdaten der Maschine

Beim Zugriff auf Zustandsdaten der Maschine ($A…) erzeugt die Steuerung internen Vorlaufstopp. Die Bearbeitung wird angehalten, bis alle vorher aufbereiteten und gespeicherten Sätze vollständig abgearbeitet sind.

```
Programmcode                                    Kommentar
N40 POSA[X]=100
N50 IF $AA_IM[X]==R100 GOTOF MARKE1             ; Zugriff auf Zustandsdaten der Ma-
                                                schine.
N60 G0 Y100
N70 WAITP(X)
N80 MARKE1:
N...
```

### Beispiel 2: Warten auf Verfahrende mit WAITP

Palettenzuführeinrichtung
Achse U: Palettenspeicher
Transport der Werkstückpalette in den Arbeitsraum
Achse V: Transfersystem zu einer Messstation, in der prozessbegleitende Stichprobenkontrollen durchgeführt werden

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 FA[U]=100 FA[V]=100` | `; Achsspezifische Vorschubangaben für die einzelnen Positionierachsen U und V.` |
| `N20 POSA[V]=90 POSA[U]=100 G0 X50 Y70` | `; Positionier- und Bahnachsen verfahren.` |
| `N50 WAITP(U)` | `; Der Programmablauf wird erst fortgesetzt, wenn die Achse U die in N20 programmierte Position erreicht hat.` |
| … | |

## Weitere Informationen

### Fahren mit POSA

Die Satzweiterschaltung bzw. der Programmablauf wird durch `POSA` nicht beeinflusst. Die Bewegung zum Endpunkt kann parallel zum Abarbeiten von nachfolgenden NC-Sätzen durchgeführt werden.

### Fahren mit POS

Die Satzweiterschaltung wird erst ausgeführt, wenn alle unter `POS` programmierten Achsen ihre Endpositionen erreicht haben.

### Warten auf Verfahrende mit WAITP

Nach einem `WAITP` gilt die Achse so lange als nicht mehr vom NC-Programm belegt, bis sie neu programmiert wird. Diese Achse kann dann durch die PLC als Positionierachse oder vom NC-Programm/PLC oder HMI als Pendelachse betrieben werden.

### Satzwechsel in der Bremsrampe mit IPOBRKA und WAITMC

Eine Achse wird nur abgebremst, wenn die Wartemarke noch nicht erreicht ist, oder ein anderes Satzendekriterium den Satzwechsel verhindert. Nach einem `WAITMC` startet die Achse sofort durch, falls nicht ein anderes Satzendekriterium den Satzwechsel verhindert.

## 2.7.3 Lagegeregelter Spindelbetrieb (SPCON, SPCOF)

Mit den Befehlen SPCON bzw. SPCOF wird der Lageregelbetrieb der Spindel explizit ein- bzw. ausgeschaltet.

**Hinweis**

Das Einschalten des Lageregelbetriebs mit SPCON benötigt maximal drei Lagereglertakte.

### Syntax

```
SPCON
SPCON(<n>)
SPCON(<n>,<m>,...)
SPCOF
SPCOF(<n>)
SPCOF(<n>,<m>,...)
```

### Bedeutung

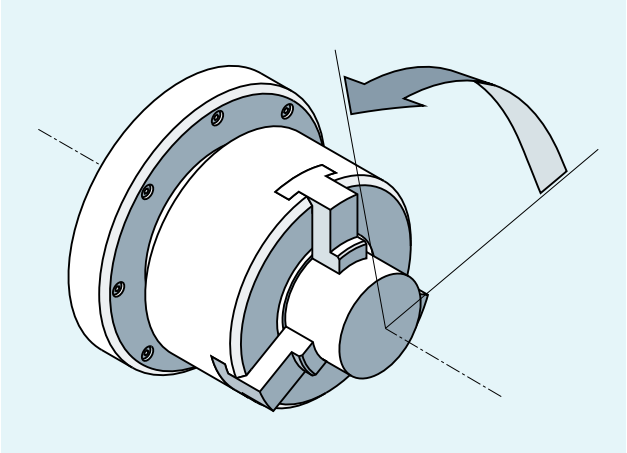
| | |
|---|---|
| `SPCON`: | Lageregelbetrieb einschalten<br>Die angegebene Spindel wird von Drehzahlregelung in Lageregelung umgeschaltet.<br>`SPCON` wirkt modal und bleibt bis `SPCOF` erhalten. |
| `SPCOF`: | Lageregelbetrieb ausschalten<br>Die angegebene Spindel wird von Lageregelung in Drehzahlregelung umgeschaltet. |
| `<n>,<m>,...`: | Spindelnummern<br>Ohne Angabe einer Spindelnummer: Masterspindel des Kanals |

**Hinweis**

Bei einer Synchronspindel mit Sollwertkopplung darf die Leitspindel nicht mit SPCOF in den drehzahlgeregelten Betrieb geschaltet werden.

## 2.7.4 Spindeln positionieren (SPOS, SPOSA, M19, M70, WAITS)

Mit SPOS, SPOSA oder M19 können Spindeln auf bestimmte Winkelstellungen positioniert werden, z. B. beim Werkzeugwechsel.



SPOS, SPOSA und M19 bewirken eine temporäre Umschaltung in den Lageregelbetrieb bis zum nächsten M3/M4/M5/M41 … M45.

**Positionieren im Achsbetrieb**

Die Spindel kann auch unter ihrer im Maschinendatum bestimmten Adresse als Bahn-, Synchron- oder Positionierachse verfahren werden. Mit Angabe des Achsbezeichners befindet sich die Spindel im Achsbetrieb. Mit M70 wird die Spindel direkt in den Achsbetrieb geschaltet.

**Positionierende**

Das Bewegungsendekriterium beim Positionieren der Spindel ist über FINEA, CORSEA, IPOENDA oder IPOBRKA programmierbar.

Sind die Bewegungsendekriterien für alle im Satz zu bearbeitenden Spindeln bzw. Achsen und außerdem das Satzwechselkriterium für die Bahninterpolation erfüllt, so erfolgt der Satzwechsel.

**Synchronisation**

Um Spindelbewegungen zu synchronisieren, kann mit WAITS bis zum Erreichen der Spindelposition gewartet werden.

### Voraussetzungen

Die zu positionierende Spindel muss im lagegeregelten Betrieb arbeiten können.

### Syntax

Spindel positionieren:

`SPOS=<Wert>` / `SPOS[<n>]=<Wert>`

`SPOSA=<Wert>` / `SPOSA[<n>]=<Wert>`

`M19` / `M<n>=19`

Spindel in den Achsbetrieb umschalten:

`M70` / `M<n>=70`

Bewegungsendekriterium festlegen:

`FINEA` / `FINEA[S<n>]`

`COARSEA` / `COARSEA[S<n>]`

`IPOENDA` / `IPOENDA[S<n>]`

`IPOBRKA` / `IPOBRKA(<Achse>[,<Zeitpunkt>])` ; Programmierung im eigenen NC-Satz!

Spindelbewegungen synchronisieren:

`WAITS` / `WAITS(<n>,<m>)` ; Programmierung im eigenen NC-Satz!

## Bedeutung

| | | | | |
|---|---|---|---|---|
| `SPOS` / `SPOSA`: | Spindel auf die angegebene Winkelstellung positionieren<br>SPOS und SPOSA haben die gleiche Funktionalität, unterscheiden sich aber im Satzwechselverhalten:<br>• Mit SPOS wird der NC-Satz wird erst weitergeschaltet, wenn die Position erreicht ist.<br>• Mit SPOSA wird der NC-Satz weitergeschaltet, auch wenn die Position nicht erreicht ist. | | | |
| | `<n>`: | Nummer der Spindel, die positioniert werden soll.<br>Ohne Angabe einer Spindelnummer oder mit Spindelnummer "0" bezieht sich SPOS bzw. SPOSA auf die Masterspindel. | | |
| | `<Wert>`: | Winkelstellung, auf die die Spindel positioniert werden soll | | |
| | | Einheit: | Grad | |
| | | Typ: | REAL | |
| | | Für die Programmierung des Positionsanfahrmodus gibt es folgende Möglichkeiten: | | |
| | | `=AC(<Wert>)`: | Absolute Maßangabe | |
| | | | Wertebereich: | 0 … 359,9999 |
| | | `=IC(<Wert>)`: | Inkrementelle Maßangabe | |
| | | | Wertebereich: | 0 … ±99 999,999 |
| | | `=DC(<Wert>)`: | Anfahren auf direktem Weg auf Absolutwert | |
| | | `=ACN(<Wert>)`: | Absolute Maßangabe, Anfahren in negativer Richtung | |
| | | `=ACP(<Wert>)`: | Absolute Maßangabe, Anfahren in positiver Richtung | |
| | | `=<Wert>`: | wie DC(<Wert>) | |
| `M<n>=19`: | Masterspindel (M19 oder M0=19) oder Spindel mit Nummer <n> (M<n>=19) auf die mit SD43240 $SA_M19_SPOS vorgegebene Winkelstellung mit dem in SD43250 $SA_M19_SPOSMODE vorgegebenen Positionsanfahrmodus positionieren<br>Der NC-Satz wird erst weitergeschaltet, wenn die Position erreicht ist. | | | |

| | | | | |
|---|---|---|---|---|
| M<n>=70: | Masterspindel (M70 oder M0=70) oder Spindel mit Nummer <n> (M<n>=70) in den Achsbetrieb umschalten<br>Es wird keine definierte Position angefahren. Der NC-Satz wird weitergeschaltet, wenn die Umschaltung ausgeführt wurde. | | | |
| FINEA: | Bewegungsende bei Erreichen von "Genauhalt fein" | | | |
| COARSEA: | Bewegungsende bei Erreichen von "Genauhalt grob" | | | |
| IPOENDA: | Bewegungsende bei Erreichen von "Interpolator-Stopp" | | | |
| S<n>: | Spindel, für die das programmierte Bewegungsendekriterium wirksam sein soll | | | |
| | <n>: | Spindelnummer | | |
| | Ohne Angabe einer Spindel [S<n>] oder mit Spindelnummer "0" bezieht sich das programmierte Bewegungsendekriterium auf die Masterspindel. | | | |
| IPOBRKA: | Satzwechsel in der Bremsrampe möglich | | | |
| | <Achse>: | Kanalachsbezeichner | | |
| | <Zeitpunkt>: | Zeitpunkt des Satzwechsels bezogen auf die Bremsrampe | | |
| | | Einheit: | Prozent | |
| | | Wertebereich: | 100 (Einsatzzeitpunkt der Bremsrampe) … 0 (Ende der Bremsrampe) | |
| | | Ohne Angabe des Parameters <Zeitpunkt> wird der aktuelle Wert des Settingdatums wirksam:<br>SD43600 $SA_IPOBRAKE_BLOCK_EXCHANGE<br>**Hinweis:**<br>IBOBRKA mit Zeitpunkt "0" ist identisch mit IPOENDA. | | |
| | | | | |
| WAITS: | Synchronisationsbefehl für die angegebene(n) Spindel(n)<br>Mit der Abarbeitung nachfolgender Sätze wird solange gewartet, bis die angegebene(n) und in einem früheren NC-Satz mit SPOSA programmierte(n) Spindel(n) ihre Position(en) erreicht haben (mit Genauhalt fein). | | | |
| | WAITS nach M5: | Warten, bis die angegebene(n) Spindel(n) stehen. | | |
| | WAITS nach M3/M4: | Warten, bis die angegebene(n) Spindel(n) ihre Solldrehzahl erreicht haben. | | |
| | <n>,<m>: | Nummern der Spindeln, für die der Synchronisationsbefehl gelten soll<br>Ohne Angabe einer Spindelnummer oder mit Spindelnummer "0" bezieht sich WAITS auf die Masterspindel. | | |

**Hinweis**

Pro NC-Satz sind 3 Spindelpositionsangaben möglich.

**Hinweis**

Bei Inkrementeller Maßangabe IC(<Wert>) ist die Spindelpositionierung über mehrere Umdrehungen möglich.

**Hinweis**

Wurde vor SPOS die Lageregelung mit SPCON eingeschaltet, bleibt diese bis SPCOF erhalten.

**Hinweis**

Die Steuerung erkennt aufgrund der Programmierabfolge selbständig den Übergang in den Achsbetrieb. Die explizite Programmierung von M70 im Teileprogramm ist daher grundsätzlich nicht mehr notwendig. M70 kann jedoch weiterhin programmiert werden, um z. B. die Lesbarkeit des Teileprogramms zu erhöhen.

## Weitere Informationen

### Positionieren mit SPOSA

Die Satzweiterschaltung bzw. der Programmablauf wird durch SPOSA nicht beeinflusst. Die Spindelpositionierung kann parallel zum Abarbeiten von nachfolgenden NC-Sätzen durchgeführt werden. Der Satzwechsel erfolgt, wenn alle im Satz programmierten Funktionen (außer der Spindel) ihr Satzendekriterium erreicht haben. Die Spindelpositionierung kann sich dabei über mehrere Sätze erstrecken (siehe WAITS).

**Hinweis**

Wird in einem nachfolgenden Satz ein Befehl gelesen, der implizit Vorlaufstopp erzeugt, so wird die Bearbeitung in diesem Satz so lange angehalten, bis alle positionierenden Spindeln stehen.

### Positionieren mit SPOS / M19

Die Satzweiterschaltung wird erst ausgeführt, wenn alle im Satz programmierten Funktionen ihr Satzendekriterium erreicht haben (z. B. alle Hilfsfunktionen von der PLC quittiert wurden, alle Achsen ihren Endpunkt erreicht haben) und die Spindel die programmierte Position erreicht hat.

Geschwindigkeit der Bewegungen:

Die Geschwindigkeit und das Verzögerungsverhalten für die Positionierung sind in Maschinendaten hinterlegt. Die projektierten Werte können durch Programmierung oder durch Synchronaktionen geändert werden, siehe:

- Vorschub für Positionierachsen/Spindeln (FA, FPR, FPRAON, FPRAOF) (Seite 126)
- Programmierbare Beschleunigungskorrektur (ACC) (Seite 130)

Angabe der Spindelpositionen:

Da die Befehle `G90`/`G91` hier nicht wirken, gelten explizit die entsprechenden Maßangaben wie z. B. `AC`, `IC`, `DC`, `ACN`, `ACP`. Ohne Angaben wird automatisch wie bei der `DC`-Angabe verfahren.

### Spindelbewegungen synchronisieren mit WAITS

Mit `WAITS` kann im NC-Programm eine Stelle gekennzeichnet werden, an der solange gewartet wird, bis eine oder mehrere in einem früheren NC-Satz unter `SPOSA` programmierte Spindeln ihre Position erreicht haben.

**Beispiel:**

| **Programmcode** | **Kommentar** |
|---|---|
| N10 SPOSA[2]=180 SPOSA[3]=0 | |
| ... | |
| N40 WAITS(2,3) | ; Im Satz wird so lange gewartet, bis Spindeln 2 und 3 die in Satz N10 angegebenen Positionen erreicht haben. |

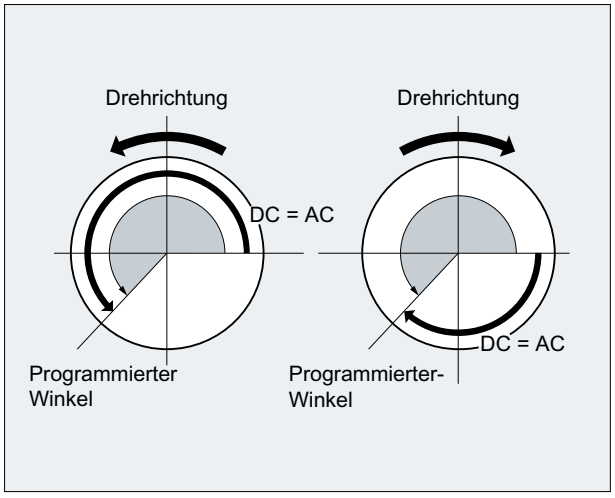
Nach M5 kann mit WAITS darauf gewartet werden, dass die Spindel(n) zum Stillstand gekommen ist/sind. Nach M3/M4 kann mit WAITS darauf gewartet werden, dass die Spindel(n) die vorgegebene Drehzahl/Drehrichtung erreicht hat/haben.

**Hinweis**

Ist die Spindel noch nicht mit Synchronmarken synchronisiert, dann wird die positive Drehrichtung aus dem Maschinendatum entnommen (Auslieferungszustand).

### Spindel aus der Drehung (M3/M4) positionieren

Bei eingeschaltetem M3 oder M4 kommt die Spindel auf dem programmierten Wert zum Stillstand.




Zwischen DC- und AC-Angabe besteht kein Unterschied. In beiden Fällen wird in der durch M3/M4 gewählten Drehrichtung bis zur absoluten Endposition weitergedreht. Bei ACN und ACP wird ggf. gebremst und die entsprechende Anfahrrichtung eingehalten. Bei der IC-Angabe wird, ausgehend von der aktuellen Spindelposition, um den angegebenen Wert weitergedreht.

### Spindel aus dem Stillstand (M5) positionieren

Der programmierte Weg wird aus dem Stillstand (M5) den Vorgaben entsprechend exakt abgefahren.

## 2.7.5 Vorschub für Positionierachsen/Spindeln (FA, FPR, FPRAON, FPRAOF)

Daneben besteht die Möglichkeit, den Umdrehungsvorschub für Bahn- und Synchronachsen oder für einzelne Positionierachsen/Spindeln von einer anderen Rundachse oder Spindel abzuleiten.

Positionierachsen, wie z. B. Werkstücktransportsysteme, Revolver oder Lünetten, werden unabhängig von Bahn- und Synchronachsen verfahren. Deshalb wird für jede Positionierachse ein eigener Vorschub definiert.

Auch für Spindeln kann ein eigener axialer Vorschub programmiert werden.

### Syntax

Vorschub für Positionierachse:
`FA[<Achse>]=…`

Axialer Vorschub für Spindel:
`FA[SPI(<n>)]=…`
`FA[S<n>]=…`

Umdrehungsvorschub für Bahn-/Synchronachsen ableiten:

```
FPR(<Rundachse>)
FPR(SPI(<n>))
FPR(S<n>)
```

Umdrehungsvorschub für Positionierachsen/Spindeln ableiten:

```
FPRAON(<Achse>,<Rundachse>)
FPRAON(<Achse>,SPI(<n>))
FPRAON(<Achse>,S<n>)
FPRAON(SPI(<n>),<Rundachse>)
FPRAON(S<n>,<Rundachse>)
FPRAON(SPI(<n>),SPI(<n>))
FPRAON(S<n>,S<n>)
FPRAOF(<Achse>,SPI(<n>),…)
FPRAOF(<Achse>,S<n>,…)
```

### Bedeutung

| | | |
|---|---|---|
| `FA[...]=...`: | Vorschub für die angegebene Positionierachse bzw. Positioniergeschwindigkeit (axialer Vorschub) für die angegebene Spindel | |
| | Einheit: | mm/min bzw. inch/min oder Grad/min |
| | Wertebereich: | … 999 999,999 mm/min, Grad/min<br>… 39 999,9999 inch/min |

| | |
|---|---|
| FPR(...): | Mit FPR wird die Rundachse (<Rundachse>) oder Spindel (SPI(<n>) / S<n>) gekennzeichnet, von der der unter G95 programmierte Umdrehungsvorschub für den Umdrehungsvorschub der Bahn- und Synchronachsen abgeleitet werden soll. |
| FPRAON(...): | Umdrehungsvorschub für Positionierachsen und Spindeln ableiten<br>Der erste Parameter (<Achse> / SPI(<n>) / S<n>) kennzeichnet die Positionierachse/Spindel, die mit Umdrehungsvorschub verfahren soll.<br>Der zweite Parameter (<Rundachse> / SPI(<n>) / S<n>) kennzeichnet die Rundachse/Spindel, von der der Umdrehungsvorschub abgeleitet werden soll.<br>**Hinweis:**<br>Der zweite Parameter kann auch entfallen, dann wird der Vorschub von der Masterspindel abgeleitet. |
| FPRAOF(...): | Mit FPRAOF wird der abgeleitete Umdrehungsvorschub für die angegebenen Achsen oder Spindeln abgewählt. |
| <Achse>: | Achsbezeichner (Positionier- oder Geometrieachse) |
| SPI(<n>) / S<n>: | Spindelbezeichner<br>SPI(<n>) und S<n> sind funktionell identisch.<br><n>: Spindelnummer<br>**Hinweis:**<br>SPI konvertiert Spindelnummer in Achsbezeichner. Der Übergabeparameter (<n>) muss eine gültige Spindelnummer enthalten. |

**Hinweis**

Der programmierte Vorschub FA[...] ist modal wirksam.

Pro NC-Satz können max. 5 Vorschübe für Positionierachsen/Spindeln programmiert werden.

**Hinweis**

Der abgeleitete Vorschub errechnet sich nach folgender Formel:

Abgeleiteter Vorschub = Programmierter Vorschub * Betrag Leitvorschub

## Beispiele

### Beispiel 1: Synchronspindelkopplung

Bei Synchronspindelkopplung kann die Positioniergeschwindigkeit der Folgespindel unabhängig von der Leitspindel programmiert werden, z. B. zum Positionieren.

```
Programmcode          Kommentar
...
FA[S2]=100            ; Positioniergeschwindigkeit der Folgespindel (Spindel 2)
                      = 100 Grad/min
...
```

### Beispiel 2: Abgeleiteter Umdrehungsvorschub für Bahnachsen

Die Bahnachsen X, Y sollen mit Umdrehungsvorschub verfahren werden, der sich von der Rundachse A ableitet:

```
Programmcode
...
N40 FPR(A)
N50 G95 X50 Y50 F500
...
```

### Beispiel 3: Umdrehungsvorschub für Masterspindel ableiten

| Programmcode | Kommentar |
|---|---|
| N30 FPRAON(S1,S2) | ; Der Umdrehungsvorschub für die Masterspindel (S1) soll von Spindel 2 abgeleitet werden. |
| N40 SPOS=150 | ; Masterspindel positionieren. |
| N50 FPRAOF(S1) | ; Abgeleiteten Umdrehungsvorschub für die Masterspindel abwählen. |

### Beispiel 4: Umdrehungsvorschub für Positionierachse ableiten

| Programmcode | Kommentar |
|---|---|
| N30 FPRAON(X) | ; Der Umdrehungsvorschub für die Positionierachse X soll von der Masterspindel abgeleitet werden. |
| N40 POS[X]=50 FA[X]=500 | ; Die Positionierachse fährt mit 500 mm/Umdrehung der Masterspindel. |
| N50 FPRAOF(X) | |

## Weitere Informationen

### FA[…]

Es gilt immer die Vorschubart G94. Ist G70/G71 aktiv, so richtet sich die Maßeinheit metrisch/inch nach der Voreinstellung im Maschinendatum. Mit G700/G710 kann die Maßeinheit im Programm verändert werden.

**Hinweis**

Wird kein FA programmiert, gilt der im Maschinendatum eingestellte Wert.

### FPR(…)

Mit FPR kann als Erweiterung des G95-Befehls (Umdrehungsvorschub bezogen auf Masterspindel) der Umdrehungsvorschub auch von einer beliebigen Spindel oder Rundachse abgeleitet werden. G95 FPR(…) gilt für Bahn- und Synchronachsen.

Falls die mit FPR gekennzeichnete Rundachse/Spindel mit Lageregelung arbeitet, gilt Sollwertkopplung, ansonsten Istwertkopplung.

### FPRAON(...)

Mit FPRAON lässt sich axial für Positionierachsen und Spindeln der Umdrehungsvorschub vom augenblicklichen Vorschub einer anderen Rundachse oder Spindel ableiten.

### FPRAOF(...)

Mit FPRAOF lässt sich der Umdrehungsvorschub für eine oder gleichzeitig mehrere Achsen/ Spindeln ausschalten.

## 2.7.6 Programmierbare Vorschubkorrektur (OVR, OVRRAP, OVRA)

Die Geschwindigkeit von Bahn-/Positionierachsen und Spindeln kann im NC-Programm modifiziert werden.

### Syntax

```
OVR=<Wert>
OVRRAP=<Wert>
OVRA[<Achse>]=<Wert>
OVRA[SPI(<n>)]=<Wert>
OVRA[S<n>]=<Wert>
```

### Bedeutung

| | | |
|---|---|---|
| OVR: | Vorschubänderung für Bahnvorschub F | |
| OVRRAP: | Vorschubänderung für Eilganggeschwindigkeit | |
| OVRA: | Vorschubänderung für Positioniervorschub FA bzw. für Spindeldrehzahl S | |
| | | |
| <Achse>: | Achsbezeichner (Positionier- oder Geometrieachse) | |
| SPI(<n>) / S<n>: | Spindelbezeichner<br>SPI(<n>) und S<n> sind funktionell identisch. | |
| | <n>: | Spindelnummer |
| | **Hinweis:**<br>SPI konvertiert Spindelnummer in Achsbezeichner. Der Übergabeparameter (<n>) muss eine gültige Spindelnummer enthalten. | |
| | | |
| <Wert>: | Vorschubänderung in Prozent<br>Der Wert bezieht sich auf bzw. überlagert sich mit dem an der Maschinensteuertafel eingestellten Vorschub-Override. | |
| | Wertebereich: | 0 … 200%, ganzzahlig |
| | **Hinweis:**<br>Bei Bahn- und Eilgangkorrektur werden die in Maschinendaten eingestellten maximalen Geschwindigkeiten nicht überschritten. | |

## 2.7.7 Programmierbare Beschleunigungskorrektur (ACC)

In kritischen Programmabschnitten kann es notwendig sein, die Beschleunigung unter die maximal möglichen Werte zu beschränken, um z. B. mechanische Schwingungen zu vermeiden.

Mit der programmierbaren Beschleunigungskorrektur kann für jede Bahnachse oder Spindel die Beschleunigung per Befehl im NC-Programm verändert werden. Die Begrenzung wirkt in allen Interpolationsarten. Als 100 % Beschleunigung gelten die in den Maschinendaten festgelegten Werte.

### Syntax

```
ACC[<Achse>]=<Wert>
ACC[SPI(<n>)]=<Wert>
ACC(S<n>)=<Wert>
```

Ausschalten:
```
ACC[...]=100
```

### Syntax

| | | |
|---|---|---|
| ACC: | Beschleunigungsänderung für die angegebene Bahnachse bzw. Drehzahländerung für die angegebene Spindel | |
| | | |
| <Achse>: | Kanalachsname der Bahnachse | |
| SPI(<n>) / S<n>: | Spindelbezeichner<br>SPI(<n>) und S<n> sind funktionell identisch. | |
| | <n>: | Spindelnummer |
| | **Hinweis:**<br>SPI konvertiert Spindelnummer in Achsbezeichner. Der Übergabeparameter (<n>) muss eine gültige Spindelnummer enthalten. | |
| <Wert>: | Beschleunigungsänderung in Prozent<br>Der Wert bezieht sich auf bzw. überlagert sich mit dem an der Maschinensteuertafel eingestellten Vorschub-Override. | |
| | Wertebereich: | 1 … 200%, ganzzahlig |

### Hinweis

Bei einer größeren Beschleunigung können die vom Maschinenhersteller zulässigen Werte überschritten werden.

### Beispiel

| **Programmcode** | **Kommentar** |
|---|---|
| N50 ACC[X]=80 | ; Der Achsschlitten in X-Richtung soll nur mit 80% Beschleunigung gefahren werden. |
| N60 ACC[SPI(1)]=50 | ; Spindel 1 soll nur mit 50% des Beschleunigungsvermögens beschleunigen bzw. bremsen. |

## Weitere Informationen

### Mit ACC programmierte Beschleunigungskorrektur

Die mit ACC[...] programmierte Beschleunigungskorrektur wird immer wie in der Systemvariablen $AA_ACC bei der Ausgabe berücksichtigt. Das Auslesen im Teileprogramm und in Synchronaktionen findet zu verschiedenen Zeitpunkten in der NC-Verarbeitung statt.

### Im Teileprogramm

Der im Teileprogramm geschriebene Wert wird nur dann in der Systemvariablen $AA_ACC wie im Teileprogramm geschrieben berücksichtigt, wenn ACC zwischenzeitlich nicht von einer Synchronaktion verändert wurde.

### In Sychronaktionen

Entsprechend gilt: Der einer Synchronaktion geschriebene Wert wird nur dann in der Systemvariablen $AA_ACC wie der Synchronaktion geschrieben berücksichtigt, wenn ACC zwischenzeitlich nicht von einem Teileprogramm verändert wurde.

Die vorgegebene Beschleunigung kann auch über Synchronaktionen verändert werden.

Beispiel:

**Programmcode**

```
...
N100 EVERY $A_IN[1] DO POS[X]=50 FA[X]=2000 ACC[X]=140
```

Der aktuelle Beschleunigungswert kann mit der Systemvariablen $AA_ACC[<Achse>] abgefragt werden. Über Maschinendatum kann eingestellt werden, ob bei RESET/ Teileprogrammende der zuletzt gesetzte ACC-Wert oder 100% gelten soll.

## 2.7.8 Vorschub mit Handradüberlagerung (FD, FDA)

Mit den Befehlen FD und FDA können Achsen während des Teileprogrammablaufs mit Handrädern verfahren werden. Die programmierten Verfahrbewegungen der Achsen werden dabei mit den als Weg- oder Geschwindigkeitsvorgaben bewerteten Handradpulsen überlagert.

### Bahnachsen

Bei Bahnachsen kann der programmierte Bahnvorschub überlagert werden. Ausgewertet wird dabei das Handrad der 1. Geometrieachse des Kanals. Die drehrichtungsabhängig bewerteten Handradpulse pro IPO-Takt entsprechen der zu überlagernden Bahngeschwindigkeit. Die durch die Handradüberlagerung erreichbaren Bahngeschwindigkeitsgrenzwerte sind:

- Minimum: 0
- Maximum: Maschinendatengrenzwerte der an der Verfahrbewegung beteiligten Bahnachsen

**Hinweis**

**Bahnvorschub**

Der Bahnvorschub F und der Handradvorschub FD dürfen nicht zusammen in einem NC-Satz programmiert werden.

**Positionierachsen**
Bei Positionierachsen können axial der Verfahrweg oder die Geschwindigkeit überlagert werden. Ausgewertet wird dabei das der Achse zugeordnete Handrad.

- Wegüberlagerung
  Die drehrichtungsabhängig bewerteten Handradpulse entsprechen dem zu verfahrenden Weg der Achse. Es werden dabei nur Handradpulse in Richtung auf die programmierte Position berücksichtigt.
- Geschwindigkeitsüberlagerung
  Die drehrichtungsabhängig bewerteten Handradpulse pro IPO-Takt entsprechen der zu überlagernden axialen Geschwindigkeit. Die durch die Handradüberlagerung erreichbaren Bahngeschwindigkeitsgrenzwerte sind:
  - Minimum: 0
  - Maximum: Maschinendatengrenzwerte der Positionierachse

## Syntax

```
FD=<Geschwindigkeit>
FDA[<Achse>]=<Geschwindigkeit>
```

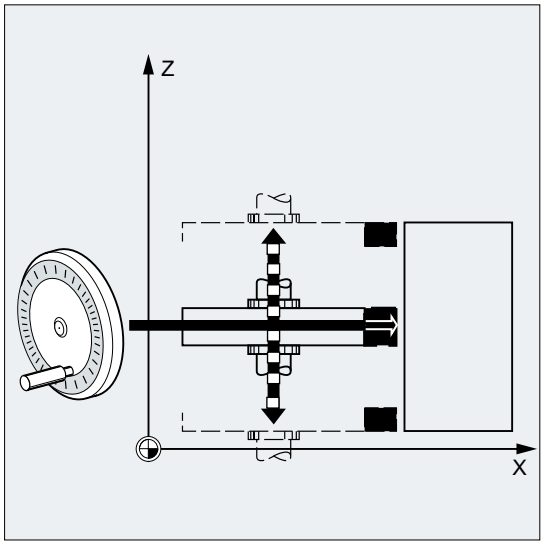
## Bedeutung

| | |
|---|---|
| `FD=< Geschwindigkeit >:` | Bahnvorschub und Freigabe der Geschwindigkeitsüberlagerung durch Handrad.<br><Geschwindigkeit>:<br>• Wert = 0: Nicht erlaubt!<br>• Wert ≠ 0: Bahngeschwindigkeit |
| `FDA[<Achse>]=<Geschwindigkeit>:` | axialer Vorschub<br><Geschwindigkeit>:<br>• Wert = 0: Wegvorgabe durch Handrad<br>• Wert ≠ 0: axiale Geschwindigkeit |
| `<Achse>`: | Achsbezeichner der Positionierachse |

**Hinweis**

`FD` und `FDA` sind satzweise wirksam.

## Beispiel




Wegvorgabe: Die in Z-Richtung pendelnde Schleifscheibe wird per Handrad in X-Richtung an das Werkstück gefahren.

Hierbei kann der Bediener manuell bis zum gleichmäßigen Funkenflug zustellen. Durch Aktivieren von "Restweglöschen" wird in den nächsten NC-Satz gewechselt und im AUTOMATIK-Betrieb weiter gearbeitet.

## Weitere Informationen

### Bahnachsen mit Geschwindigkeitsüberlagerung verfahren ( FD=<Geschwindigkeit> )

Für den Teileprogrammsatz in dem die Bahngeschwindigkeitsüberlagerung programmiert ist, müssen folgende Voraussetzungen erfüllt sein:

- Wegbefehl `G1`, `G2` oder `G3` aktiv
- Genauhalt `G60` aktiv
- Linearvorschub `G94` aktiv

### Vorschub-Override

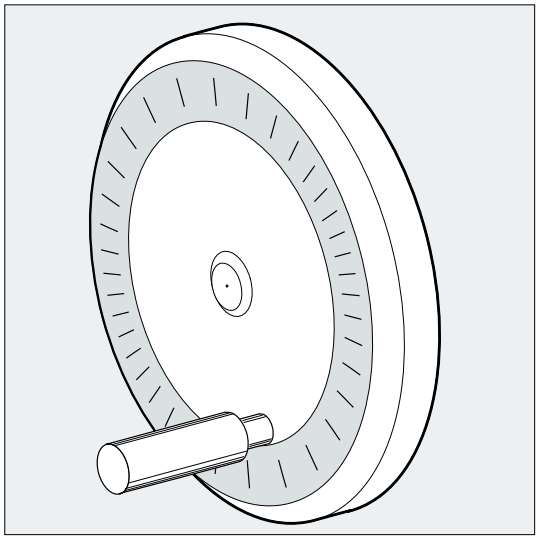
Der Vorschub-Override wirkt nur auf die programmierte Bahngeschwindigkeit, nicht auf den per Handrad erzeugten Geschwindigkeitsanteil (Ausnahme: Vorschub-Override = 0).

### Beispiel:

| **Programmcode** | **Beschreibung** |
|---|---|
| `N10 X… Y… F500` | `; Bahnvorschub = 500 mm/min` |
| `N20 X… Y… FD=700` | `; Bahnvorschub = 700 mm/min und Geschwindigkeitsüberlagerung durch Handrad.` |
| | `; In N20 wird von 500 auf 700 mm/min beschleunigt. Über Handrad kann drehrichtungsabhängig die Bahngeschwindigkeit zwischen 0 und Maximalwert (Maschinendaten) verändert werden.` |

### Positionierachsen mit Wegvorgabe verfahren ( FDA[<Achse>]=0 )

Im NC-Satz mit programmiertem `FDA[<Achse>]=0` wird der Vorschub auf Null gesetzt, so dass vom Programm her keine Fahrbewegung erfolgt. Die programmierte Fahrbewegung zur Zielposition wird jetzt ausschließlich vom Bediener durch Drehen des Handrads gesteuert.



**Beispiel:**

| Programmcode | Beschreibung |
|---|---|
| ... | |
| N20 POS[V]=90 FDA[V]=0 | ; Zielposition = 90 mm, axialer Vorschub = 0 mm/min und |
| | Wegüberlagerung durch Handrad. |
| | ; Geschwindigkeit der Achse V am Satzanfang = 0 mm/min. |
| | ; Weg- und Geschwindigkeitsvorgabe erfolgt durch Handradpulse |

Bewegungsrichtung, Fahrgeschwindigkeit:
Die Achsen fahren vorzeichenrichtig den per Handradpulse vorgegebenen Weg. Je nach Drehrichtung kann vorwärts und rückwärts gefahren werden. Je schneller das Handrad gedreht wird, desto höher die Verfahrgeschwindigkeit.

Verfahrbereich:
Der Verfahrbereich wird durch die Startposition und den programmierten Endpunkt begrenzt.

### Positionierachsen mit Geschwindigkeitsüberl. verfahren ( FDA[<Achse>]=<Geschwindigkeit> )

Im NC-Satz mit programmiertem `FDA[…]=…` wird der Vorschub vom zuletzt programmierten `FA`-Wert auf den unter `FDA` programmierten Wert beschleunigt bzw. verzögert. Ausgehend vom aktuellen Vorschub `FDA` kann die programmierte Bewegung zur Zielposition durch Drehen des Handrades beschleunigt oder bis Null verzögert werden. Als Maximalgeschwindigkeit gelten die in Maschinendaten parametrierten Werte.

Beispiel:

| Programmcode | Beschreibung |
|---|---|
| N10 POS[V]=… FA[V]=100 | ; axialer Vorschub = 100 mm/min |

| Programmcode | Beschreibung |
|---|---|
| N20 POS[V]=100 FAD[V]=200 | ; axiale Zielposition = 100, axialer Vorschub = 200 mm/min<br>; und Geschwindigkeitsüberlagerung durch Handrad.<br>; In N20 wird von 100 auf 200 mm/min beschleunigt. Über<br>; Handrad kann drehrichtungsabhängig die Geschwindigkeit<br>; zwischen 0 und Maximalwert (Maschinendaten) verändert<br>werden. |

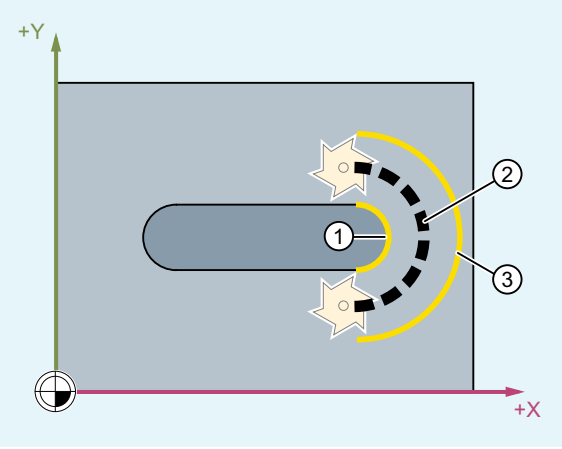
Verfahrbereich:
Der Verfahrbereich wird durch die Startposition und den programmierten Endpunkt begrenzt.

## 2.7.9 Vorschuboptimierung bei gekrümmten Bahnstücken (CFTCP, CFC, CFIN)

Der programmierte Vorschub bezieht sich bei eingeschaltetem Korrekturbetrieb G41/G42 für den Fräser-Radius zunächst auf die Fräsermittelpunktsbahn (vgl. Kapitel "Koordinatentransformationen (Frames) (Seite 308)").

Beim Fräsen eines Kreises (gleiches gilt für Polynom- und Spline-Interpolation) verändert sich der Vorschub am Fräserrand unter Umständen so stark, dass das Bearbeitungsergebnis darunter leidet.

Beispiel: Fräsen eines kleinen Außenradius mit einem größeren Werkzeug. Der Weg, den die Außenseite des Fräsers zurücklegen muss, ist sehr viel größer als der Weg entlang der Kontur.




Hierdurch wird an der Kontur mit einem sehr kleinen Vorschub gearbeitet. Um solche Effekte zu verhindern, sollte bei gekrümmten Konturen der Vorschub entsprechend geregelt werden.

### Syntax

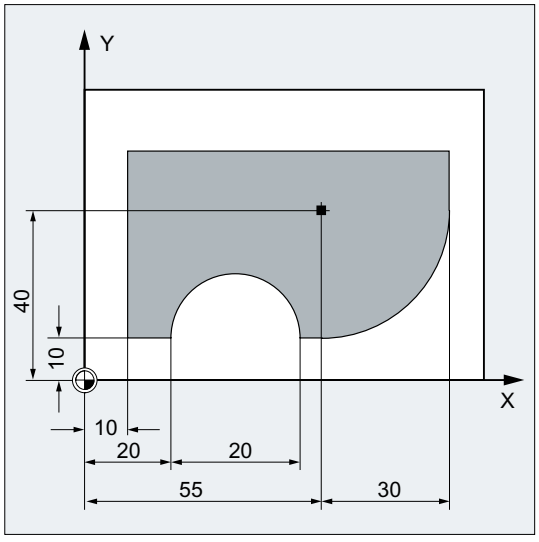
```
CFTCP
CFC
CFIN
```

## Bedeutung

| | |
|---|---|
| CFTCP: | Konstanter Vorschub an der Fräsermittelpunktsbahn<br>Die Steuerung hält die Vorschubgeschwindigkeit konstant, Vorschub-Overrides werden ausgeschaltet. |
| CFC: | Konstanter Vorschub an der Kontur (Werkzeugschneide)<br>Diese Funktion ist standardmäßig voreingestellt. |
| CFIN: | Konstanter Vorschub an der Werkzeugschneide nur an innengekrümmten Konturen, sonst auf der Fräsermittelpunktsbahn<br>Die Vorschubgeschwindigkeit wird bei Innenradien reduziert. |

## Beispiel

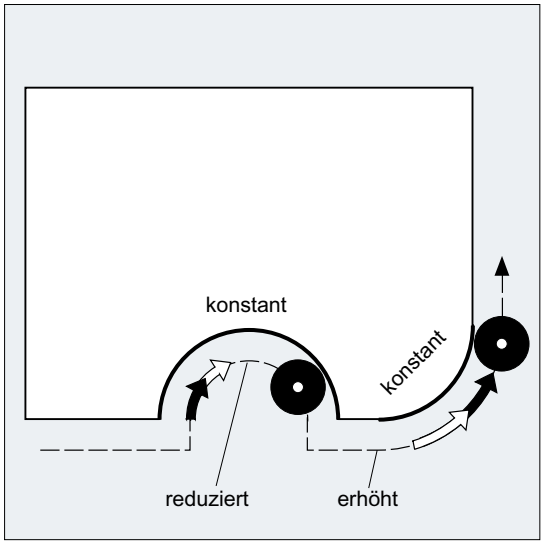
In diesem Beispiel wird zunächst mit CFC-korrigiertem Vorschub die Kontur hergestellt. Beim Schlichten wird der Fräsgrund mit CFIN zusätzlich bearbeitet. Hierdurch lässt sich verhindern, dass der Fräsgrund an Außenradien durch zu hohe Vorschubgeschwindigkeit beschädigt wird.




| **Programmcode** | **Kommentar** |
|---|---|
| N10 G17 G54 G64 T1 M6 | |
| N20 S3000 M3 CFC F500 G41 | |
| N30 G0 X-10 | |
| N40 Y0 Z-10 | ; Zustellen auf erste Schnitt-Tiefe |
| N50 KONTUR1 | ; Unterprogramm-Aufruf |
| N40 CFIN Z-25 | ; Zustellen auf zweite Schnitt-Tiefe |
| N50 KONTUR1 | ; Unterprogramm-Aufruf |
| N60 Y120 | |
| N70 X200 M30 | |

### Weitere Informationen

**Konstanter Vorschub an der Kontur mit CFC**




Die Vorschubgeschwindigkeit wird bei Innenradien reduziert, bei Außenradien erhöht. Hierdurch bleibt die Geschwindigkeit an der Werkzeugschneide und damit an der Kontur konstant.

## 2.7.10 Mehrere Vorschubwerte in einem Satz (F, ST, SR, FMA, STA, SRA)

Mit der Funktion "Mehrere Vorschubwerte in einem Satz" können abhängig von externen digitalen und/oder analogen Eingängen verschiedene Vorschubwerte eines NC-Satzes, Verweilzeit sowie Rückzug bewegungssynchron aktiviert werden

### Syntax

Bahnbewegung:
```
F=... F7=... F6=... F5=... F4=... F3=... F2=... ST=... SR=...
```

Axiale Bewegung:
```
FA[<Ax>]=... FMA[7,<Ax>]=... FMA[6,<Ax>]=... FMA[5,<Ax>]=...
FMA[4,<Ax>]=... FMA[3,<Ax>]=... FMA[2,<Ax>]=... STA[<Ax>]=...
SRA[<Ax>]=...
```

### Bedeutung

| | | |
|---|---|---|
| `F=...`: | Unter der Adresse `F` wird der Bahnvorschub programmiert, der gültig ist, solange kein Eingangssignal ansteht. | |
| | Wirksamkeit: | modal |

| | | |
|---|---|---|
| F2=... bis F7=... : | Zusätzlich zum Bahnvorschub können bis zu 6 weitere Vorschübe im Satz programmiert werden. Die numerische Erweiterung gibt die Bitnummer des Eingangs an, mit dessen Änderung der Vorschub wirksam wird. | |
| | Wirksamkeit: | satzweise |
| ST=... : | Verweilzeit in s (bei der Technologie Schleifen: Ausfeuerzeit) | |
| | Eingangsbit: | 1 |
| | Wirksamkeit: | satzweise |
| SR=... : | Rückzugsweg<br>Die Einheit für den Rückzugsweg bezieht sich auf die aktuell gültige Maßeinheit (mm oder inch). | |
| | Eingangsbit: | 0 |
| | Wirksamkeit: | satzweise |
| FA[<Ax>]=... : | Unter der Adresse FA wird der axiale Vorschub programmiert, der gültig ist, solange kein Eingangssignal ansteht. | |
| | Wirksamkeit: | modal |
| FMA[2,<Ax>]=... bis FMA[7,<Ax>]=... : | Zusätzlich zum axialen Vorschub FA können mit FMA bis zu 6 weitere Vorschübe pro Achse im Satz programmiert werden. Der erste Parameter gibt die Bitnummer des Eingangs an, der zweite die Achse, für die der Vorschub gelten soll. | |
| | Wirksamkeit: | satzweise |
| STA[<Ax>]=...: | axiale Verweilzeit in s (bei der Technologie Schleifen: Ausfeuerzeit) | |
| | Eingangsbit: | 1 |
| | Wirksamkeit: | satzweise |
| SRA[<Ax>]=...: | axialer Rückzugsweg | |
| | Eingangsbit: | 0 |
| | Wirksamkeit: | satzweise |
| <Ax>: | Achse, für die der Vorschub gelten soll | |

**Hinweis**

**Priorität der Signale**

Die Abfragereihenfolge der Signale erfolgt von Eingangsbit 0 (E0) aufsteigend. Somit besitzen die Rückzugsbewegung die höchste und der Vorschub F7 die niedrigste Priorität. Verweilzeit und Rückzugsbewegung beenden die Vorschubbewegungen, die mit F2 bis F7 aktiviert wurden.

Das Signal mit der höchsten Priorität bestimmt den aktuellen Vorschub.

**Hinweis**

**Restweglöschen**

Wenn der Eingang Bit 1 für Verweilzeit bzw. Rückzugsweg Bit 0 aktiviert wird, wird der Restweg für Bahnachsen oder die betreffenden Einzelachsen gelöscht und die Verweilzeit bzw. der Rückzug gestartet.

**Hinweis**

**Rückzugsweg**

Die Einheit für den Rückzugsweg bezieht sich auf die aktuell gültige Maßeinheit (mm oder inch).

Der Rückhub erfolgt immer in Gegenrichtung zur aktuellen Bewegung. Mit `SR`/`SRA` wird immer der Betrag des Rückhubs programmiert. Es wird kein Vorzeichen programmiert.

**Hinweis**

**POS statt POSA**

Wenn für eine Achse Vorschübe, Verweilzeit oder Rückzugsweg aufgrund eines externen Eingangs programmiert sind, darf diese Achse in diesem Satz nicht als POSA-Achse (Positionierachse über Satzgrenzen hinweg) programmiert werden.

**Hinweis**

**Statusabfrage**

Der Status eines Eingangs kann auch für Synchronbefehle unterschiedlicher Achsen abgefragt werden.

**Hinweis**

**LookAhead**

LookAhead ist auch bei mehreren Vorschüben in einem Satz wirksam. Damit kann der aktuelle Vorschub durch LookAhead begrenzt werden.

## Beispiele

### Beispiel 1: Bahnbewegung

| **Programmcode** | **Kommentar** |
|---|---|
| `G1 X48 F1000 F7=200 F6=50 F5=25 F4=5 ST=1.5 SR=0.5` | `; Bahnvorschub = 1000`<br>`; zusätzliche Bahnvorschubwerte:`<br>`; 200 (Eingangsbit 7)`<br>`; 50 (Eingangsbit 6)`<br>`; 25 (Eingangsbit 5)`<br>`; 5 (Eingangsbit 4)`<br>`; Verweilzeit 1,5s`<br>`; Rückzug 0,5mm` |

**Beispiel 2: Axiale Bewegung**

| Programmcode | Kommentar |
|---|---|
| POS[A]=300 FA[A]=800 FMA[7,A]=720 FMA[6,A]=640<br>FMA[5,A]=560 STA[A]=1.5 SRA[A]=0.5 | ; Vorschub für Achse A = 800<br>; zusätzliche Vorschubwerte für Achse A:<br>720(Eingangsbit 7)<br>; 640 (Eingangsbit 6)<br>; 560 (Eingangsbit 5)<br>; axiale Verweilzeit: 1,5s<br>; axialer Rückzug: 0,5mm |

**Beispiel 3: Mehrere Arbeitsgänge in einem Satz**

| Programmcode | Kommentar |
|---|---|
| N20 T1 D1 F500 G0 X100 | Ausgangsstellung |
| N25 G1 X105 F=20 F7=5 F3=2.5 F2=0.5 ST=1.5 SR=0.5 | ; Normalvorschub mit F,<br>; Schruppen mit F7,<br>; Schlichten mit F3,<br>; Feinschlichten mit F2,<br>; Verweilzeit 1,5 s,<br>; Rückzugsweg 0,5 mm |
| ... | |

## 2.7.11 Satzweiser Vorschub (FB)

Mit der Funktion "Satzweiser Vorschub" kann für einen einzelnen Satz ein separater Vorschub vorgegeben werden. Nach diesem Satz ist der zuvor wirksame modale Vorschub wieder aktiv.

### Syntax

```
FB=<Wert>
```

### Bedeutung

| | |
|---|---|
| FB: | Vorschub nur für den aktuellen Satz |
| <WERT>: | Der programmierte Wert muss größer Null sein.<br>Die Interpretation erfolgt entsprechend des aktiven Vorschubtyps:<br>• G94: Vorschub in mm/min oder Grad/min<br>• G95: Vorschub in mm/Umdr. oder inch/Umdr.<br>• G96: konstante Schnittgeschwindigkeit |

**Hinweis**

Ist im Satz keine Verfahrbewegung programmiert (z. B. Rechensatz), bleibt `FB` ohne Wirkung.

Ist kein expliziter Vorschub für Fase/Rundung programmiert, gilt der Wert von `FB` auch für ein in diesem Satz vorhandenes Konturelement Fase/Rundung.

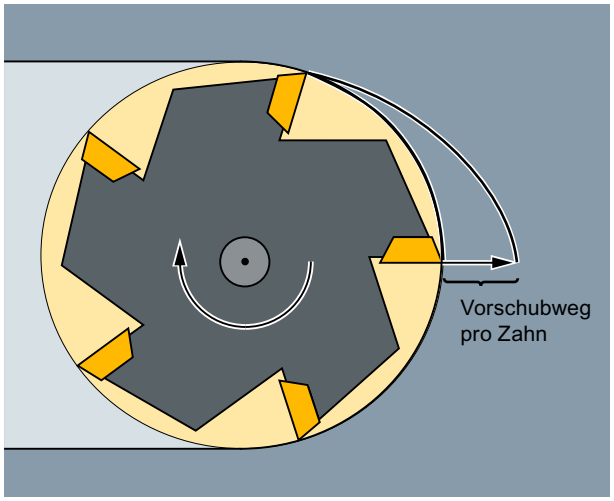
Vorschubinterpolationen `FLIN`, `FCUB`, ... sind uneingeschränkt möglich.

Die gleichzeitige Programmierung von `FB` und `FD` (manuelles Verfahren perr Handrad mit Vorschubüberlagerung) oder `F` (modaler Bahnvorschub) ist **nicht** möglich.

**Beispiel**

| Programmcode | Kommentar |
|---|---|
| `N10 G0 X0 Y0 G17 F100 G94` | `; Ausgangsstellung` |
| `N20 G1 X10` | `; Vorschub 100 mm/min` |
| `N30 X20 FB=80` | `; Vorschub 80 mm/min` |
| `N40 X30` | `; Vorschub ist wieder 100 mm/min.` |
| `...` | |

## 2.7.12 Zahnvorschub (G95 FZ)




Über den Werkzeugparameter $TC_DPNT (Anzahl der Zähne) des aktiven Werkzeugkorrekturdatensatzes berechnet die Steuerung aus dem programmierten Zahnvorschub für jeden Verfahrsatz den wirksamen Umdrehungsvorschub:

F = FZ * $TC_DPNT

| | | |
|---|---|---|
| mit: | F: | Umdrehungsvorschub in mm/U bzw. inch/U |
| | FZ: | Zahnvorschub in mm/Zahn bzw. inch/Zahn |
| | $TC_DPNT: | Systemvariable Werkzeugparameter: Anzahl der Zähne/U |

Der Werkzeugtyp ($TC_DP1) des aktiven Werkzeugs wird nicht berücksichtigt.

Der programmierte Zahnvorschub ist unabhängig von Werkzeugwechsel und An-/Abwahl eines Werkzeugkorrekturdatensatzes und bleibt modal erhalten.

Eine Änderung des Werkzeugparameters $TC_DPNT der aktiven Schneide wird mit der nächsten Werkzeugkorrekturanwahl bzw. der nächsten Aktualisierung der aktiven Korrekturdaten wirksam.

Werkzeugwechsel und An-/Abwahl eines Werkzeugkorrekturdatensatzes führen zu einer Neuberechnung des wirksamen Umdrehungsvorschubs.

**Hinweis**

Der Zahnvorschub bezieht sich nur auf die Bahn, eine achsspezifische Programmierung ist nicht möglich.

## Syntax

```
G95 FZ...
```

## Bedeutung

| | | |
|---|---|---|
| G95: | Vorschubart: Umdrehungsvorschub in mm/U bzw. inch/U (abhängig von G700/G710)<br>Zu G95 siehe "Vorschub (G93, G94, G95, F, FGROUP, FL, FGREF) (Seite 108)" | |
| FZ: | Zahnvorschubgeschwindigkeit | |
| | Aktivierung: | mit G95 |
| | Wirksamkeit: | modal |
| | Maßeinheit: | mm/Zahn bzw. inch/Zahn (abhängig von G700/G710) |

**ACHTUNG**

**Werkzeugwechsel / Wechsel der Masterspindel**

Ein nachfolgender Werkzeugwechsel oder Wechsel der Masterspindel muss vom Anwender durch entsprechende Programmierung, z. B. erneute Programmierung von FZ, berücksichtigt werden.

**ACHTUNG**

**Werkzeugeingriffe undefiniert**

Technologische Belange wie z. B. Gleich- oder Gegenlauffräsen, Stirn- oder Umfangsplanfräsen usw. werden ebenso wie die Bahngeometrie (Gerade, Kreis, ...) nicht automatisch berücksichtigt. Diese Faktoren müssen daher bei der Programmierung des Zahnvorschubs beachtet werden.

**Hinweis**

**Umschaltung zwischen G95 F... und G95 FZ...**

Mit Umschaltung zwischen `G95 F...` (Umdrehungsvorschub) und `G95 FZ...` (Zahnvorschub) wird der jeweils nicht aktive Vorschubwert gelöscht.

**Hinweis**

**Vorschub ableiten mit FPR**

Mit `FPR` kann analog zum Umdrehungsvorschub auch der Zahnvorschub von einer beliebigen Rundachse oder Spindel abgeleitet werden (siehe "Vorschub für Positionierachsen/Spindeln (FA, FPR, FPRAON, FPRAOF) (Seite 126)").

## Beispiele

### Beispiel 1: Fräser mit 5 Zähnen ($TC_DPNT = 5)

| Programmcode | Kommentar |
|---|---|
| N10 G0 X100 Y50 | |
| N20 G1 G95 FZ=0.02 | ; Zahnvorschub 0,02 mm/Zahn |
| N30 T3 D1 | ; Werkzeug einwechseln und Werkzeugkorrekturdatensatz aktivieren. |
| M40 M3 S200 | ; Spindeldrehzahl 200 U/min |
| N50 X20 | ; Fräsen mit: |
| | FZ = 0,02 mm/Zahn |
| | wirksamer Umdrehungsvorschub: |
| | F = 0,02 mm/Zahn * 5 Zähne/U = 0,1 mm/U |
| | bzw. |
| | F = 0,1 mm/U * 200 U/min = 20 mm/min |
| ... | |

### Beispiel 2: Umschaltung zwischen G95 F... und G95 FZ...

| Programmcode | Kommentar |
|---|---|
| N10 G0 X100 Y50 | |
| N20 G1 G95 F0.1 | ; Umdrehungsvorschub 0,1 mm/U |
| N30 T1 M6 | |
| N35 M3 S100 D1 | |
| N40 X20 | |
| N50 G0 X100 M5 | |
| N60 M6 T3 D1 | ; Werkzeug mit z.B. 5 Zähnen ($TC_DPNT = 5) einwechseln. |
| N70 X22 M3 S300 | |
| N80 G1 X3 G95 FZ=0.02 | ; Wechsel G95 F… auf G95 FZ…, Zahnvorschub mit 0,02 mm/Zahn aktiv. |
| ... | |

### Beispiel 3: Zahnvorschub von einer Spindel ableiten (FBR)

```
Programmcode                Kommentar
…
N41 FPR(S4)                 ; Werkzeug auf Spindel 4 (nicht Masterspindel).
N51 G95 X51 FZ=0.5          ; Zahnvorschub 0,5 mm/Zahn abhängig von Spindel S4.
...
```

### Beispiel 4: Nachfolgender Werkzeugwechsel

```
Programmcode                Kommentar
N10 G0 X50 Y5
N20 G1 G95 FZ=0.03          ; Zahnvorschub 0,03 mm/Zahn
N30 M6 T11 D1               ; Werkzeug mit z.B. 7 Zähnen ($TC_DPNT = 7) einwechseln.
N30 M3 S100
N40 X30                     ; wirksamer Umdrehungsvorschub 0,21 mm/U
N50 G0 X100 M5
N60 M6 T33 D1               ; Werkzeug mit z.B. 5 Zähnen ($TC_DPNT = 5) einwechseln.
N70 X22 M3 S300
N80 G1 X3                   ; Zahnvorschub modal 0,03 mm/Zahn,
                            wirksamer Umdrehungsvorschub 0,15 mm/U
...
```

### Beispiel 5: Wechsel der Masterspindel

```
Programmcode                  Kommentar
N10 SETMS(1)                  ; Spindel 1 ist Masterspindel.
N20 T3 D3 M6                  ; Werkzeug 3 wird auf Spindel 1 gewechselt.
N30 S400 M3                   ; Drehzahl S400 der Spindel 1 (und damit T3).
N40 G95 G1 FZ0.03             ; Zahnvorschub 0,03 mm/Zahn
N50 X50                       ; Bahnbewegung, der wirksame Vorschub ist abhängig von:
                              - Zahnvorschub FZ
                              - Drehzahl der Spindel 1
                              - Anzahl der Zähne des aktiven Werkzeugs T3
N60 G0 X60
...
N100 SETMS(2)                 ; Spindel 2 wird Masterspindel.
N110 T1 D1 M6                 ; Werkzeug 1 wird auf Spindel 2 gewechselt.
N120 S500 M3                  ; Drehzahl S500 der Spindel 2 (und damit T1).
N130 G95 G1 FZ0.03 X20        ; Bahnbewegung, der wirksame Vorschub ist abhängig von:
                              - Zahnvorschub FZ
                              - Drehzahl der Spindel 2
                              - Anzahl der Zähne des aktiven Werkzeugs T1
```

**Hinweis**

Nach dem Wechsel der Masterspindel (N100) muss ein Werkzeug eingewechselt werden (N110), das von Spindel 2 angetrieben wird.

## Weitere Informationen

### Wechsel zwischen G93, G94 und G95

FZ kann auch bei nicht aktivem G95 programmiert werden, hat jedoch keine Wirkung und wird mit der G95-Anwahl gelöscht, d. h. mit Wechsel zwischen G93, G94 und G95 wird analog zu F auch der der FZ-Wert gelöscht.

### Erneute Anwahl von G95

Eine erneute Anwahl von G95 bei schon aktivem G95 hat keine Wirkung (falls hierbei kein Wechsel zwischen F und FZ programmiert ist).

### Satzweise wirksamer Vorschub (FB)

Ein satzweise wirksamer Vorschub FB... wird bei aktivem G95 FZ... (modal) als Zahnvorschub interpretiert.

### SAVE-Mechanismus

Bei Unterprogrammen mit dem SAVE-Attribut wird FZ analog zu F auf den Wert vor Unterprogrammstart geschrieben.

### Mehrere Vorschubwerte in einem Satz

Die Funktion "Mehrere Vorschubwerte in einem Satz" ist bei Zahnvorschub nicht möglich.

### Synchronaktionen

Die Vorgabe von FZ aus Synchronaktionen ist nicht möglich.

### Zahnvorschubgeschwindigkeit und Bahnvorschub-Typ lesen

Die Zahnvorschubgeschwindigkeit und der Bahnvorschub-Typ können über Systemvariablen gelesen werden:

- Mit Vorlauf-Stop im Teileprogramm über die Systemvariablen:

| | | | |
|---|---|---|---|
| | $AC_FZ | Zahnvorschubgeschwindigkeit, die bei der Aufbereitung des aktuellen Hauptlaufsatzes wirksam war. | |
| | $AC_F_TYPE | Bahnvorschub-Typ, der bei der Aufbereitung des aktuellen Hauptlaufsatzes wirksam war. | |
| | | **Wert:** | **Bedeutung:** |
| | | 0 | mm/min |
| | | 1 | mm/U |
| | | 2 | inch/min |
| | | 3 | inch/U |
| | | 11 | mm/Zahn |
| | | 33 | inch/Zahn |

- Ohne Vorlauf-Stop im Teileprogramm über die Systemvariablen:

| | | | |
|---|---|---|---|
| | $P_FZ | Programmierte Zahnvorschubgeschwindigkeit | |
| | $P_F_TYPE | Programmierter Bahnvorschub-Typ | |
| | | **Wert:** | **Bedeutung:** |
| | | 0 | mm/min |
| | | 1 | mm/U |
| | | 2 | inch/min |
| | | 3 | inch/U |
| | | 11 | mm/Zahn |
| | | 33 | inch/Zahn |

**Hinweis**

Ist `G95` nicht aktiv, liefern die Variablen $P_FZ und $AC_FZ immer den Wert Null.

# 2.8 Geometrie-Einstellungen

## 2.8.1 Einstellbare Nullpunktverschiebung (G54 ... G57, G505 ... G599, G53, G500, SUPA, G153)

Über die Befehle G54 bis G57 und G505 bis G599 werden die über die Bedienoberfläche einstellbaren Nullpunktverschiebungen zur Verschiebung des Werkstückkoordinatensystems gegenüber dem Basiskoordinatensystem aktiviert.

### Syntax

**Einschalten:**

```
G54
...
G57
G505
...
G599
```

**Ausschalten bzw. Unterdrücken:**

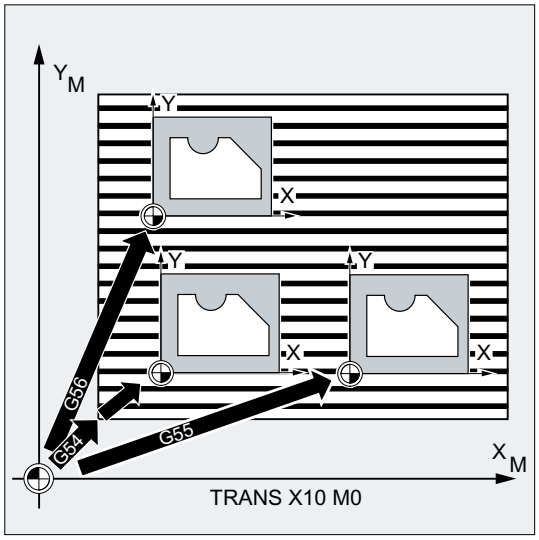
`G500` / `G53` / `G153` / `SUPA`

### Bedeutung

| | | |
|---|---|---|
| `G54` ... `G57`: | Aufruf der 1. bis 4. einstellbaren Nullpunktverschiebung (NV) | |
| `G505` ... `G599`: | Aufruf der 5. bis 99. einstellbaren NV | |
| `G500`: | Ausschalten der aktuellen einstellbaren NV | |
| | G500=Nullframe:<br>(Standardeinstellung; enthält keine Verschiebung, Drehung, Spiegelung oder Skalierung) | Ausschalten der einstellbaren NV bis zum nächsten Aufruf, Aktivierung des Gesamt-Basisframes ($P_ACTBFRAME). |
| | G500 ungleich 0: | Aktivierung der ersten einstellbaren Nullpunktverschiebung ($P_UIFR[0]) und Aktivierung des Gesamt-Basisframes ($P_ACTBFRAME) bzw. ein evtl. geänderter Basisframe wird aktiviert. |
| `G53`: | G53 unterdrückt satzweise einstellbare NV und programmierbare NV. | |
| `G153`: | G153 wirkt wie G53 und unterdrückt darüber hinaus den Gesamt-Basisframe. | |
| `SUPA`: | SUPA wirkt wie G153 und unterdrückt darüber hinaus:<br>• Handradverschiebungen (DRF)<br>• überlagerte Bewegungen<br>• externe NV<br>• PRESET-Verschiebung | |

### Beispiel

Drei Werkstücke, die auf einer Palette entsprechend der Nullpunktverschiebewerte `G54` bis `G56` angeordnet sind, sollen nacheinander bearbeitet werden. Die Bearbeitungsfolge ist im Unterprogramm L47 programmiert.




| **Programmcode** | **Kommentar** |
|---|---|
| `N10 G0 G90 X10 Y10 F500 T1` | `; Anfahren` |
| `N20 G54 S1000 M3` | `; Aufruf der ersten NV, Spindel rechts` |
| `N30 L47` | `; Programmlauf als Unterprogramm` |
| `N40 G55 G0 Z200` | `; Aufruf der zweiten NV, Z über Hindernis` |
| `N50 L47` | `; Programmlauf als Unterprogramm` |
| `N60 G56` | `; Aufruf der dritten NV` |
| `N70 L47` | `; Programmlauf als Unterprogramm` |
| `N80 G53 X200 Y300 M30` | `; Nullpunktverschiebung unterdrücken, Programmende` |

## 2.8.2 Einstellbare Nullpunktverschiebung (G54 ... G57, G505 ... G599, G53, G500, SUPA, G153): Weitere Informationen

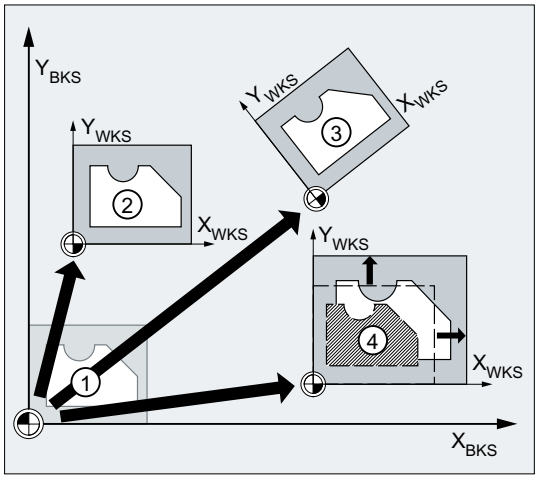
### Weitere Informationen

#### Einstellbare Nullpunktverschiebung

Eine einstellbare Nullpunktverschiebung ist im Prinzip ein einstellbarer Frame (Seite 308). Daher stehen auch bei einer einstellbaren Nullpunktverschiebung folgende Komponenten bzw. Frame-Werte zur Verfügung:

- Verschiebung
- Rotation

- Skalierung
- Maßstab




① Ausgangslage im BKS
② Verschiebung
③ Verschiebung + Drehung
④ Verschiebung + Skalierung

Die Eingabe der Frame-Werte für die einstellbaren Nullpunktverschiebungen erfolgt über die Bedienoberfläche:

SINUMERIK Operate: Bedienbereich "Parameter" > "Nullpunktverschiebungen" > "Details"

**Parametrierbare Anzahl einstellbarer Frames (G505 - G599)**

Die Anzahl der anwenderspezifischen einstellbaren Nullpunktverschiebungen (G505 - G599) ist kanalspezifisch einstellbar über:

MD28080 $MC_MM_NUM_USER_FRAMES = <Anzahl>

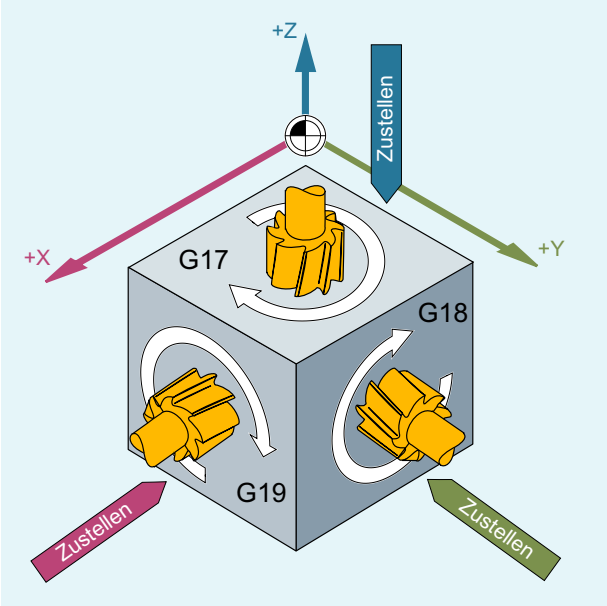
### Siehe auch

Programmierbare Nullpunktverschiebung (G58, G59) (Seite 317)

## 2.8.3 Wahl der Arbeitsebene (G17/G18/G19)

Durch die Angabe der Arbeitsebene, in der die gewünschte Kontur gefertigt werden soll, werden zugleich folgende Funktionen festgelegt:

- Die Ebene für die Werkzeugradiuskorrektur.
- Die Zustellrichtung für die Werkzeuglängenkorrektur in Abhängigkeit vom Werkzeugtyp.
- Die Ebene für die Kreisinterpolation.




## Syntax

```
G17/G18/G19 ...
```

## Bedeutung

| | |
|---|---|
| `G17`: | Arbeitsebene X/Y<br>Zustellrichtung Z Ebenenanwahl 1. - 2. Geometrieachse |
| `G18`: | Arbeitsebene Z/X<br>Zustellrichtung Y Ebenenanwahl 3. - 1. Geometrieachse |
| `G19`: | Arbeitsebene Y/Z<br>Zustellrichtung X Ebenenanwahl 2. - 3. Geometrieachse |

### Hinweis

In der Grundeinstellung ist für Fräsen G17 (X/Y-Ebene) und für Drehen G18 (Z/X-Ebene) voreingestellt.

Mit Aufruf der Werkzeug-Bahnkorrektur G41/G42 (siehe Kapitel "Werkzeugradiuskorrekturen (Seite 254)") muss die Arbeitsebene angegeben werden, damit die Steuerung Werkzeuglänge und -radius korrigieren kann.

## Beispiel

Die "klassische" Vorgehensweise beim Fräsen ist:

1. Arbeitsebene (`G17` Grundeinstellung für Fräsen) definieren.
2. Werkzeugtyp (`T`) und Werkzeugkorrekturwerte (`D`) aufrufen.

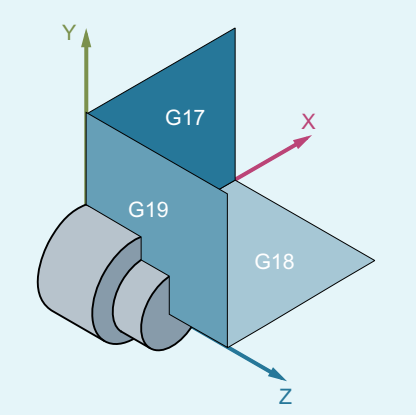
3. Bahnkorrektur (`G41`) einschalten.
4. Fahrbewegungen programmieren.

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 G17 T5 D8` | `; Aufruf der Arbeitsebene X/Y, Werkzeugaufruf. Die Längenkorrektur erfolgt in Z-Richtung.` |
| `N20 G1 G41 X10 Y30 Z-5 F500` | `; Die Radiuskorrektur erfolgt in der X/Y-Ebene.` |
| `N30 G2 X22.5 Y40 I50 J40` | `; Kreisinterpolation/Werkzeugradius-korrektur in der X/Y-Ebene.` |

## Weitere Informationen

### Allgemein

Es empfiehlt sich, die Arbeitsebene G17 bis G19 bereits am Programmanfang festzulegen. In der Grundeinstellung ist für Drehen G18 die Z/X-Ebene voreingestellt:

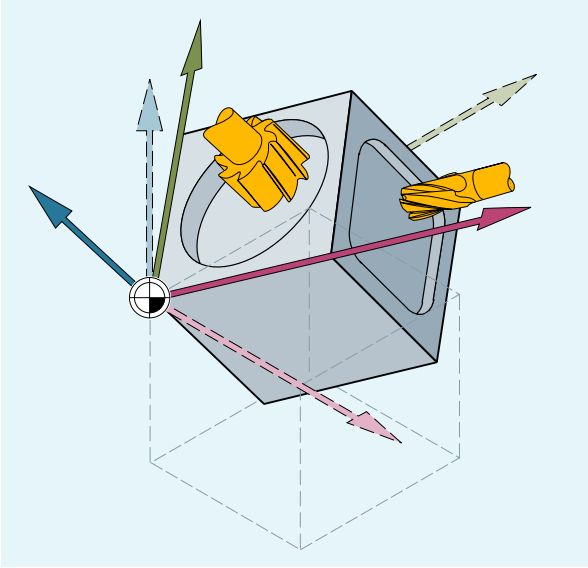



Die Steuerung benötigt zur Berechnung des Drehsinns die Angabe der Arbeitsebene. Weitere Informationen hierzu siehe Kapitel "Kreisinterpolation (Seite 191)".

### Bearbeitung in schräg liegenden Ebenen

Durch Drehung des Koordinatensystems mit ROT (Seite 319) werden die Koordinatenachsen auf die schräg liegende Fläche gelegt. Die Arbeitsebenen drehen sich entsprechend mit.

**Werkzeuglängenkorrektur in schräg liegenden Ebenen**



Die Werkzeuglängenkorrektur wird generell immer bezogen auf die raumfeste, nicht gedrehte Arbeitsebene errechnet.

**Hinweis**

Mit den Funktionalitäten zur "Werkzeuglängenkorrektur für orientierbare Werkzeuge" können die Werkzeuglängenkomponenten passend zu den gedrehten Arbeitsebenen errechnet werden.

Die Wahl der Korrekturebene erfolgt mit CUT2D, CUT2DF. Weitere Informationen hierzu und zur Beschreibung dieser Berechnungsmöglichkeit siehe Kapitel "Werkzeugradiuskorrekturen (Seite 254)".

Für die räumliche Festlegung der Arbeitsebene bietet die Steuerung sehr komfortable Möglichkeiten für Koordinatentransformationen. Weitere Informationen hierzu siehe Kapitel "Koordinatentransformationen (Frames) (Seite 308)".

## 2.8.4 Maßangaben

Grundlage der meisten NC-Programme ist eine Werkstückzeichnung mit konkreten Maßangaben.

Diese Maßangaben können sein:

- im Absolutmaß oder Kettenmaß
- in Millimetern oder Inch
- im Radius oder Durchmesser (beim Drehen)

Damit die Angaben aus einer Maßzeichnung direkt (ohne Umrechnung) in das NC-Programm übernommen werden können, stehen dem Anwender für die verschiedenen Möglichkeiten zur Maßangabe spezifische Programmierbefehle zur Verfügung.

### 2.8.4.1 Absolutmaßangabe (G90, AC)

Bei der Absolutmaßangabe beziehen sich die Positionsangaben immer auf den Nullpunkt des aktuell gültigen Koordinatensystems, d. h. es wird die absolute Position programmiert, auf die das Werkzeug fahren soll.

#### Modal wirksame Absolutmaßangabe

Die modal wirksame Absolutmaßangabe wird aktiviert mit dem Befehl `G90`. Sie ist für alle Achsen wirksam, die in den jeweils folgenden NC-Sätzen programmiert werden.

#### Satzweise wirksame Absolutmaßeingabe

Bei voreingestelltem Kettenmaß (`G91`) kann mit Hilfe des Befehls `AC` für einzelne Achsen satzweise Absolutmaßangabe eingestellt werden.

**Hinweis**

Die satzweise wirksame Absolutmaßangabe(`AC`) ist auch für Spindelpositionierungen (`SPOS`, `SPOSA`) und Interpolationsparameter (`I`, `J`, `K`) möglich.

## Syntax

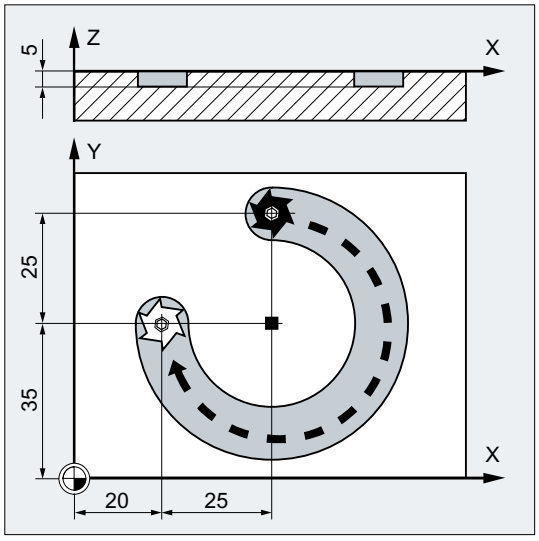
```
G90
<Achse>=AC(<Wert>)
```

## Bedeutung

| | |
|---|---|
| `G90`: | Befehl zur Aktivierung der modal wirksamen Absolutmaßangabe |
| `AC`: | Befehl zur Aktivierung der satzweise wirksamen Absolutmaßangabe |
| `<Achse>`: | Achsbezeichner der zu verfahrenden Achse |
| `<Wert>`: | Sollposition der zu verfahrenden Achse im Absolutmaß |

## Beispiele

### Beispiel 1: Fräsen

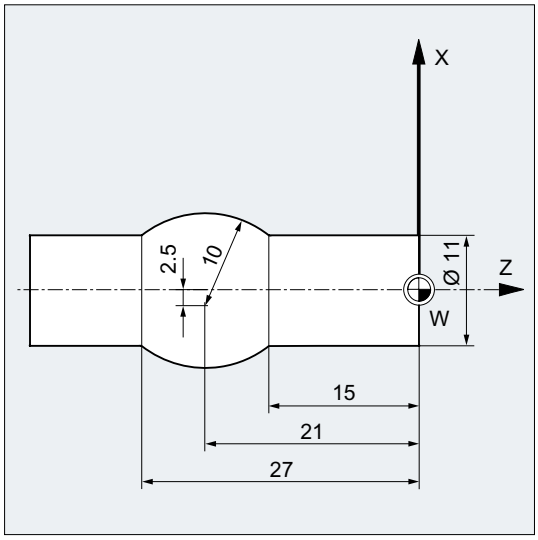



| Programmcode | Kommentar |
|---|---|
| N10 G90 G0 X45 Y60 Z2 T1 S2000 M3 | ; Absolutmaßeingabe, im Eilgang auf Position XYZ, Werkzeug-Anwahl, Spindel ein mit Drehrichtung rechts. |
| N20 G1 Z-5 F500 | ; Geradeninterpolation, Zustellen des Werkzeugs. |
| N30 G2 X20 Y35 I=AC(45) J=AC(35) | ; Kreisinterpolation im Uhrzeigersinn, Kreisendpunkt und Kreismittelpunkt im Absolutmaß. |
| N40 G0 Z2 | ; Herausfahren. |
| N50 M30 | ; Satzende. |

**Hinweis**

Zur Eingabe der Kreismittelpunktskoordinaten I und J siehe Kapitel "Kreisinterpolation".

### Beispiel 2: Drehen




| Programmcode | Kommentar |
|---|---|
| N5 T1 D1 S2000 M3 | ; Einwechseln von Werkzeug T1, Spindel ein mit Drehrichtung rechts. |
| N10 G0 G90 X11 Z1 | ; Absolutmaßeingabe, im Eilgang auf Position XZ. |
| N20 G1 Z-15 F0.2 | ; Geradeninterpolation, Zustellen des Werkzeugs. |
| N30 G3 X11 Z-27 I=AC(-5) K=AC(-21) | ; Kreisinterpolation gegen Uhrzeigersinn, Kreisendpunkt und Kreismittelpunkt im Absolutmaß. |
| N40 G1 Z-40 | ; Herausfahren. |
| N50 M30 | ; Satzende. |

**Hinweis**

Zur Eingabe der Kreismittelpunktskoordinaten I und J siehe Kapitel "Kreisinterpolation".

## Siehe auch

Absolut- und Kettenmaßangabe beim Drehen und Fräsen (G90/G91) (Seite 159)

### 2.8.4.2 Kettenmaßangabe (G91, IC)

Bei der Kettenmaßangabe bezieht sich eine Positionsangabe auf den zuletzt angefahrenen Punkt, d. h. die Programmierung im Kettenmaß beschreibt, um wie viel das Werkzeug verfahren soll.

#### Modal wirksame Kettenmaßangabe

Die modal wirksame Kettenmaßangabe wird aktiviert mit dem Befehl `G91`. Sie ist für alle Achsen wirksam, die in den jeweils folgenden NC-Sätzen programmiert werden.

#### Satzweise wirksame Kettenmaßangabe

Bei voreingestelltem Absolutmaß (G90) kann mit Hilfe des Befehls IC für einzelne Achsen satzweise Kettenmaßangabe eingestellt werden.

**Hinweis**

Die satzweise wirksame Kettenmaßangabe(IC) ist auch für Spindelpositionierungen (SPOS, SPOSA) und Interpolationsparameter (I, J, K) möglich.

## Syntax

```
G91
<Achse>=IC(<Wert>)
```

## Bedeutung

| | |
|---|---|
| G91: | Befehl zur Aktivierung der modal wirksamen Kettenmaßangabe |
| IC: | Befehl zur Aktivierung der satzweise wirksamen Kettenmaßangabe |
| <Achse>: | Achsbezeichner der zu verfahrenden Achse |
| <Wert>: | Sollposition der zu verfahrenden Achse im Kettenmaß |

## G91-Erweiterung

Für bestimmte Anwendungen wie z. B. dem Ankratzen ist es erforderlich, im Kettenmaß nur den programmierten Weg zu fahren. Die aktive Nullpunktverschiebung oder Werkzeuglängenkorrektur wird nicht gefahren.

Dieses Verhalten kann getrennt für die aktive Nullpunktverschiebung und Werkzeuglängenkorrektur über folgende Settingdaten eingestellt werden:

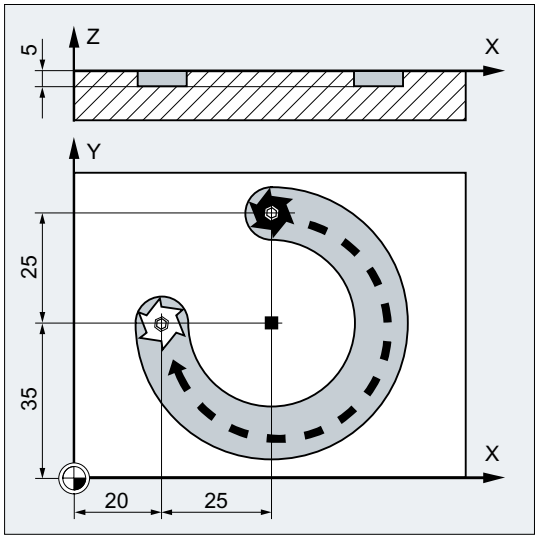
SD42440 $SC_FRAME_OFFSET_INCR_PROG (Nullpunktsverschiebungen in Frames)

SD42442 $SC_TOOL_OFFSET_INCR_PROG (Werkzeuglängenkorrekturen)

| Wert | Bedeutung |
|---|---|
| 0 | Bei inkrementeller Programmierung (Kettenmaßangabe) einer Achse wird die aktive Nullpunktverschiebung bzw. die Werkzeuglängenkorrektur **nicht** herausgefahren. |
| 1 | Bei inkrementeller Programmierung (Kettenmaßangabe) einer Achse wird die aktive Nullpunktverschiebung bzw. die Werkzeuglängenkorrektur herausgefahren. |

## Beispiele

### Beispiel 1: Fräsen

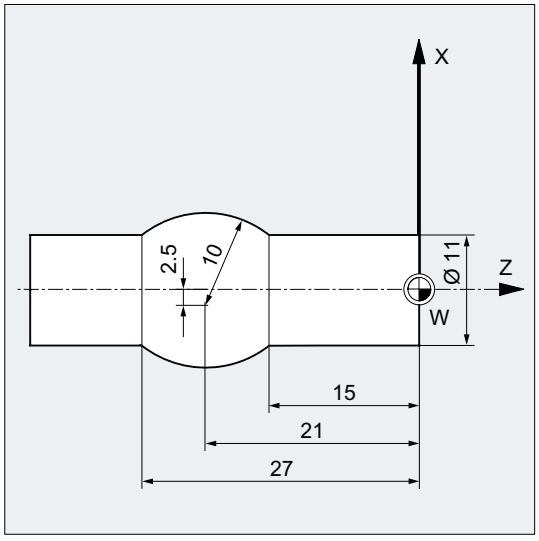



| Programmcode | Kommentar |
|---|---|
| N10 G90 G0 X45 Y60 Z2 T1 S2000 M3 | ; Absolutmaßeingabe, im Eilgang auf Position XYZ, Werkzeug-Anwahl, Spindel ein mit Drehrichtung rechts. |
| N20 G1 Z-5 F500 | ; Geradeninterpolation, Zustellen des Werkzeugs. |
| N30 G2 X20 Y35 I0 J-25 | ; Kreisinterpolation im Uhrzeigersinn, Kreisendpunkt im Absolutmaß, Kreismittelpunkt im Kettenmaß. |
| N40 G0 Z2 | ; Herausfahren. |
| N50 M30 | ; Satzende. |

**Hinweis**

Zur Eingabe der Kreismittelpunktskoordinaten I und J siehe Kapitel "Kreisinterpolation".

### Beispiel 2: Drehen




| Programmcode | Kommentar |
|---|---|
| N5 T1 D1 S2000 M3 | ; Einwechseln von Werkzeug T1, Spindel ein mit Drehrichtung rechts. |
| N10 G0 G90 X11 Z1 | ; Absolutmaßangabe, im Eilgang auf Position XZ. |
| N20 G1 Z-15 F0.2 | ; Geradeninterpolation, Zustellen des Werkzeugs. |
| N30 G3 X11 Z-27 I-8 K-6 | ; Kreisinterpolation gegen Uhrzeigersinn, Kreisendpunkt im Absolutmaß, Kreismittelpunkt im Kettenmaß. |
| N40 G1 Z-40 | ; Herausfahren. |
| N50 M30 | ; Satzende. |

### Hinweis

Zur Eingabe der Kreismittelpunktskoordinaten I und J siehe Kapitel "Kreisinterpolation".

### Beispiel 3: Kettenmaßangabe ohne Herausfahren der aktiven Nullpunktverschiebung

Einstellungen:

- G54 enthält eine Verschiebung in X um 25
- SD42440 $SC_FRAME_OFFSET_INCR_PROG = 0

| Programmcode | Kommentar |
|---|---|
| N10 G90 G0 G54 X100 | |
| N20 G1 G91 X10 | ; Kettenmaßangabe aktiv, Fahren in X um 10mm (die Nullpunktverschiebung wird nicht gefahren). |
| N30 G90 X50 | ; Absolutmaßangabe aktiv, Fahren auf Position X75 (die Nullpunktverschiebung wird gefahren). |

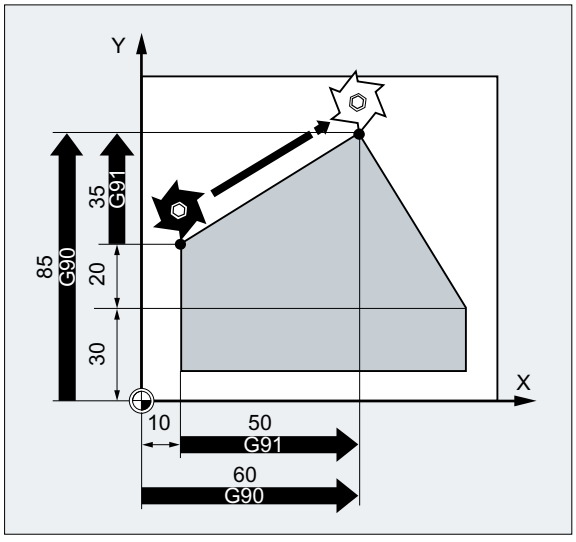
## Siehe auch

Absolut- und Kettenmaßangabe beim Drehen und Fräsen (G90/G91) (Seite 159)

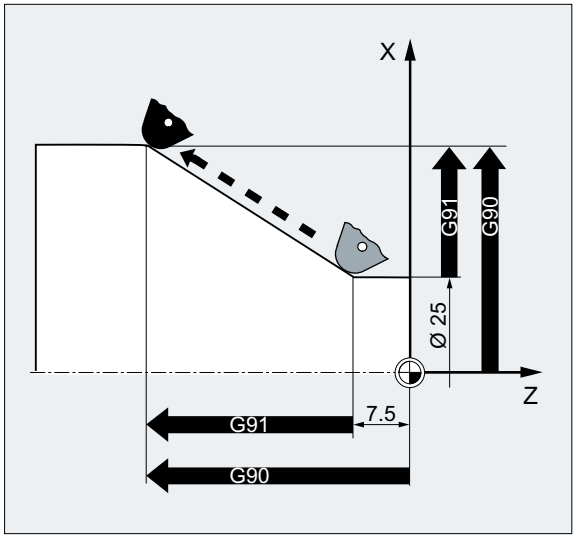
### 2.8.4.3 Absolut- und Kettenmaßangabe beim Drehen und Fräsen (G90/G91)

Die beiden folgenden Abbildungen veranschaulichen die Programmierung mit Absolutmaßangabe (`G90`) bzw. Kettenmaßangabe (`G91`) am Beispiel der Technologien Drehen und Fräsen.

**Fräsen:**




**Drehen:**



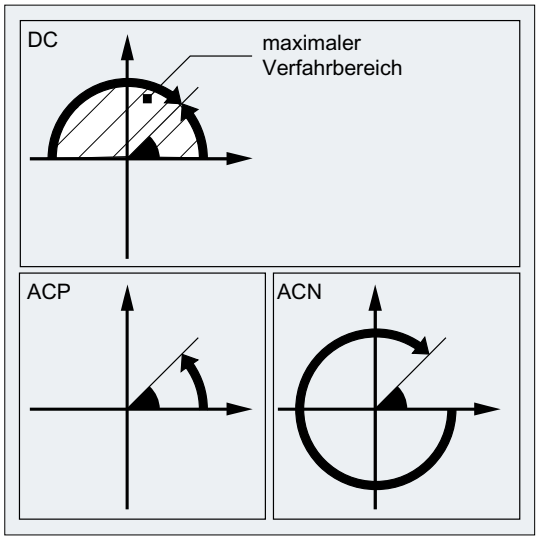
**Hinweis**

An konventionellen Drehmaschinen ist es üblich, inkrementelle Verfahrsätze in der Planachse als Radiuswerte zu betrachten, während Durchmesserangaben für die Bezugsmaße gelten. Diese Umstellung für `G90` erfolgt mit den Befehlen `DIAMON`, `DIAMOF` bzw. `DIAM90`.

### 2.8.4.4 Absolutmaßangabe für Rundachsen (DC, ACP, ACN)

Für die Positionierung von Rundachsen im Absolutmaß stehen die satzweise wirksamen und von G90/G91 unabhängigen Befehle DC, ACP und ACN zur Verfügung.

DC, ACP und ACN unterscheiden sich in der zu Grunde liegenden Anfahrstrategie:




## Syntax

```
<Rundachse>=DC(<Wert>)
<Rundachse>=ACP(<Wert>)
<Rundachse>=ACN(<Wert>)
```

## Bedeutung

| | |
|---|---|
| `<Rundachse>`: | Bezeichner der Rundachse, die verfahren werden soll (z. B. A, B oder C) |
| `DC`: | Befehl zum **direkten** Anfahren der Position<br>Die Rundachse fährt die programmierte Position auf direktem, kürzestem Weg an. Die Rundachse verfährt maximal in einem Bereich von 180°. |
| `ACP`: | Befehl zum Anfahren der Position in **positiver** Richtung<br>Die Rundachse fährt die programmierte Position in positiver Achsdrehrichtung (Gegenuhrzeigersinn) an. |

| | | |
|---|---|---|
| ACN: | Befehl zum Anfahren der Position in **negativer** Richtung<br>Die Rundachse fährt die programmierte Position in negativer Achsdrehrichtung (Uhrzeigersinn) an. | |
| <Wert>: | Anzufahrende Rundachsposition im Absolutmaß | |
| | Wertebereich: | 0 - 360 Grad |

**Hinweis**

Die positive Drehrichtung (Uhrzeiger- oder Gegenuhrzeigersinn) wird im Maschinendatum eingestellt.

**Hinweis**

Für die Positionierung mit Richtungsangabe (ACP, ACN) muss im Maschinendatum der Verfahrbereich zwischen 0° und 360° eingestellt sein (Modulo-Verhalten). Um Modulo-Rundachsen in einem Satz um mehr als 360° zu verfahren, ist G91 bzw. IC zu programmieren.

**Hinweis**

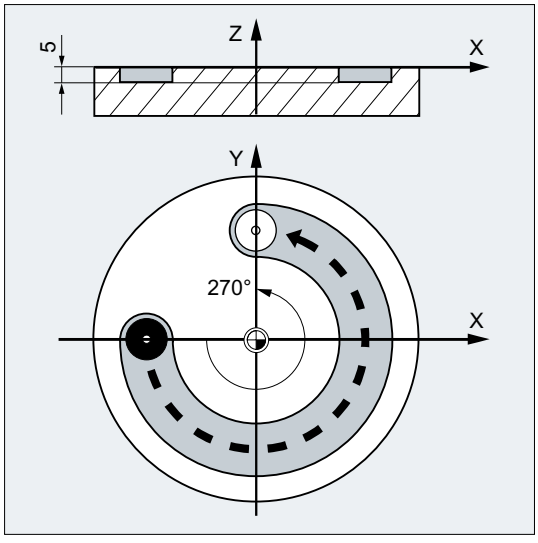
Die Befehle DC, ACP und ACN können auch für die Spindelpositionierung (SPOS, SPOSA) aus dem Stillstand genutzt werden.

Beispiel: SPOS=DC(45)

## Beispiel

### Fräsbearbeitung auf einem Rundtisch

Das Werkzeug steht, der Tisch dreht sich auf 270° **im Uhrzeigersinn**. Dabei entsteht eine Kreisnut.




| Programmcode | Kommentar |
|---|---|
| N10 SPOS=0 | ; Spindel in Lageregelung. |

| Programmcode | Kommentar |
|---|---|
| N20 G90 G0 X-20 Y0 Z2 T1 | ; Absolutmaßangabe, im Eilgang Werkzeug T1 zustellen. |
| N30 G1 Z-5 F500 | ; Im Vorschub Werkzeug absenken. |
| N40 C=ACP(270) | ; Tisch dreht sich auf 270 Grad im Uhrzeigersinn (positiv), das Werkzeug fräst eine Kreisnut. |
| N50 G0 Z2 M30 | ; Abheben, Programmende. |

### 2.8.4.5 Maßsystem inch/metrisch (G70/G71, G700/G710)

Mit den Befehlen der G-Gruppe 13 (Maßsystem inch/metrisch) kann innerhalb des Teileprogramms zwischen dem metrischen Maßsystem und dem Inch-Maßsystem umgeschaltet werden.

#### Aktivierung

Damit die Befehle G700 und G710 zur Verfügung stehen, muss die erweiterte Maßsystem-Funktionalität eingeschaltet werden (MD10260 $MN_CONVERT_SCALING_SYSTEM = 1).

## Syntax

```
G70
G71
G700
G710
```

## Bedeutung

| | | |
|---|---|---|
| G70: | Einschalten des Inch-Maßsystems<br>**Längenbehaftete geometrische** Daten werden im **Inch-Maßsystem** gelesen und geschrieben.<br>**Längenbehaftete technologische** Daten (z. B. Vorschübe, Werkzeugkorrekturen, einstellbare Nullpunktverschiebungen, Maschinendaten und Systemvariablen) werden im **parametrierten Grundsystem** gelesen und geschrieben. | |
| | G-Gruppe: | 13 |
| | Grundstellung: | Einstellbar über MD20150 $MC_GCODE_RESET_VALUES |
| | Wirksamkeit: | Modal |
| G71: | Einschalten des metrischen Maßsystems<br>**Längenbehaftete geometrische** Daten werden im **metrischen Maßsystem** gelesen und geschrieben.<br>**Längenbehaftete technologische** Daten (z. B. Vorschübe, Werkzeugkorrekturen, einstellbare Nullpunktverschiebungen, Maschinendaten und Systemvariablen) werden im **parametrierten Grundsystem** gelesen und geschrieben. | |
| | G-Gruppe: | 13 |
| | Grundstellung: | Einstellbar über MD20150 $MC_GCODE_RESET_VALUES |
| | Wirksamkeit: | Modal |

| | | |
|---|---|---|
| G700: | Einschalten des Inch-Maßsystems<br>Alle längenbehafteten geometrischen und technologischen Daten werden im Inch-Maßsystem gelesen und geschrieben. | |
| | G-Gruppe: | 13 |
| | Grundstellung: | Einstellbar über MD20150 $MC_GCODE_RESET_VALUES |
| | Wirksamkeit: | Modal |
| G710: | Einschalten des metrischen Maßsystems<br>Alle längenbehafteten geometrischen und technologischen Daten werden im metrischen Maßsystem gelesen und geschrieben. | |
| | G-Gruppe: | 13 |
| | Grundstellung: | Einstellbar über MD20150 $MC_GCODE_RESET_VALUES |
| | Wirksamkeit: | Modal |

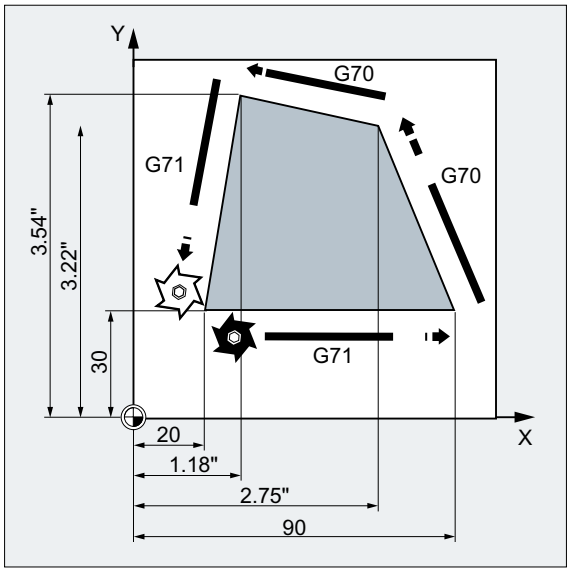
**ACHTUNG**

**Achsspezifische Daten von Rundachsen**

Achsspezifische Daten von Rundachsen werden immer im parametrierten Grundsystem gelesen und geschrieben.

## Beispiel

Das Grundsystem ist metrisch (MD10240 $MN_SCALING_SYSTEM_IS_METRIC = 1). Die Werkstückzeichnung enthält jedoch Maßangaben in inch. Es wird deshalb innerhalb des Teileprogramms auf das Inch-Maßsystem umgeschaltet. Nach den inch-Maßangaben wird wieder zum metrischen Maßsystem gewechselt.




| **Programmcode** | **Kommentar** |
|---|---|
| N10 G0 G90 X20 Y30 Z2 S2000 M3 T1 | ; X=20 mm, Y=30 mm, Z=2 mm, F=Eilgang mm/min |
| N20 G1 Z-5 F500 | ; Z=-5 mm, F=500 mm/min |

| **Programmcode** | **Kommentar** |
|---|---|
| N30 X90 | ; X=90 mm |
| N40 **G70** X2.75 Y3.22 | ; Programmiertes Maßsystem: inch |
| | ; X=2.75 inch, Y=3.22 inch, F=500 mm/min |
| N50 X1.18 Y3.54 | ; X=1.18 inch, Y=3.54 inch, F=500 mm/min |
| N60 **G71** X20 Y30 | ; Programmiertes Maßsystem: metrisch |
| | ; X=20 mm, Y=30 mm, F=500 mm/min |
| N70 G0 Z2 | ; Z=2 mm, F=Eilgang mm/min |
| N80 M30 | ; Programmende |

## Weitere Informationen

### Lesen und Schreiben von Daten bei G70/G71 und G700/G710

| Datenbereich | G70 / G71 | | G700 / G710 | |
|---|---|---|---|---|
| | Lesen | Schreiben | Lesen | Schreiben |
| Anzeigen, Nachkommastellen (WKS) | P | P | P | P |
| Anzeigen, Nachkommastellen (MKS) | G | G | G | G |
| Vorschübe | G | G | P | P |
| Positionsangaben X, Y, Z | P | P | P | P |
| Interpolationsparameter I, J, K | P | P | P | P |
| Kreisradius (CR) | P | P | P | P |
| Polarradius (RP) | P | P | P | P |
| Gewindesteigung | P | P | P | P |
| Programmierbarer FRAME | P | P | P | P |
| Einstellbare FRAMES | G | G | P | P |
| Basisframes | G | G | P | P |
| Externe Nullpunktverschiebungen | G | G | P | P |
| axiale Preset-Verschiebung | G | G | P | P |
| Arbeitsfeldbegrenzungen (G25/G26) | G | G | P | P |
| Schutzbereiche | P | P | P | P |
| Werkzeugkorrekturen | G | G | P | P |
| Längenbehaftete Maschinendaten | G | G | P | P |
| Längenbehaftete Settingdaten | G | G | P | P |
| Längenbehaftete Systemvariablen | G | G | P | P |
| GUD's | G | G | G | G |
| LUD's | G | G | G | G |
| PUD's | G | G | G | G |
| R-Parameter | G | G | G | G |
| Siemenszyklen | P | P | P | P |
| Inkrementbewertung JOG/Handrad | G | G | G | G |
| P: Schreiben/Lesen erfolgt im programmierten Maßsystem<br>G: Schreiben/Lesen erfolgt im parametrierten Grundsystem | | | | |

**Synchronaktionen**

**Hinweis**

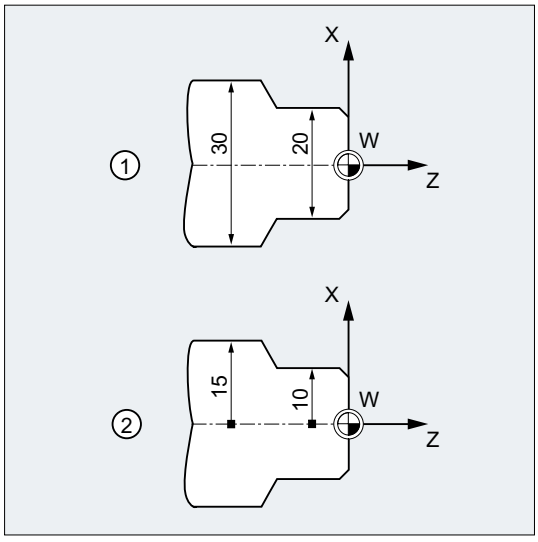
**Lesen von Positionsdaten in Synchronaktionen**

Ohne explizite Programmierung des Maßsystems in der Synchronaktion (Bedingungsteil und/ oder Aktionsteil bzw. Technologiefunktion) werden **längenbehaftete Positionsdaten** in der Synchronaktion immer im parametrierten **Grundsystem** gelesen.

**Weitere Informationen:** Funktionshandbuch Synchronaktionen

### 2.8.4.6 Kanalspezifische Durchmesser-/Radius-Programmierung (DIAMON, DIAM90, DIAMOF, DIAMCYCOF)

Beim Drehen können die Maße **für die Planachse** im Durchmesser (①) oder im Radius (②) angegeben sein:




Damit die Maßangaben direkt ohne Umrechnung aus der technischen Zeichnung in das NC-Programm übernommen werden können, wird über die modal wirksamen Befehle `DIAMON`, `DIAM90`, `DIAMOF` und `DIAMCYCOF` die kanalspezifische Durchmesser- oder Radius-Programmierung eingeschaltet.

**Hinweis**

Die kanalspezifische Durchmesser-/Radius-Programmierung bezieht sich auf die über MD20100 $MC_DIAMETER_AX_DEF als Planachse definierte Geometrieachse (→ siehe Angaben des Maschinenherstellers!).

Über MD20100 kann pro Kanal nur eine Planachse definiert sein.

## Syntax

```
DIAMON
```

```
DIAM90
DIAMOF
```

## Bedeutung

| | | |
|---|---|---|
| DIAMON: | Befehl zum Einschalten der **unabhängigen** kanalspezifischen Durchmesser-Programmierung<br>Die Wirkung von DIAMON ist unabhängig vom programmierten Maßangabe-Modus (Absolutmaßangabe G90 oder Kettenmaßangabe G91): | |
| | • bei G90: | Maßangabe im Durchmesser |
| | • bei G91: | Maßangabe im Durchmesser |
| DIAM90: | Befehl zum Einschalten der **abhängigen** kanalspezifischen Durchmesser-Programmierung<br>Die Wirkung von DIAM90 ist abhängig vom programmierten Maßangabe-Modus: | |
| | • bei G90: | Maßangabe im Durchmesser |
| | • bei G91: | Maßangabe im Radius |
| DIAMOF: | Befehl zum Ausschalten der kanalspezifischen Durchmesser-Programmierung<br>Mit dem Ausschalten der Durchmesser-Programmierung wird die kanalspezifische Radius-Programmierung wirksam. Die Wirkung von DIAMOF ist unabhängig vom programmierten Maßangabe-Modus: | |
| | • bei G90: | Maßangabe im Radius |
| | • bei G91: | Maßangabe im Radius |
| DIAMCYCOF: | Befehl zum Ausschalten der kanalspezifischen Durchmesser-Programmierung während der Zyklusbearbeitung<br>Im Zyklus können somit Berechnungen immer im Radius erfolgen. Für die Positionsanzeige und die Basis-Satzanzeige bleibt der zuletzt aktive G-Befehl dieser Gruppe aktiv. | |

### Hinweis

Mit DIAMON oder DIAM90 werden die Istwerte der Planachse immer als Durchmesser angezeigt. Das gilt auch für das Lesen der Istwerte im Werkstückkoordinatensystem bei MEAS, MEAW, $P_EP[x] und $AA_IW[x].

## Beispiel

| **Programmcode** | **Kommentar** |
|---|---|
| N10 G0 X0 Z0 | ; Startpunkt anfahren. |
| N20 DIAMOF | ; Durchmesser-Programmierung aus. |
| N30 G1 X30 S2000 M03 F0.7 | ; X-Achse = Planachse, Radius-Programmierung aktiv, Fahren auf Radius-Position X30. |
| N40 DIAMON | ; Für die Planachse ist Durchmesser-Programmierung aktiv. |
| N50 G1 X70 Z-20 | ; Fahren auf Durchmesserposition X70 und Z-20. |
| N60 Z-30 | |

| Programmcode | Kommentar |
|---|---|
| N70 DIAM90 | ; Durchmesser-Programmierung für Bezugsmaß und Radius-Programmierung für Kettenmaß. |
| N80 G91 X10 Z-20 | ; Kettenmaß aktiv. |
| N90 G90 X10 | ; Bezugsmaß aktiv. |
| N100 M30 | ; Programmende. |

## Weitere Informationen

### Durchmesserwerte (DIAMON/DIAM90)

Die Durchmesserwerte gelten für folgende Daten:

- Istwertanzeige der Planachse im Werkstückkoordinatensystem
- JOG-Betrieb: Inkremente für Schrittmaß und manuelles Verfahren per Handrad
- Programmierung von Endpositionen:
  Interpolationsparameter `I`, `J`, `K` bei `G2`/`G3`, falls diese mit `AC` absolut programmiert sind.
  Bei inkrementeller Programmierung (`IC`) von `I`, `J`, `K` wird immer Radius verrechnet.
- Istwerte lesen im Werkstückkoordinatensystem bei:
  `MEAS`, `MEAW`, `$P_EP[X]`, `$AA_IW[X]`

## 2.8.4.7 Achsspezifische Durchmesser-/Radius-Programmierung (DIAMONA, DIAM90A, DIAMOFA, DIACYCOFA, DIAMCHANA, DIAMCHAN, DAC, DIC, RAC, RIC)

Zusätzlich zur kanalspezifischen Durchmesser-Programmierung ermöglicht die achsspezifische Durchmesser-Programmierung für eine oder mehrere Achsen die modal oder satzweise wirksame Maßangabe und Anzeige im Durchmesser.

### Hinweis

Die achsspezifische Durchmesser-Programmierung ist nur möglich bei Achsen, die über MD30460 $MA_BASE_FUNCTION_MASK als weitere Planachsen für die achsspezifische Durchmesser-Programmierung zugelassen sind (→ siehe Angaben des Maschinenherstellers!).

## Syntax

Modal wirksame achsspezifische Durchmesser-Programmierung für mehrere Planachsen im Kanal:

```
DIAMONA[<Achse>]
DIAM90A[<Achse>]
DIAMOFA[<Achse>]
DIACYCOFA[<Achse>]
```

Übernahme der kanalspezifischen Durchmesser-/Radius-Programmierung:

```
DIAMCHANA[<Achse>]
DIAMCHAN
```

Satzweise wirksame achsspezifische Durchmesser-/Radius-Programmierung:

```
<Achse>=DAC(<Wert>)
<Achse>=DIC(<Wert>)
<Achse>=RAC(<Wert>)
<Achse>=RIC(<Wert>)
```

## Bedeutung

| **Modal wirksame achsspezifische Durchmesser-Programmierung** | | |
|---|---|---|
| DIAMONA: | Befehl zum Einschalten der **unabhängigen** achsspezifischen Durchmesser-Programmierung<br>Die Wirkung von DIAMONA ist unabhängig vom programmierten Maßangabe-Modus (G90/G91 bzw. AC/IC): | |
| | • bei G90, AC: | Maßangabe im Durchmesser |
| | • bei G91, IC: | Maßangabe im Durchmesser |
| DIAM90A: | Befehl zum Einschalten der **abhängigen** achsspezifischen Durchmesser-Programmierung<br>Die Wirkung von DIAM90A ist abhängig vom programmierten Maßangabe-Modus: | |
| | • bei G90, AC: | Maßangabe im Durchmesser |
| | • bei G91, IC: | Maßangabe im Radius |
| DIAMOFA: | Befehl zum Ausschalten der achsspezifischen Durchmesser-Programmierung<br>Mit dem Ausschalten der Durchmesser-Programmierung wird die achsspezifische Radius-Programmierung wirksam. Die Wirkung von DIAMOFA ist unabhängig vom programmierten Maßangabe-Modus: | |
| | • bei G90, AC: | Maßangabe im Radius |
| | • bei G91, IC: | Maßangabe im Radius |
| DIACYCOFA: | Befehl zum Ausschalten der achsspezifischen Durchmesser-Programmierung während der Zyklusbearbeitung<br>Im Zyklus können somit Berechnungen immer im Radius erfolgen. Für die Positionsanzeige und die Basis-Satzanzeige bleibt der zuletzt aktive G-Befehl dieser Gruppe aktiv. | |
| <Achse>: | Achsbezeichner der Achse, für die die achsspezifische Durchmesser-Programmierung aktiviert werden soll<br>Zugelassene Achsbezeichner sind:<br>• Geometrie-/Kanalachsname<br>oder<br>• Maschinenachsname | |
| | Wertebereich: | Die angegebene Achse muss eine im Kanal bekannte Achse sein.<br>Sonstige Bedingungen:<br>• Die Achse muss über MD30460 $MA_BASE_FUNCTION_MASK für die achsspezifische Durchmesser-Programmierung zugelassen sein.<br>• Rundachsen sind als Planachsen nicht zugelassen. |
| **Übernahme der kanalspezifischen Durchmesser-/Radius-Programmierung** | | |

| | |
|---|---|
| DIAMCHANA: | Mit dem Befehl DIAMCHANA[<Achse>] übernimmt die **angegebene Achse** den Kanalzustand der Durchmesser-/Radius-Programmierung und wird in Folge der kanalspezifischen Durchmesser-/Radius-Programmierung unterstellt. |
| DIAMCHAN: | Mit dem Befehl DIAMCHAN übernehmen **alle** für die achsspezifische Durchmesser-Programmierung zugelassenen Achsen den Kanalzustand der Durchmesser-/Radius-Programmierung und werden in Folge der kanalspezifischen Durchmesser-/Radius-Programmierung unterstellt. |
| **Satzweise wirksame achsspezifische Durchmesser-/Radius-Programmierung**<br>Die satzweise wirksame achsspezifische Durchmesser-/Radius-Programmierung legt die Art der Maßangabe als Durchmesser- oder Radiuswert im Teileprogramm und Synchronaktionen fest. Der modale Zustand der Durchmesser-/Radius-Programmierung wird nicht verändert. | |
| DAC: | Mit dem Befehl DAC ist für die angegebene Achse folgende Maßangabe satzweise wirksam:<br>Durchmesser im Absolutmaß |
| DIC: | Mit dem Befehl DIC ist für die angegebene Achse folgende Maßangabe satzweise wirksam:<br>Durchmesser im Kettenmaß |
| RAC: | Mit dem Befehl RAC ist für die angegebene Achse folgende Maßangabe satzweise wirksam:<br>Radius im Absolutmaß |
| RIC: | Mit dem Befehl RIC ist für die angegebene Achse folgende Maßangabe satzweise wirksam:<br>Radius im Kettenmaß |

**Hinweis**

Mit DIAMONA[<Achse>] oder DIAM90A[<Achse>] werden die Istwerte der Planachse immer als Durchmesser angezeigt. Das gilt auch für das Lesen der Istwerte im Werkstückkoordinatensystem bei MEAS, MEAW, $P_EP[x] und $AA_IW[x].

**Hinweis**

Beim Achstausch einer zusätzlichen Planachse aufgrund einer GET-Anforderung wird mit RELEASE[<Achse>] der Zustand der Durchmesser-/Radius-Programmierung im anderen Kanal übernommen.

## Beispiele

### Beispiel 1: Modal wirksame achsspezifische Durchmesser-/Radius-Programmierung

X ist Planachse im Kanal, für Y ist achsspezifische Durchmesser-Programmierung zugelassen.

| Programmcode | Kommentar |
|---|---|
| N10 G0 X0 Z0 DIAMON | ; Kanalspezifische Durchmesser-Programmierung aktiv für X. |
| N15 DIAMOF | ; Kanalspezifische Durchmesser-Programmierung aus. |
| N20 DIAMONA[Y] | ; Modal wirksame achsspezifische Durchmesser-Programmierung aktiv für Y. |
| N25 X200 Y100 | ; Radius-Programmierung aktiv für X. |

| Programmcode | Kommentar |
|---|---|
| N30 DIAMCHANA[Y] | ; Y übernimmt den Zustand der kanalspezifischen Durchmesser-/Radius-Programmierung und ist dieser unterstellt |
| N35 X50 Y100 | ; Radius-Programmierung aktiv für X und Y. |
| N40 DIAMON | ; Kanalspezifische Durchmesser-Programmierung ein. |
| N45 X50 Y100 | ; Durchmesser-Programmierung aktiv für X und Y. |

### Beispiel 2: Satzweise wirksame achsspezifische Durchmesser-/Radius-Programmierung

X ist Planachse im Kanal, für Y ist achsspezifische Durchmesser-Programmierung zugelassen.

| Programmcode | Kommentar |
|---|---|
| N10 DIAMON | ; Kanalspezifische Durchmesser-Programmierung ein. |
| N15 G0 G90 X20 Y40 DIAMONA[Y] | ; Modal wirksame achsspezifische Durchmesser-Programmierung aktiv für Y. |
| N20 G01 X=RIC(5) | ; Für diesen Satz wirksame Maßangabe für X: Radius im Kettenmaß. |
| N25 X=RAC(80) | ; Für diesen Satz wirksame Maßangabe für X: Radius im Absolutmaß. |
| N30 WHEN $SAA_IM[Y]>50 DO POS[X]=RIC(1) | ; X ist Kommandoachse. Für diesen Satz wirksame Maßangabe für X: Radius im Kettenmaß. |
| N40 WHEN $SAA_IM[Y]>60 DO POS[X]=DAC(10) | ; X ist Kommandoachse. Für diesen Satz wirksame Maßangabe für X: Radius im Absolutmaß. |
| N50 G4 F3 | |

## Weitere Informationen

### Durchmesserwerte (DIAMONA/DIAM90A)

Die Durchmesserwerte gelten für folgende Daten:

- Istwertanzeige der Planachse im Werkstückkoordinatensystem
- JOG-Betrieb: Inkremente für Schrittmaß und manuelles Verfahren per Handrad
- Programmierung von Endpositionen:
  Interpolationsparameter `I`, `J`, `K` bei `G2`/`G3`, falls diese mit `AC` absolut programmiert sind.
  Bei inkrementeller Programmierung `IC` von `I`, `J`, `K` wird immer Radius verrechnet.
- Istwerte lesen im Werkstückkoordinatensystem bei:
  `MEAS`, `MEAW`, `$P_EP[X]`, `$AA_IW[X]`

### Satzweise wirksame achsspezifische Durchmesserprogrammierung (DAC, DIC, RAC, RIC)

Die Anweisungen DAC, DIC, RAC, RIC sind für alle Befehle zugelassen, für die die kanalspezifische Durchmesserprogrammierung berücksichtigt wird:

- Achsposition: X..., POS, POSA
- Pendeln: OSP1, OSP2, OSS, OSE, POSP
- Interpolationsparameter: I, J, K
- Konturzug: Gerade mit Winkelangabe
- Schnellabheben: POLF[AX]
- Verfahren in Werkzeugrichtung: MOVT
- Weiches An– und Abfahren:
  G140 bis G143, G147, G148, G247, G248, G347, G348, G340, G341

## 2.8.5 Lage des Werkstücks beim Drehen

### Achsbezeichnungen

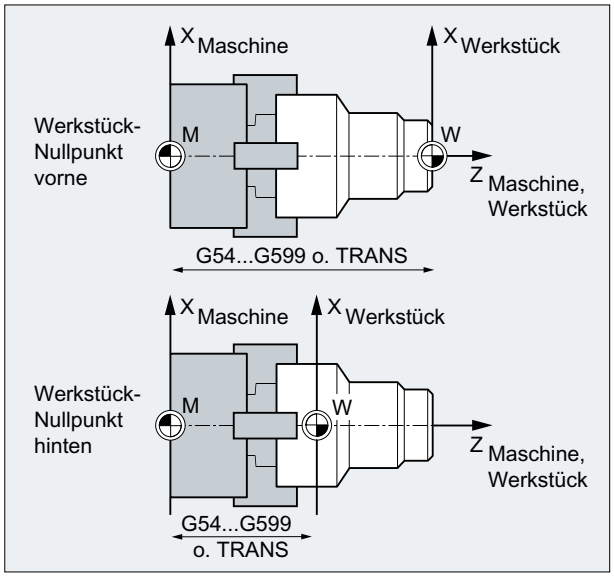
Die beiden aufeinander senkrecht stehenden Geometrieachsen werden üblicherweise bezeichnet als:

| | |
|---|---|
| **Längsachse** | = Z-Achse (Abszisse) |
| **Planachse** | = X-Achse (Ordinate) |

### Werkstück-Nullpunkt

Während der Maschinen-Nullpunkt fest vorgegeben ist, ist die Lage des Werkstück-Nullpunkts auf der Längsachse frei wählbar. Im Allgemeinen liegt der Werkstück-Nullpunkt an der Vorder- oder Hinterseite des Werkstücks.

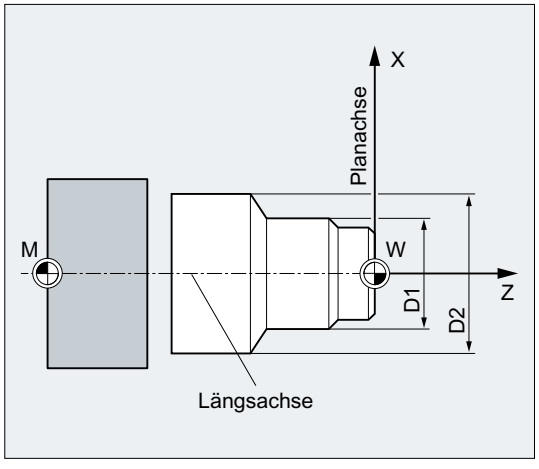
Sowohl Maschinen- als auch Werkstück-Nullpunkt liegen auf Drehmitte. Die einstellbare Verschiebung in der X-Achse ergibt sich damit zu Null.




| | |
|---|---|
| M | Maschinen-Nullpunkt |
| W | Werkstück-Nullpunkt |
| `Z` | Längsachse |
| `X` | Planachse |
| `G54` bis `G599` oder `TRANS` | Aufruf für die Lage des Werkstück-Nullpunkts |

## Planachse

Für die Planachse erfolgen die Maßangaben im Allgemeinen als Durchmesser-Angaben (doppeltes Wegmaß gegenüber den anderen Achsen):




Welche Geometrieachse als Planachse dient, ist im Maschinendatum festzulegen (→ Maschinenhersteller!).

# 2.9 Wegbefehle

## 2.9.1 Allgemeine Informationen zu den Wegbefehlen

### Konturelemente

Die programmierte Werkstückkontur kann sich aus folgenden Konturelementen zusammensetzen:

- Geraden
- Kreisbögen
- Schraubenlinien (durch Überlagerung von Geraden und Kreisbögen)

### Fahrbefehle

Zur Herstellung dieser Konturelemente stehen verschiedene Fahrbefehle zur Verfügung:

- Eilgangbewegung (`G0`)
- Geradeninterpolation (`G1`)
- Kreisinterpolation im Uhrzeigersinn (`G2`)
- Kreisinterpolation gegen den Uhrzeigersinn (`G3`)

Die Fahrbefehle sind modal wirksam.

### Zielpositionen

Ein Bewegungssatz enthält die Zielpositionen für die zu verfahrenden Achsen (Bahnachsen, Synchronachsen, Positionierachsen).

Die Programmierung der Zielpositionen kann in kartesischen Koordinaten oder in Polarkoordinaten erfolgen.

**Hinweis**

Eine Achsadresse darf pro Satz nur einmal programmiert werden.

### Startpunkt-Zielpunkt

Die Fahrbewegung verläuft immer von der zuletzt angefahrenen Position zur programmierten Zielposition. Diese Zielposition ist wiederum die Startposition für den nächsten Fahrbefehl.

## Werkstückkontur

| ACHTUNG |
| --- |
| **Werkzeugeingriff undefiniert** |
| Vor Beginn eines Bearbeitungsablaufs muss das Werkzeug so vorpositionieren werden, dass eine Beschädigung von Werkzeug und Werkstück ausgeschlossen ist. |

Nacheinander ausgeführt ergeben die Bewegungssätze die Werkstückkontur:

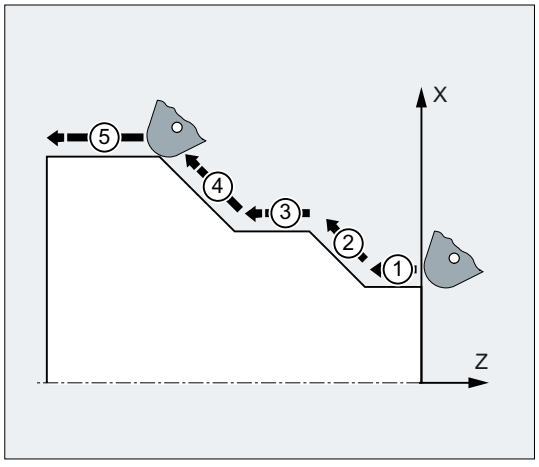



Bild 2-1 Bewegungssätze beim Drehen

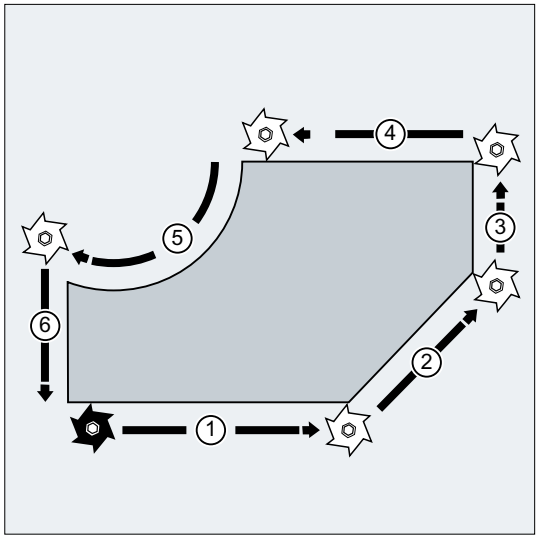



Bild 2-2 Bewegungssätze beim Fräsen

## 2.9.2 Fahrbefehle mit kartesischen Koordinaten (G0, G1, G2, G3, X..., Y..., Z...)

Die im NC-Satz mit kartesischen Koordinaten angegebene Position kann mit Eilgangbewegung `G0`, Geradeninterpolation `G1` oder Kreisinterpolation `G2` /`G3` angefahren werden.

### Syntax

```
G0 X... Y... Z...
G1 X... Y... Z...
G2 X... Y... Z... ...
G3 X... Y... Z... ...
```

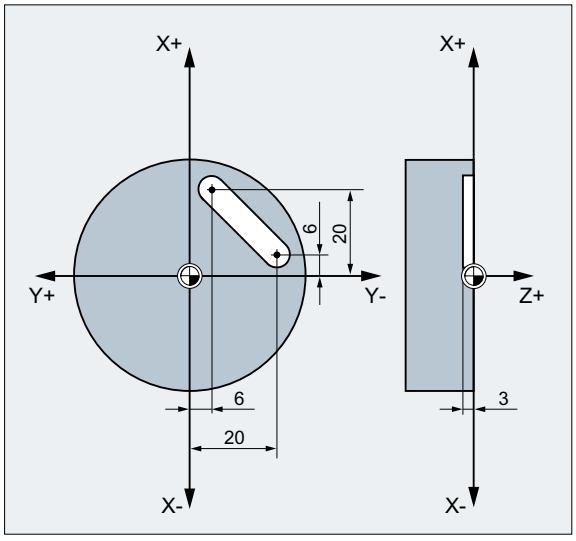
### Bedeutung

| | |
|---|---|
| `G0`: | Befehl zum Einschalten der Eilgangbewegung |
| `G1`: | Befehl zum Einschalten der Geradeninterpolation |
| `G2`: | Befehl zum Einschalten der Kreisinterpolation im Uhrzeigersinn |
| `G3`: | Befehl zum Einschalten der Kreisinterpolation gegen den Uhrzeigersinn |
| `X...`: | Kartesische Koordinate der Zielposition in X-Richtung |
| `Y...`: | Kartesische Koordinate der Zielposition in Y-Richtung |
| `Z...`: | Kartesische Koordinate der Zielposition in Z-Richtung |

**Hinweis**

Die Kreisinterpolation `G2` / `G3` benötigt außer den Koordinaten der Zielposition `X...`, `Y...`, `Z...` noch weitere Angaben (z. B. die Kreismittelpunkt-Koordinaten; siehe "Übersicht (Seite 191)").

### Beispiel



| **Programmcode** | **Kommentar** |
|---|---|
| `N10 G17 S400 M3` | `; Wahl der Arbeitsebene, Spindel rechts` |
| `N20 G0 X40 Y-6 Z2` | `; Anfahren der mit kartesischen Koordinaten angegebenen Start-position im Eilgang` |
| `N30 G1 Z-3 F40` | `; Einschalten der Geradeninterpolation, Zustellen des Werk-zeugs` |
| `N40 X12 Y-20` | `; Fahren auf einer schräg liegenden Geraden auf die mit kar-tesischen Koordinaten angegebene Endposition` |
| `N50 G0 Z100 M30` | `; Freifahren zum Werkzeugwechsel im Eilgang` |

## 2.9.3 Fahrbefehle mit Polarkoordinaten

### 2.9.3.1 Bezugspunkt der Polarkoordinaten (G110, G111, G112)

Der Punkt, von dem die Vermaßung ausgeht, heißt Pol.

Die Angabe des Pols kann in kartesischen oder polaren Koordinaten erfolgen.

Mit den Befehlen `G110` bis `G112` wird der Bezugspunkt für die Polkoordinaten eindeutig festgelegt. Absolut- oder Kettenmaßeingabe haben deshalb keinen Einfluss.

#### Syntax

```
G110/G111/G112 X… Y… Z…
G110/G111/G112 AP=… RP=…
```

#### Bedeutung

| | | | |
|---|---|---|---|
| `G110 ...`: | Mit dem Befehl `G110` beziehen sich die nachfolgenden Polkoordinaten **auf die zuletzt angefahrene Position**. | | |
| `G111 ...`: | Mit dem Befehl `G111` beziehen sich die nachfolgenden Polkoordinaten **auf den Nullpunkt des aktuellen Werkstück-Koordinatensystems**. | | |
| `G112 ...`: | Mit dem Befehl `G112` beziehen sich die nachfolgenden Polkoordinaten **auf den zuletzt gültigen Pol**. | | |
| | **Hinweis:**<br>Die Befehle `G110...G112` müssen im eigenen NC-Satz programmiert werden. | | |
| `X… Y… Z…`: | Angabe des Pols in kartesischen Koordinaten | | |
| `AP=… RP=…`: | Angabe des Pols in Polarkoordinaten | | |
| | `AP=…`: | Polarwinkel<br>Winkel zwischen dem Polarradius und der waagrechten Achse der Arbeitsebene (z. B. X-Achse bei `G17`). Die positive Drehrichtung läuft im Gegenuhrzeigersinn. | |
| | | Wertebereich: | ± 0…360° |
| | `RP=…`: | Polarradius<br>Die Angabe erfolgt **immer in absoluten positiven Werten** in [mm] oder [inch]. | |

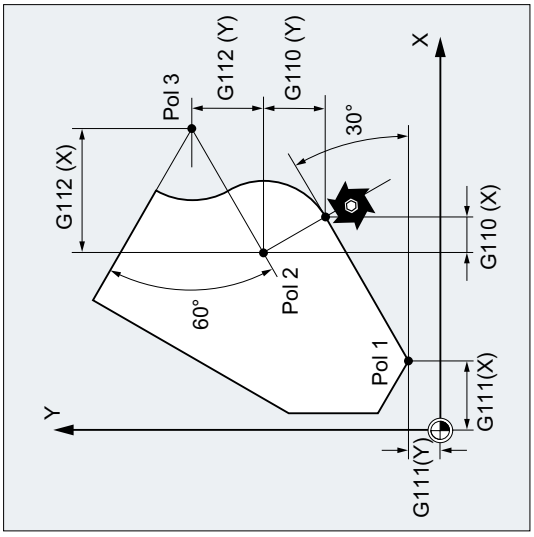
**Hinweis**

Es ist möglich, im NC-Programm satzweise zwischen polaren und kartesischen Maßangaben zu wechseln. Durch Verwendung der kartesischen Koordinatenbezeichner (X..., Y..., Z...) kommt man direkt wieder in das kartesische System zurück. Der definierte Pol bleibt darüber hinaus bis Programmende erhalten.

**Hinweis**

Falls kein Pol angegeben wird, gilt der Nullpunkt des aktuellen Werkstück-Koordinatensystems.

## Beispiel

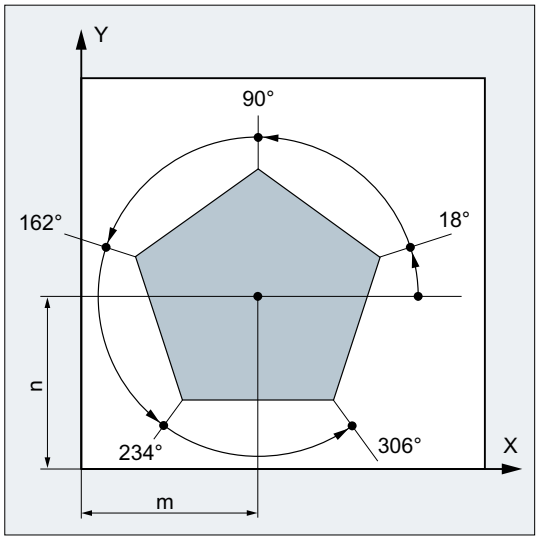



Die Pole 1 bis 3 werden wie folgt definiert:

- Pol 1 mit `G111 X… Y…`
- Pol 2 mit `G110 X… Y…`
- Pol 3 mit `G112 X… Y…`

### 2.9.3.2 Fahrbefehle mit Polarkoordinaten (G0, G1, G2, G3, AP, RP)

Fahrbefehle mit Polarkoordinaten sind dann sinnvoll, wenn die Bemaßung eines Werkstücks oder eines Teils eines Werkstücks von einem zentralen Punkt ausgeht und die Maße mit Winkeln und Radien angegeben sind (z. B. bei Bohrbildern).




## Syntax

```
G0/G1/G2/G3 AP=… RP=…
```

## Bedeutung

| | | |
|---|---|---|
| G0: | Befehl zum Einschalten der Eilgangbewegung | |
| G1: | Befehl zum Einschalten der Geradeninterpolation | |
| G2: | Befehl zum Einschalten der Kreisinterpolation im Uhrzeigersinn | |
| G3: | Befehl zum Einschalten der Kreisinterpolation gegen den Uhrzeigersinn | |
| AP: | Polarwinkel<br>Winkel zwischen dem Polarradius und der waagrechten Achse der Arbeitsebene (z. B. X-Achse bei G17). Die positive Drehrichtung läuft im Gegenuhrzeigersinn. | |
| | Wertebereich: | ± 0…360° |
| | Die Winkelangabe kann sowohl absolut als auch inkrementell erfolgen: | |
| | AP=AC(...): | Absolutmaßeingabe |
| | AP=IC(...): | Kettenmaßeingabe<br>Bei Kettenmaßeingabe gilt der zuletzt programmierte Winkel als Bezug. |
| | Der Polarwinkel bleibt solange gespeichert, bis ein neuer Pol definiert oder die Arbeitsebene gewechselt wird. | |
| RP: | Polarradius<br>Die Angabe erfolgt **immer in absoluten positiven Werten** in [mm] oder [inch].<br>Der Polarradius bleibt bis zur Eingabe eines neuen Werts gespeichert. | |

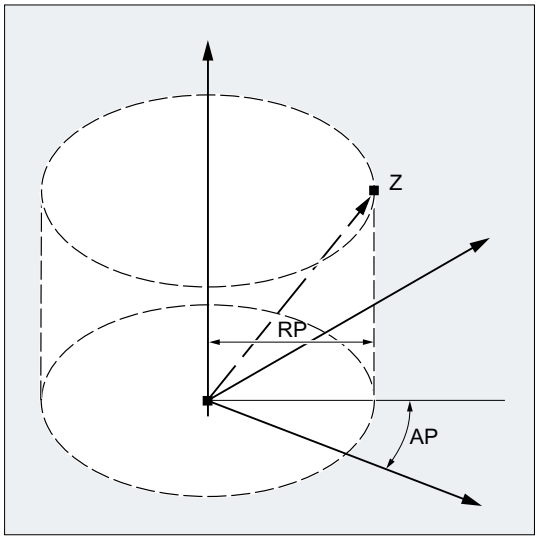
**Hinweis**

Die Polarkoordinaten beziehen sich auf den mit `G110` ... `G112` festgelegten Pol und gelten in der mit `G17` bis `G19` gewählten Arbeitsebene.

**Hinweis**

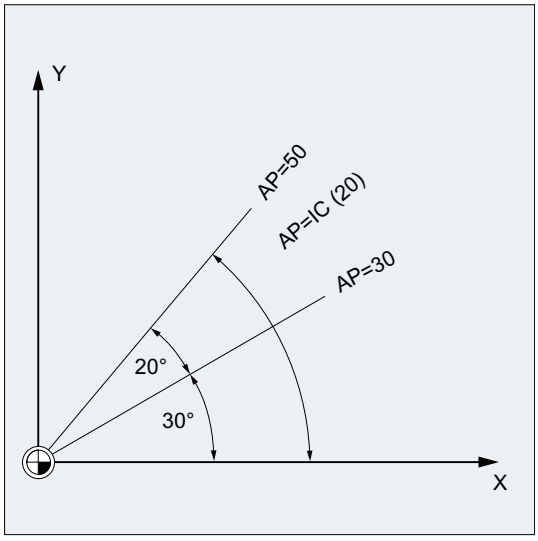
Die senkrecht zur Arbeitsebene stehende 3. Geometrieachse kann zusätzlich als kartesische Koordinate angegeben werden (siehe folgendes Bild). Damit sind räumliche Angaben in Zylinderkoordinaten programmierbar.

Beispiel: `G17 G0 AP… RP… Z…`



## Randbedingungen

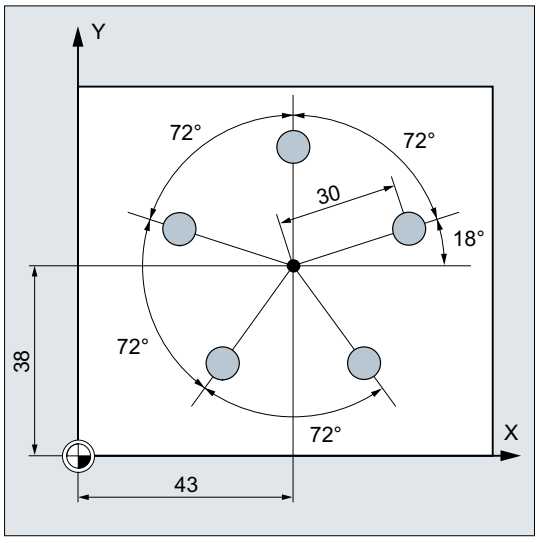
- In NC-Sätzen mit polaren Endpunktangaben dürfen für die angewählte Arbeitsebene keine kartesischen Koordinaten wie Interpolationsparameter, Achsadressen, usw. programmiert werden.
- Wenn mit `G110` ... `G112` kein Pol definiert wird, dann wird automatisch der Nullpunkt des aktuellen Werkstückkoordinatensystems als Pol betrachtet:




- Polarradius RP = 0
  Der Polarradius errechnet sich aus dem Abstand zwischen Startpunktvektor in der Polebene und dem aktiven Polvektor. Anschließend wird der errechnete Polarradius modal gespeichert.
  Das gilt unabhängig von einer gewählten Poldefinition (`G110` ... `G112`). Sind beide Punkte identisch programmiert, so wird dieser Radius = 0 und der Alarm 14095 generiert.
- Nur Polarwinkel AP ist programmiert
  Wenn im aktuellen Satz kein Polarradius RP, aber ein Polarwinkel AP programmiert ist, dann wird bei einer Differenz zwischen aktueller Position und Pol in Werkstückkoordinaten diese Differenz als Polarradius genutzt und modal gespeichert. Ist die Differenz = 0, werden erneut die Polkoordinaten vorgegeben und der modale Polarradius bleibt auf Null.

## Beispiel

### Herstellung eines Bohrbilds




Die Positionen der Bohrungen sind in Polarkoordinaten angegeben.

Jede Bohrung wird mit dem gleichen Fertigungsablauf hergestellt:

Vorbohren, Bohren auf Maß, Reiben …

Die Bearbeitungsfolge ist im Unterprogramm abgelegt.

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 G17 G54` | `; Arbeitsebene X/Y, Werkstücknullpunkt.` |
| `N20 G111 X43 Y38` | `; Festlegung des Pols.` |
| `N30 G0 RP=30 AP=18 Z5` | `; Startpunkt anfahren, Angabe in Zylinderkoordinaten.` |
| `N40 L10` | `; Unterprogramm-Aufruf.` |
| `N50 G91 AP=72` | `; Nächste Position im Eilgang anfahren, Polarwinkel im Kettenmaß, Polarradius von Satz N30 bleibt gespeichert und muss nicht angeben werden.` |
| `N60 L10` | `; Unterprogramm-Aufruf.` |
| `N70 AP=IC(72)` | `.` |
| `N80 L10` | … |
| `N90 AP=IC(72)` | |
| `N100 L10` | … |
| `N110 AP=IC(72)` | |
| `N120 L10` | … |
| `N130 G0 X300 Y200 Z100 M30` | `; Werkzeug freifahren, Programmende.` |

## Siehe auch

Übersicht (Seite 191)

## 2.9.4 Eilgangbewegungen

### 2.9.4.1 Eilgang aktivieren (G0)

Das Verfahren der Bahnachsen mit Eilganggeschwindigkeit wird eingeschaltet mit dem G-Befehl G0.

### Syntax

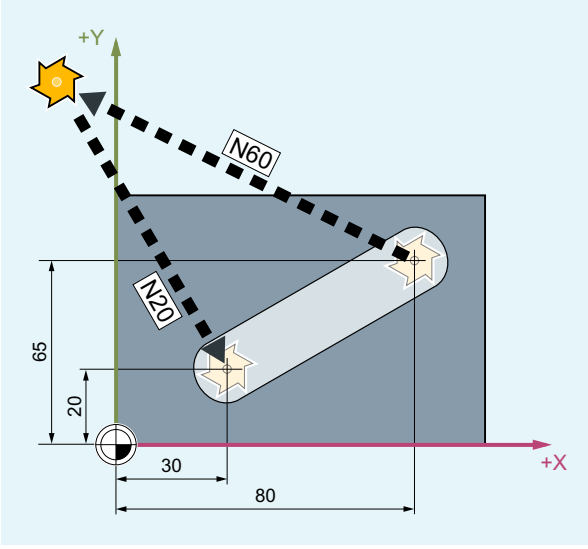
```
G0 X... Y... Z...
G0 RP=... AP=...
```

### Bedeutung

| | | |
|---|---|---|
| G0: | Verfahren der Achsen mit Eilgangsgeschwindigkeit | |
| | Wirksamkeit: | modal |
| X... Y... Z...: | Angabe des Endpunkts in kartesischen Koordinaten | |
| RP=... AP=... : | Angabe des Endpunkts in Polarkoordinaten | |

### Beispiele

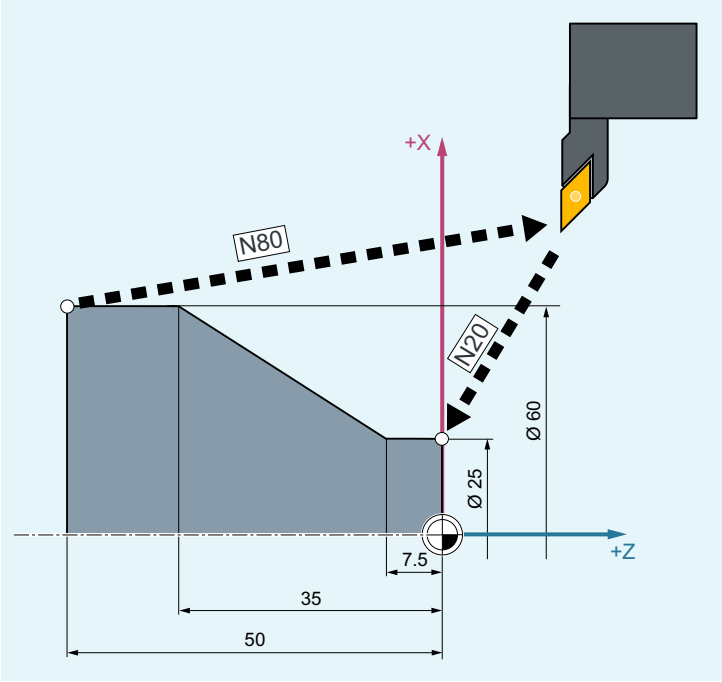
#### Beispiel 1: Fräsen




| Programmcode | Kommentar |
|---|---|
| N10 G90 S400 M3 | ; Absolutmaßeingabe, Spindel rechts |
| N20 G0 X30 Y20 Z2 | ; Anfahren der Startposition |
| N30 G1 Z-5 F1000 | ; Zustellen des Werkzeugs |
| N40 X80 Y65 | ; Fahren auf einer Geraden |
| N50 G0 Z2 | |

| Programmcode | Kommentar |
|---|---|
| N60 G0 X-20 Y100 Z100 M30 | ; Werkzeug freifahren, Programmende |

### Beispiel 2: Drehen




| Programmcode | Kommentar |
|---|---|
| N10 G90 S400 M3 | ; Absolutmaßeingabe, Spindel rechts |
| N20 G0 X25 Z5 | ; Anfahren der Startposition |
| N30 G1 G94 Z0 F1000 | ; Zustellen des Werkzeugs |
| N40 G95 Z-7.5 F0.2 | |
| N50 X60 Z-35 | ; Fahren auf einer Geraden |
| N60 Z-50 | |
| N70 G0 X62 | |
| N80 G0 X80 Z20 M30 | ; Werkzeug Freifahren, Programmende |

## 2.9.4.2 Lineare Interpolation für Eilgangbewegungen ein-/ausschalten (RTLION, RTLIOF)

Unabhängig von der Voreinstellung (MD20730 $MC_G0_LINEAR_MODE) kann das Interpolationsverhalten bei Eilgangbewegungen mit den Befehlen der G-Gruppe 55 auch im Teileprogramm eingestellt werden.

### Syntax

```
RTLIOF
...
RTLION
```

## Bedeutung

| | | |
|---|---|---|
| RTLIOF: | G-Befehl zum Ausschalten der linearen Interpolation<br>⇒ Bei Eilgang (G0) ist die **nicht-lineare** Interpolation aktiv. Alle Bahnachsen erreichen unabhängig voneinander ihren Endpunkt. | |
| | Wirksamkeit: | modal |
| RTLION: | G-Befehl zum Einschalten der linearen Interpolation<br>⇒ Bei Eilgang (G0) ist die **lineare** Interpolation aktiv. Alle Bahnachsen erreichen gleichzeitig ihren Endpunkt. | |
| | Wirksamkeit: | modal |

**Hinweis**

**Voraussetzungen für RTLIOF**

Damit bei RTLIOF **nicht-linear** interpoliert wird, müssen folgende Voraussetzungen erfüllt sein:

- Keine Transformation (TRAORI, TRANSMIT, ...) aktiv.
- G60 aktiv (Anhalten am Satzende).
- Kein Kompressor aktiv (COMPOF).
- Keine Werkzeugradiuskorrektur aktiv (G40).
- Kein Konturhandrad angewählt.
- Kein Nibbeln aktiv.

Ist eine dieser Voraussetzungen nicht erfüllt, wird wie bei RTLION linear interpoliert.

## Beispiel

```
Programmcode                    Kommentar
                                ; Lineare Interpolation ist voreingestellt:
                                ; MD20730 $MC_GO_LINEAR_MODE == TRUE
...
N30 RTLIOF                      ; Lineare Interpolation ausschalten.
N40 G0 X0 Y10                   ; G0-Sätze werden mit nicht-linearer Interpolation verfahren.
N50 G41 X20 Y20                 ; WRK aktiv ⇒ G0-Sätze werden mit linearer Interpolation verfahren.
N60 G40 X30 Y30                 ; WRK nicht aktiv ⇒ G0-Sätze werden mit nicht-linearer Interpolation verfahren.
N70 RTLION                      ; Lineare Interpolation einschalten.
...
```

## Weitere Informationen

### Lesen des aktuellen Interpolationsverhaltens

Über die Systemvariablen $AA_G0MODE kann das aktuelle Interpolationsverhalten gelesen werden.

### 2.9.4.3 Toleranzen für Eilgangbewegungen anpassen (STOLF, CTOLG0, OTOLG0)

Die über Maschinendaten projektierten Toleranzen für Eilgangbewegungen (G0-Toleranzen) können im Teileprogramm temporär angepasst werden. Die Einstellungen in den Maschinendaten werden dabei nicht verändert. Nach Kanal- bzw. Programmende-Reset werden wieder die projektierten Toleranzen wirksam.

## Voraussetzungen

G0-Toleranzen werden nur wirksam, wenn folgende Bedingungen erfüllt sind:

- Eine der folgenden Funktionen ist aktiv:
  - Kompressorfunktion COMP...
  - Überschleiffunktion G642 oder G645
  - Orientierungsüberschleifen OST
  - Orientierungsglättung ORISON
  - Glättung bei bahnrelativer Orientierung ORIPATH
- Im Teileprogramm folgen mehrere (≥ 2) G0-Sätze aufeinander.
  Bei einem einzelnen G0-Satz werden G0-Toleranzen nicht wirksam, da beim Übergang von einer Nicht-G0-Bewegung zu einer G0-Bewegung (und umgekehrt) grundsätzlich die "**kleinere Toleranz**" (Werkstückbearbeitungstoleranz) wirkt!

## Syntax

**Anpassung der relativen G0-Toleranz**

```
STOLF=<Value>
```

**Anpassung der absoluten G0-Toleranzen**

```
CTOLG0=<Value>
OTOLG0=<Value>
```

## Bedeutung

| | | | | |
|---|---|---|---|---|
| STOLF: | Adresse zum Programmieren eines temporär wirksamen Toleranzfaktors für Eilgangbewegungen | | | |
| | <Value>: | G0-Toleranzfaktor | | |
| | | Typ: | REAL | |
| | | Wert: | ≥ 0: | Der G0-Toleranzfaktor kann sowohl größer als auch kleiner 1.0 sein. Ist der Faktor gleich 1.0 (Standardwert), sind für Eilgangbewegungen dieselben Toleranzen wirksam wie für Nicht-Eilgangbewegungen. Im Normalfall wird der Toleranzfaktor > 1.0 eingestellt.<br>Der programmierte G0-Toleranzfaktor ist so lange wirksam, bis er durch eine erneute STOLF-Programmierung überschrieben bzw. durch eine CTOLG0/OTOLG0-Programmierung abgelöst oder durch Kanal- bzw. Programmende-Reset gelöscht wird. |
| | | | < 0: | Löschen des programmierten Toleranzfaktors<br>⇒ Es gilt wieder der im Maschinendatum voreingestellte Toleranzfaktor. |
| CTOLG0: | Adresse zum Programmieren einer temporär wirksamen Konturtoleranz für Eilgangbewegungen | | | |
| | <Value>: | Absolutwert für die Konturtoleranz | | |
| | | Typ: | REAL | |
| | | Wert: | ≥ 0: | Der programmierte Absolutwert für die Konturtoleranz ist so lange wirksam, bis er durch eine erneute CTOLG0-Programmierung überschrieben bzw. durch eine STOLF-Programmierung abgelöst oder durch Kanal- bzw. Programmende-Reset gelöscht wird. |
| | | | < 0: | Löschen des programmierten Toleranzwerts<br>⇒ Es gilt wieder der im Maschinendatum voreingestellte Toleranzwert. |
| OTOLG0: | Adresse zum Programmieren einer temporär wirksamen Orientierungstoleranz für Eilgangbewegungen | | | |
| | <Value>: | Absolutwert für die Orientierungstoleranz | | |
| | | Typ: | REAL | |
| | | Wert: | ≥ 0: | Der programmierte Absolutwert für die Orientierungstoleranz ist so lange wirksam, bis er durch eine erneute OTOLG0-Programmierung überschrieben bzw. durch eine STOLF-Programmierung abgelöst oder durch Kanal- bzw. Programmende-Reset gelöscht wird. |
| | | | < 0: | Löschen des programmierten Toleranzwerts<br>⇒ Es gilt wieder der im Maschinendatum voreingestellte Toleranzwert. |

**Hinweis**

Vorrang hat immer die zuletzt programmierte Adresse, wie die folgenden Beispiele zeigen:

- Bei Programmierung von CTOLG0 bei bestehendem STOLF wird für das Überschleifen der Kontur der mit CTOLG0 programmierte Toleranzwert verwendet.
- Ebenso wird bei Programmierung von OTOLG0 bei bestehendem STOLF für das Überschleifen der Orientierung der mit OTOLG0 programmierte Toleranzwert verwendet.
- Nach erneuter Programmierung von STOLF wird wieder der Toleranzfaktor für die Kontur- und Orientierungstoleranz verwendet.

## Beispiele

### Beispiel 1: Anpassung der relativen G0-Toleranz

| Programmcode | Kommentar |
|---|---|
| **COMPCAD** G645 G1 F10000 | ; Kompressor-Funktion COMPCAD |
| X... Y... Z... | ; Hier wirken die Maschinen- und Settingdaten. |
| X... Y... Z... | |
| X... Y... Z... | |
| G0 X... Y... Z... | |
| G0 X... Y... Z... | ; Hier wirkt das Maschinendatum $MC_G0_TOLERANCE_FACTOR (z.B. =3), also eine Überschleiftoleranz von: $MC_G0_TOLERANCE_FACTOR * $MA_COMPRESS_POS_TOL |
| CTOL=0.02 | |
| **STOLF=4** | |
| G1 X... Y... Z... | ; Ab hier wirkt eine Konturtoleranz von 0,02 mm. |
| X... Y... Z... | |
| X... Y... Z... | |
| G0 X... Y... Z... | |
| X... Y... Z... | ; Ab hier wirkt ein G0-Toleranzfaktor von 4, also eine Konturtoleranz von 0,08 mm. |
| ... | |

### Beispiel 2: Anpassung der absoluten G0-Toleranzen

In den Maschinendaten sollen folgende absolute G0-Toleranzen voreingestellt sein:

- G0-Konturtoleranz: 0.1
- G0-Orientierungstoleranz: 1.0

Im Teileprogramm sollen diese Toleranzen temporär angepasst werden:

| Programmcode | Kommentar |
|---|---|
| **COMPCAD** G645 G1 F10000 | ; Kompressor-Funktion COMPCAD |
| X... Y... Z... | ; Ab hier wirken die projektierten Werkstückbearbeitungstoleranzen. |
| X... Y... Z... | |

| Programmcode | Kommentar |
|---|---|
| X... Y... Z... | |
| G0 X... Y... Z... | |
| G0 X... Y... Z... | ; Hier wirken die projektierten absoluten G0-Toleranzen. |
| **CTOLG0=0.2 OTOLG0=2.0** | ; Programmierung der absoluten G0-Toleranzen. |
| G1 X... Y... Z... | |
| X... Y... Z... | |
| X... Y... Z... | |
| G0 X... Y... Z... | |
| X... Y... Z... | ; Ab hier wirken die programmierten G0-Toleranzen. |
| ... | |

## Weitere Informationen

### Lesen des G0-Toleranzfaktors

Der aktuell wirksame Toleranzfaktor für Eilgangbewegungen ist über Systemvariablen lesbar:

- In Synchronaktionen oder mit Vorlauf-Stopp im Teileprogramm über die Systemvariable:
  $AC_STOLF — Aktiver G0-Toleranzfaktor
  G0-Toleranzfaktor, der bei der Aufbereitung des aktuellen Hauptlaufsatzes wirksam war.

- Ohne Vorlauf-Stopp im Teileprogramm über die Systemvariable:
  $P_STOLF — Programmierter G0-Toleranzfaktor

Ist im aktiven Teileprogramm kein Wert mit STOLF programmiert, dann liefern diese beiden Systemvariablen den im Maschinendatum projektierten Wert.

Ist in einem Satz kein Eilgang (G0) aktiv, dann liefern diese Systemvariablen immer den Wert 1.

### Lesen der absoluten G0-Toleranzen

Die aktuell wirksamen absoluten Toleranzen für Eilgangbewegungen sind über Systemvariablen lesbar:

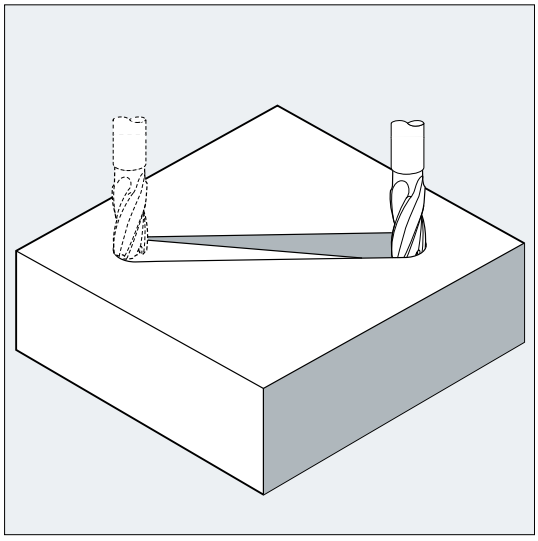
- In Synchronaktionen oder mit Vorlauf-Stopp im Teileprogramm über die Systemvariablen:
  $AC_CTOL_G0_ABS — Aktive Konturtoleranz bei G0-Bewegungen
  G0-Konturtoleranz, die bei der Aufbereitung des aktuellen Hauptlaufsatzes wirksam war.
  $AC_OTOL_ G0_ABS — Aktive Orientierungstoleranz bei G0-Bewegungen
  G0-Orientierungstoleranz, die bei der Aufbereitung des aktuellen Hauptlaufsatzes wirksam war.

- Ohne Vorlauf-Stopp im Teileprogramm über die Systemvariablen:
  $P_CTOL_ G0_ABS — Programmierte Konturtoleranz bei G0-Bewegungen
  $P_OTOL_ G0_ABS — Programmierte Orientierungstoleranz bei G0-Bewegungen

Sind im aktiven Teileprogramm keine absoluten G0-Toleranzen mit CTOLG0 und OTOLG0 programmiert, dann liefern diese Systemvariablen die in den Maschinendaten projektierten Werte.

## 2.9.5 Geradeninterpolation (G1)

Mit `G1` fährt das Werkzeug auf achsparallelen, schräg liegenden oder beliebig im Raum liegenden Geraden. Die Geradeninterpolation ermöglicht die Herstellung von 3D-Flächen, Nuten uvm.



### Syntax

```
G1 X… Y… Z … F…
G1 AP=… RP=… F…
```

### Bedeutung

| | |
|---|---|
| `G1`: | Geradeninterpolation (Linearinterpolalation mit Vorschub) |
| `X... Y... Z...`: | Endpunkt in kartesischen Koordinaten |
| `AP=...`: | Endpunkt in Polarkoordinaten, hier Polarwinkel |
| `RP=...`: | Endpunkt in Polarkoordinaten, hier Polarradius |
| `F...`: | Vorschubgeschwindigkeit in mm/min. Das Werkzeug fährt mit Vorschub F auf einer Geraden vom aktuellen Startpunkt zum programmierten Zielpunkt. Den Zielpunkt geben Sie in kartesischen Koordinaten oder Polarkoordinaten ein. Auf dieser Bahn wird das Werkstück bearbeitet.<br>Beispiel: `G1 G94 X100 Y20 Z30 A40 F100`<br>Der Endpunkt in X, Y, Z wird mit Vorschub 100 mm/min angefahren, die Rundachse A wird als Synchronachse so verfahren, dass alle vier Bewegungen zeitgleich abgeschlossen werden. |

**Hinweis**

`G1` ist modal wirksam.

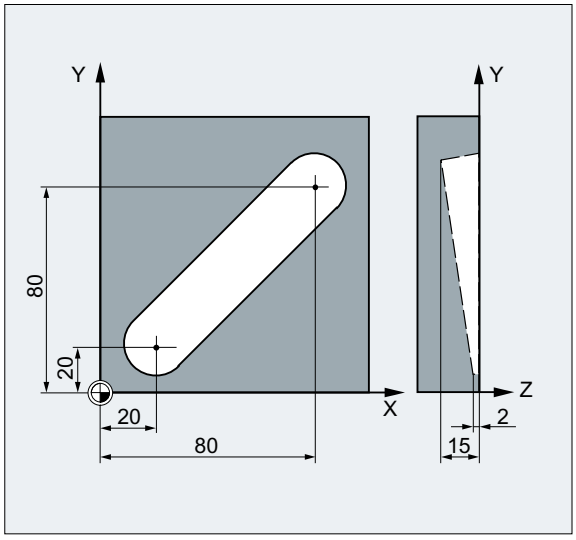
Für die Bearbeitung müssen Spindeldrehzahl `S` und Spindeldrehrichtung `M3`/`M4` angegeben werden.

Mit `FGROUP` können Achsgruppen festgelegt werden, für die Bahnvorschub `F` gilt. Mehr Informationen hierzu im Kapitel "Bahnverhalten".

## Beispiele

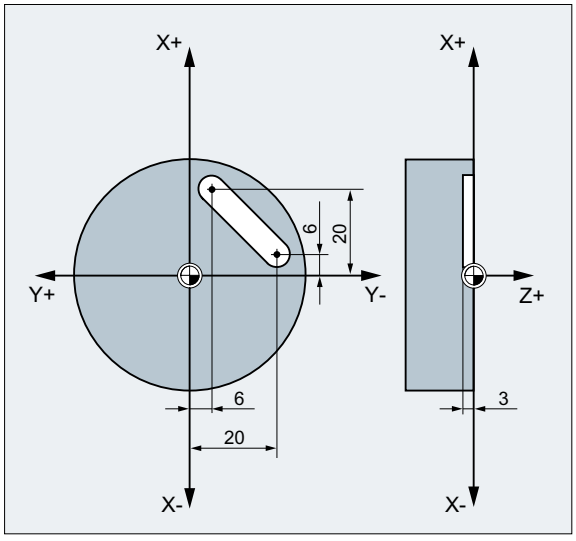
### Beispiel 1: Herstellung einer Nut (Fräsen)

Das Werkzeug fährt vom Start- zum Endpunkt in X/Y-Richtung. Gleichzeitig wird in Z-Richtung zugestellt.




| Programmcode | Kommentar |
|---|---|
| N10 G17 S400 M3 | ; Wahl der Arbeitsebene, Spindel rechts |
| N20 G0 X20 Y20 Z2 | ; Anfahren der Startposition |
| N30 G1 Z-2 F40 | ; Zustellen des Werkzeugs |
| N40 X80 Y80 Z-15 | ; Fahren auf einer schräg liegenden Geraden |
| N50 G0 Z100 M30 | ; Freifahren zum Werkzeugwechsel |

**Beispiel 2: Herstellung einer Nut (Drehen)**




| Programmcode | Kommentar |
|---|---|
| N10 G17 S400 M3 | ; Wahl der Arbeitsebene, Spindel rechts |
| N20 G0 X40 Y-6 Z2 | ; Anfahren der Startposition |
| N30 G1 Z-3 F40 | ; Zustellen des Werkzeugs |
| N40 X12 Y-20 | ; Fahren auf einer schräg liegenden Geraden |
| N50 G0 Z100 M30 | ; Freifahren zum Werkzeugwechsel |

## 2.9.6 Kreisinterpolation

### 2.9.6.1 Übersicht

Die Kreisinterpolation ermöglicht die Herstellung von Vollkreisen oder Kreisbögen.

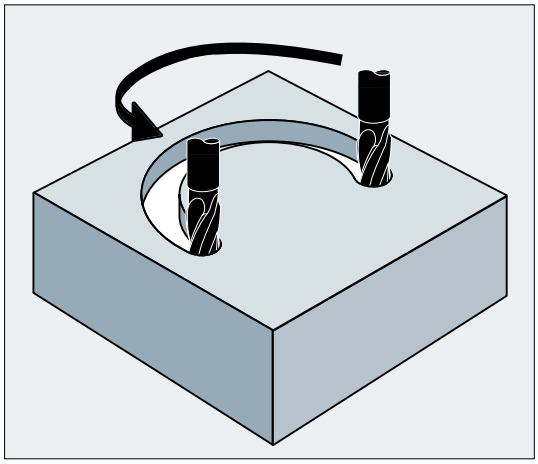


Bild 2-3 Anwendungsbeispiel: Fräsen einer kreisförmigen Nut

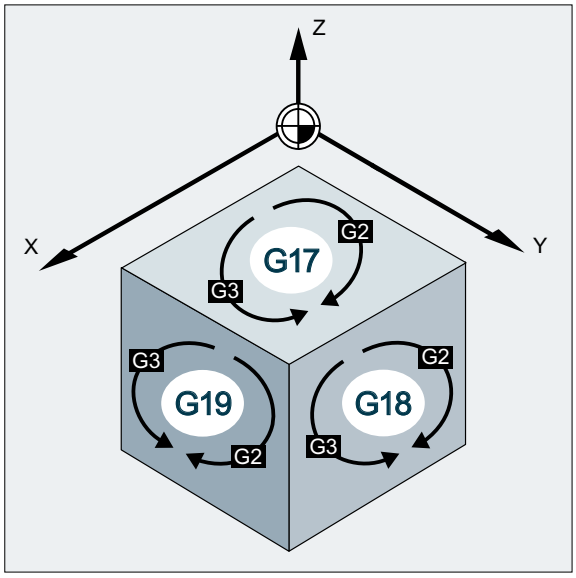
### Programmiervarianten

Die Steuerung bietet verschiedene Möglichkeiten, Kreisbewegungen zu programmieren. Damit kann der Anwender praktisch jede Art der Zeichnungsbemaßung direkt umsetzen:

- Kreisinterpolation mit Mittelpunkt und Endpunkt (G2/G3, X... Y... Z..., I... J... K...) (Seite 192)
- Kreisinterpolation mit Radius und Endpunkt (G2/G3, X... Y... Z..., CR) (Seite 195)
- Kreisinterpolation mit Öffnungswinkel und Endpunkt / Mittelpunkt (G2/G3, X... Y... Z... / I... J... K..., AR) (Seite 197)
- Kreisinterpolation mit Polarkoordinaten (G2/G3, AP, RP) (Seite 199)
- Kreisinterpolation mit Zwischen- und Endpunkt (CIP, X... Y... Z..., I1... J1... K1...) (Seite 200)
- Kreisinterpolation mit tangentialem Übergang (CT, X... Y... Z...) (Seite 203)

### Ebene für die Kreisinterpolation

Die Steuerung benötigt zur Berechnung des Kreisdrehsinns, mit `G2` im Uhrzeigersinn oder `G3` gegen den Uhrzeigersinn, die Angabe der Arbeitsebene (Seite 149).




Ausnahme:

Es ist auch möglich, außerhalb der gewählten Arbeitsebene Kreise herzustellen (nicht bei Öffnungswinkelangabe und Schraubenlinie). In diesem Fall bestimmen die Achsadressen, die der Programmierer als Kreisendpunkt angibt, die Kreisebene.

## 2.9.6.2 Kreisinterpolation mit Mittelpunkt und Endpunkt (G2/G3, X... Y... Z..., I... J... K...)

Kreisinterpolationsvariante, die den **Mittelpunkt** und **Endpunkt** eines kreisförmigen Konturelements für die Interpolation verwendet.

Wird der Kreis ohne Endpunkt programmiert, entsteht ein Vollkreis.

## Syntax

```
G2/G3 X… Y… Z… I… J… K…
G2/G3 X… Y… Z… I=AC(…) J=AC(…) K=(AC…)
```

## Bedeutung

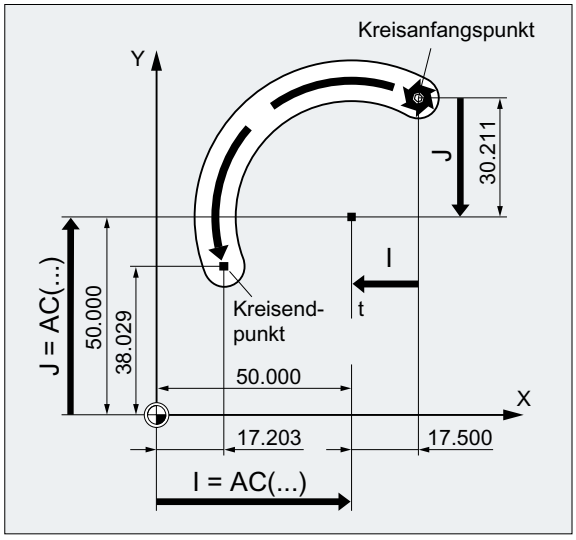
| | | |
|---|---|---|
| G2: | Kreisinterpolation im Uhrzeigersinn | |
| | Wirksamkeit: | modal |
| G3: | Kreisinterpolation im Gegenuhrzeigersinn | |
| | Wirksamkeit: | modal |
| X... Y... Z...: | Kreisendpunkt in kartesischen Koordinaten<br>Abhängig von der aktuell gültigen Maßangaben-Einstellung G90/G91 bzw. ...=AC(...) / ...=IC(...) werden die Kreisendpunkt-Koordinaten entweder im Absolutmaß oder im Kettenmaß interpretiert. | |
| I... J... K...: | Interpolationsparameter zur Angabe der Kreismittelpunkt-Koordinaten in Richtung X, Y, Z<br>Die Kreismittelpunkt-Koordinaten werden standardmäßig im Kettenmaß bezogen auf den Kreisanfangspunkt angegeben.<br>Sollen die Kreismittelpunkt-Koordinaten im Absolutmaß bezogen auf den Werkstücknullpunkt angegeben werden, sind die Interpolationsparameter I, J, K wie folgt zu programmieren:<br>I=AC(…) J=AC(…) K=AC(…)<br>**Hinweis**<br>Ein Interpolationsparameter mit Wert "0" kann entfallen, der zugehörige zweite Parameter muss in jedem Fall angegeben werden. | |

**Hinweis**

Die Voreinstellung G90/G91 Absolut- oder Kettenmaß ist nur für den Kreisendpunkt gültig.

## Beispiele

### Beispiel 1: Fräsen



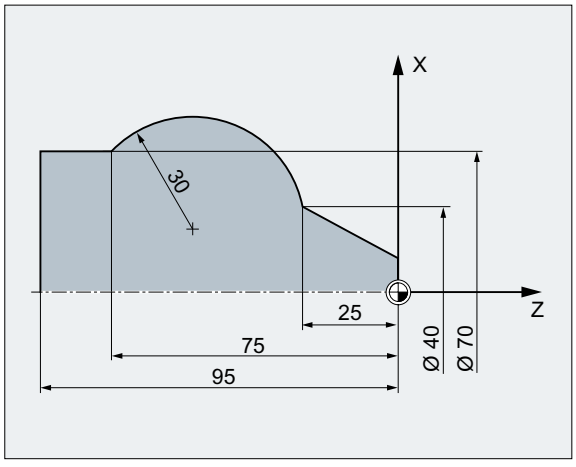

**Mittelpunktangabe im Kettenmaß**

```
N10 G0 X67.5 Y80.211
N20 G3 X17.203 Y38.029 I-17.5 J-30.211 F500
```

**Mittelpunktangabe im Absolutmaß**

```
N10 G0 X67.5 Y80.211
N20 G3 X17.203 Y38.029 I=AC(50) J=AC(50)
```

### Beispiel 2: Drehen




**Mittelpunktangabe im Kettenmaß**

```
N120 G0 X12 Z0
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 I-3.335 K-29.25
N135 G1 Z-95
```

**Mittelpunktangabe im Absolutmaß**

```
N120 G0 X12 Z0
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 I=AC(33.33) K=AC(-54.25)
N135 G1 Z-95
```

### 2.9.6.3 Kreisinterpolation mit Radius und Endpunkt (G2/G3, X... Y... Z..., CR)

Kreisinterpolationsvariante, die den **Radius** und **Endpunkt** eines kreisförmigen Konturelements für die Interpolation verwendet.

**Hinweis**

Vollkreise (Verfahrwinkel 360°) können mit dieser Variante **nicht** programmiert werden.

## Syntax

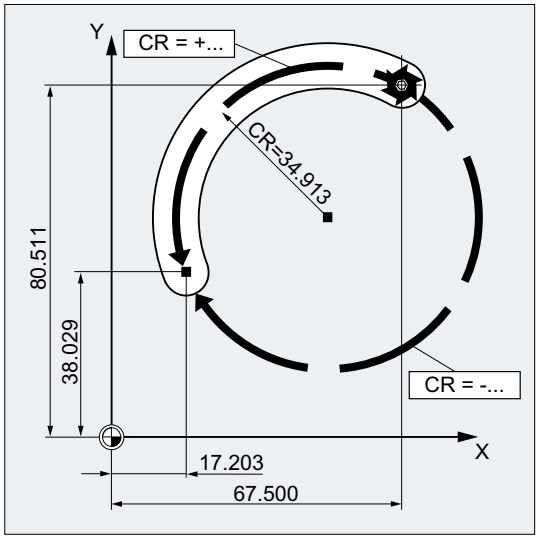
```
G2/G3 X… Y… Z… CR=±...
```

## Bedeutung

| | | |
|---|---|---|
| G2: | Kreisinterpolation im Uhrzeigersinn | |
| | Wirksamkeit: | modal |
| G3: | Kreisinterpolation im Gegenuhrzeigersinn | |
| | Wirksamkeit: | modal |
| X... Y... Z...: | Kreisendpunkt in kartesischen Koordinaten<br>Abhängig von der aktuell gültigen Maßangaben-Einstellung G90/G91 bzw. ...=AC(...) / ...=IC(...) werden die Endpunktkoordinaten entweder im Absolutmaß oder im Kettenmaß interpretiert. | |
| CR=±...: | Kreisradius<br>Durch das Vorzeichen wird angeben, ob der Verfahrwinkel größer oder kleiner 180° sein soll. Ein positives Vorzeichen kann entfallen. | |
| | CR=+...: | Verfahrwinkel ≤ 180° |
| | CR=-...: | Verfahrwinkel > 180° |
| | **Hinweis**<br>Es gibt keine praxisrelevante Beschränkung für die Größe des maximal programmierbaren Radius. | |

## Beispiele

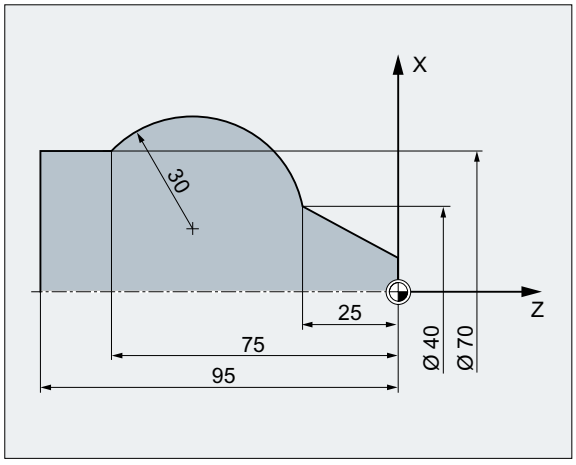
### Beispiel 1: Fräsen




**Programmcode**

```
N10 G0 X67.5 Y80.511
N20 G3 X17.203 Y38.029 CR=34.913 F500
...
```

### Beispiel 2: Drehen




**Programmcode**

```
...
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 CR=30
N135 G1 Z-95
...
```

### 2.9.6.4 Kreisinterpolation mit Öffnungswinkel und Endpunkt / Mittelpunkt (G2/G3, X... Y... Z... / I... J... K..., AR)

Kreisinterpolationsvariante, die den **Öffnungswinkel** und den **Mittelpunkt** oder **Endpunkt** eines kreisförmigen Konturelements für die Interpolation verwendet.

**Hinweis**

Vollkreise (Verfahrwinkel 360°) können mit dieser Variante **nicht** programmiert werden.

## Syntax

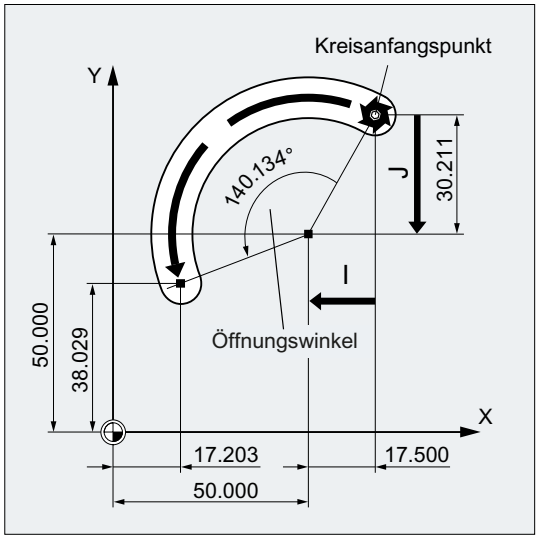
```
G2/G3 X… Y… Z… AR=...
G2/G3 I… J… K… AR=...
```

## Bedeutung

| | | |
|---|---|---|
| G2: | Kreisinterpolation im Uhrzeigersinn | |
| | Wirksamkeit: | modal |
| G3: | Kreisinterpolation im Gegenuhrzeigersinn | |
| | Wirksamkeit: | modal |
| X... Y... Z...: | Kreisendpunkt in kartesischen Koordinaten<br>Abhängig von der aktuell gültigen Maßangaben-Einstellung G90/G91 bzw. ...=AC(...) / ...=IC(...) werden die Kreisendpunkt-Koordinaten entweder im Absolutmaß oder im Kettenmaß interpretiert. | |
| I... J... K...: | Interpolationsparameter zur Angabe der Kreismittelpunkt-Koordinaten in Richtung X, Y, Z<br>Die Kreismittelpunkt-Koordinaten werden standardmäßig im Kettenmaß bezogen auf den Kreisanfangspunkt angegeben.<br>Sollen die Kreismittelpunkt-Koordinaten im Absolutmaß bezogen auf den Werkstücknullpunkt angegeben werden, sind die Interpolationsparameter I, J, K wie folgt zu programmieren:<br>I=AC(…) J=AC(…) K=AC(…)<br>**Hinweis**<br>Ein Interpolationsparameter mit Wert "0" kann entfallen, der zugehörige zweite Parameter muss in jedem Fall angegeben werden. | |
| AR=...: | Öffnungswinkel | |
| | Wertebereich: | 0° ... 360° |

## Beispiele

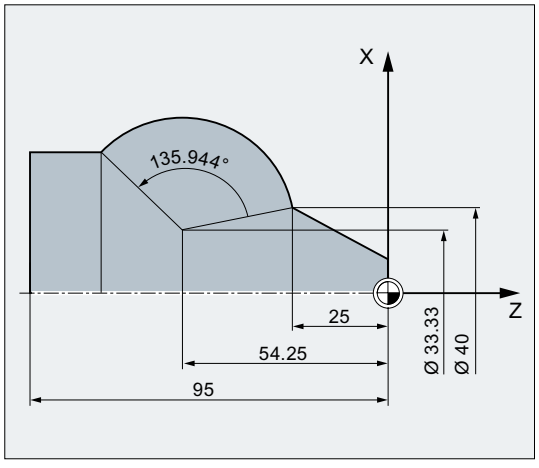
### Beispiel 1: Fräsen




**Programmcode**

```
N10 G0 X67.5 Y80.211
N20 G3 X17.203 Y38.029 AR=140.134 F500
N20 G3 I-17.5 J-30.211 AR=140.134 F500
```

### Beispiel 2: Drehen




**Programmcode**

```
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 AR=135.944
N130 G3 I-3.335 K-29.25 AR=135.944
N130 G3 I=AC(33.33) K=AC(-54.25) AR=135.944
```

| **Programmcode** |
|---|
| `N135 G1 Z-95` |

### 2.9.6.5 Kreisinterpolation mit Polarkoordinaten (G2/G3, AP, RP)

Kreisinterpolationsvariante, die den **Kreisendpunkt in Polarkoordinaten** für die Interpolation verwendet.

Hierbei gilt folgende Vereinbarung:

- Der Pol liegt im Kreismittelpunkt.
- Der Polarradius entspricht dem Kreisradius.

## Syntax

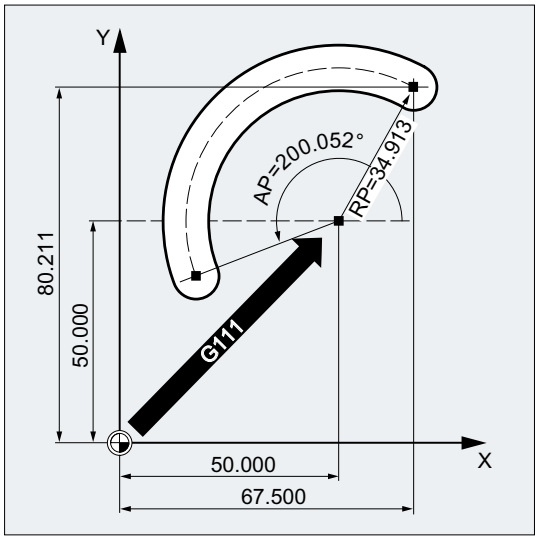
```
G2/G3 AP=... RP=...
```

## Bedeutung

| | | |
|---|---|---|
| `G2`: | Kreisinterpolation im Uhrzeigersinn | |
| | Wirksamkeit: | modal |
| `G3`: | Kreisinterpolation im Gegenuhrzeigersinn | |
| | Wirksamkeit: | modal |
| `AP=... RP=...`: | Kreisendpunkt in Polarkoordinaten | |
| | `AP=...`: | Polarwinkel |
| | `RP=...`: | Polarradius (≙ Kreisradius) |

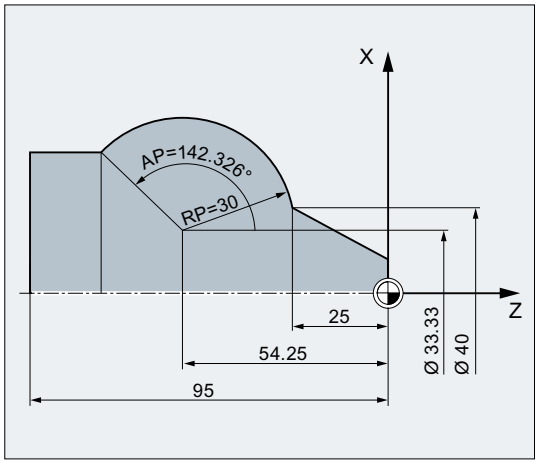
## Beispiele

**Beispiel 1: Fräsen**



| **Programmcode** |
|---|
| N10 G0 X67.5 Y80.211 |
| N20 G111 X50 Y50 |
| N30 G3 RP=34.913 AP=200.052 F500 |

**Beispiel 2: Drehen**



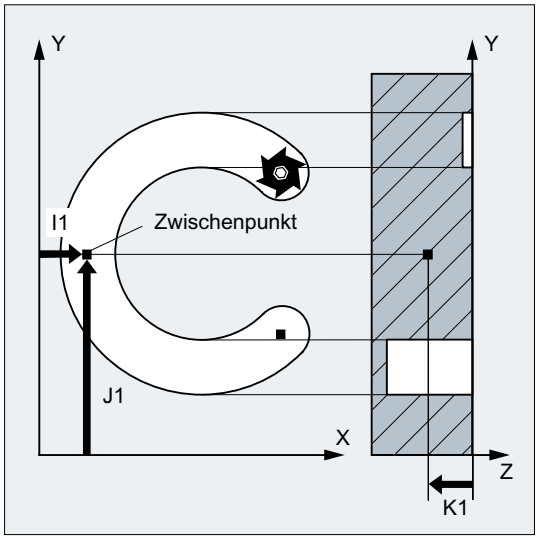

| **Programmcode** |
|---|
| N125 G1 X40 Z-25 F0.2 |
| N130 G111 X33.33 Z-54.25 |
| N135 G3 RP=30 AP=142.326 |
| N140 G1 Z-95 |

### 2.9.6.6 Kreisinterpolation mit Zwischen- und Endpunkt (CIP, X... Y... Z..., I1... J1... K1...)

Die mit dem G-Befehl `CIP` programmierte Kreisinterpolationsvariante ermöglicht die Interpolation von schräg im Raum liegenden Kreisbögen.

Der Kreisbewegung wird durch den **Zwischenpunkt** und den **Endpunkt** der kreisförmigen Kontur beschrieben.

Die Verfahrrichtung ergibt sich aus der Reihenfolge Anfangspunkt → Zwischenpunkt → Endpunkt.




## Syntax

```
CIP X… Y… Z… I1=AC(…) J1=AC(…) K1=(AC…)
```

## Bedeutung

| | | |
|---|---|---|
| `CIP`: | Kreisinterpolation über Zwischenpunkt | |
| | Wirksamkeit: | modal |
| `X... Y... Z...`: | Kreisendpunkt in kartesischen Koordinaten<br>Abhängig von der aktuell gültigen Maßangaben-Einstellung `G90`/`G91` bzw. `...=AC(...)` / `...=IC(...)` werden die Kreisendpunkt-Koordinaten entweder im Absolutmaß oder im Kettenmaß interpretiert. | |
| `I1... J1... K1...`: | Interpolationsparameter zur Angabe der Kreiszwischenpunkt-Koordinaten in Richtung X, Y, Z<br>Abhängig von der aktuell gültigen Maßangaben-Einstellung `G90`/`G91` bzw. `...=AC(...)` / `...=IC(...)` werden die Kreiszwischenpunkt-Koordinaten entweder im Absolutmaß oder im Kettenmaß interpretiert.<br>**Hinweis**<br>Ein Interpolationsparameter mit Wert "0" kann entfallen, der zugehörige zweite Parameter muss in jedem Fall angegeben werden. | |

### Hinweis

Die Voreinstellungen `G90`/`G91` (Absolut- oder Kettenmaßangabe) sind für Kreiszwischenpunkt und Kreisendpunkt gültig.

Bei aktiver Kettenmaßangabe `G91` bzw. `...=IC(...)` gilt für Zwischen- und Endpunkt der Kreisanfangspunkt als Bezug.

**Hinweis**

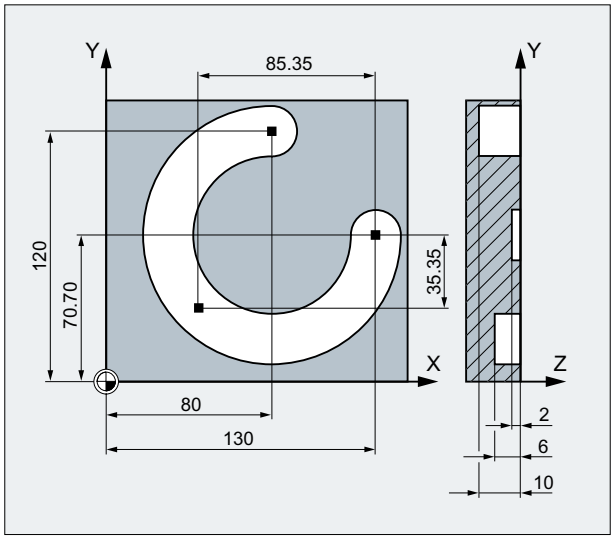
**Technologie Drehen**

Die Durchmesserprogrammierung des Interpolationsparameters für die Planachse wird bei der Kreisprogrammierung mit `CIP` nicht unterstützt. Der Interpolationsparameter für die Planachse ist daher im **Radius** zu programmieren.

## Beispiele

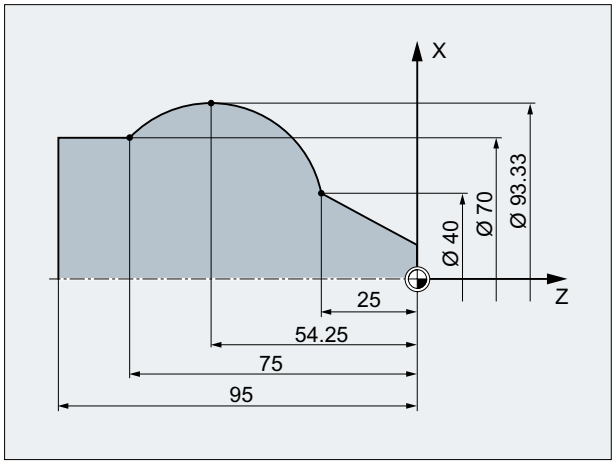
### Beispiel 1: Fräsen

Für die Herstellung einer schräg im Raum liegenden Kreisnut wird ein Kreis über Zwischenpunktangabe mit 3 Interpolationsparametern und Endpunkt mit ebenfalls 3 Koordinaten beschrieben.




| **Programmcode** | **Kommentar** |
|---|---|
| `N10 G0 G90 X130 Y70.70 S800 M3` | `; Startpunkt anfahren.` |
| `N20 G17 G1 Z-2 F100` | `; Zustellen des Werkzeugs.` |
| `N30 CIP X80 Y120 Z-10 I1=IC(-85.35) J1=IC(-35.35) K1=-6` | `; Kreisendpunkt und Zwischenpunkt.` `; Koordinaten für alle 3 Geometrieachsen.` |
| `N40 M30` | `; Programmende.` |

**Beispiel 2: Drehen**




| Programmcode | Kommentar |
|---|---|
| ... | |
| N125 G1 G90 X40 Z-25 F0.2 | |
| N130 CIP X70 Z-75 **I1=IC(26.665)** K1=IC(-29.25) | ; Interpolationsparameter I1 für Planachse muss im Radius programmiert sein. |
| ; bzw. | |
| ; N130 CIP X70 Z-75 **I1=46.665** K1=-54.25 | |
| N135 G1 Z-95 | |

## 2.9.6.7 Kreisinterpolation mit tangentialem Übergang (CT, X... Y... Z...)

Die mit dem G-Befehl `CT` programmierte Kreisinterpolationsvariante ermöglicht die Interpolation von Kreisbögen, die tangential an das zuvor programmierte Konturelement anschließen.

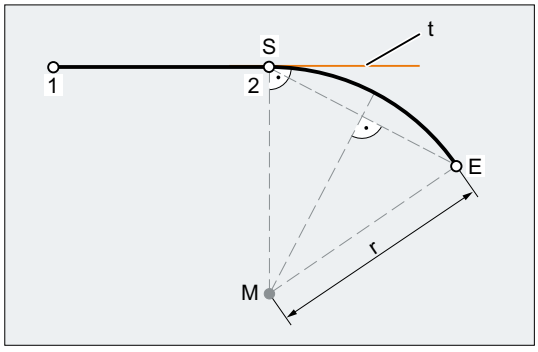
Der Kreis wird dabei durch den **Start- und Endpunkt** und die **Tangentenrichtung im Startpunkt** definiert.

**Hinweis**

**Tangentenrichtung im Startpunkt**

Die Tangentenrichtung im Startpunkt eines CT-Satzes wird aus der Endtangente der programmierten Kontur des letzten Vorgängersatzes mit einer Verfahrbewegung bestimmt.

Zwischen diesem Satz und dem aktuellen Satz können beliebig viele Sätze ohne Verfahrinformation liegen.




S Startpunkt
E Endpunkt
M Kreismittelpunkt
r Kreisradius
t Endtangente der programmierten Kontur des letzten Vorgängersatzes mit einer Verfahrbewegung

Bild 2-4 Tangential an das Geradenstück 1-2 anschließende Kreisbahn S-E

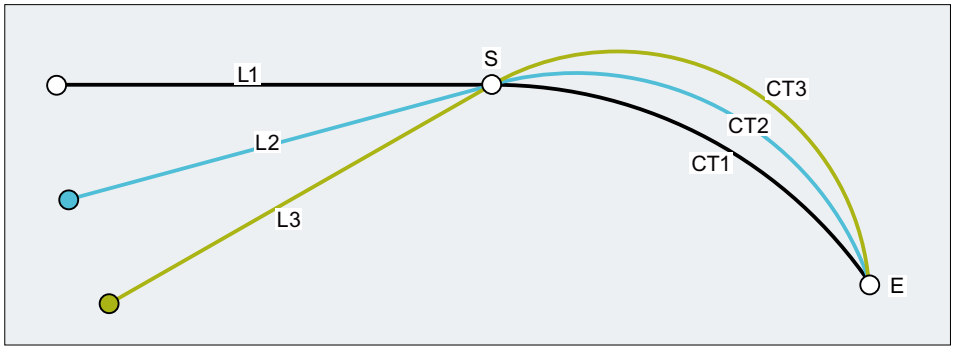



Bild 2-5 Tangential anschließende Kreisbahnen hängen vom vorhergehenden Konturelement ab

## Syntax

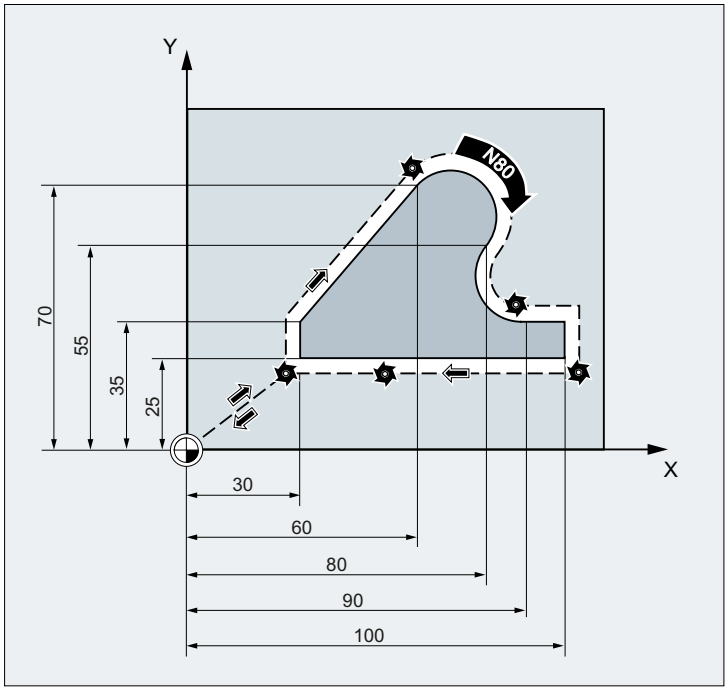
```
CT X... Y... Z...
```

## Bedeutung

| | | |
|---|---|---|
| `CT`: | Kreisinterpolation mit tangentialem Übergang | |
| | Wirksamkeit: | modal |
| `X... Y... Z...`: | Kreisendpunkt in kartesischen Koordinaten<br>Abhängig von der aktuell gültigen Maßangaben-Einstellung `G90`/`G91` bzw. `...=AC(...)` / `...=IC(...)` werden die Kreisendpunkt-Koordinaten entweder im Absolutmaß oder im Kettenmaß interpretiert. | |

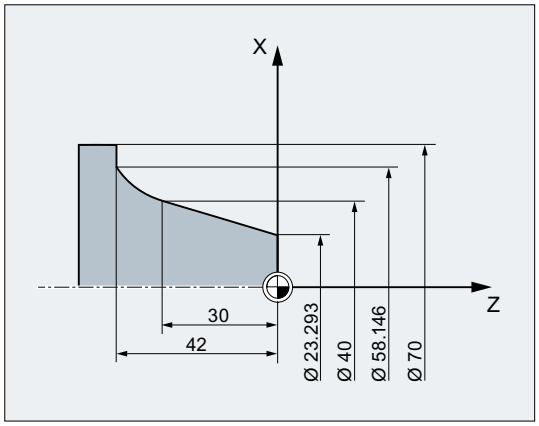
## Beispiele

### Beispiel 1: Fräsen




| Programmcode | Kommentar |
|---|---|
| N10 G0 Z100 | |
| N20 G17 T1 M6 | |
| N30 G0 X0 Y0 Z2 M3 S300 D1 | |
| N40 Z-5 F1000 | ; Werkzeug zustellen. |
| N50 G41 X30 Y25 G1 F1000 | ; Einschalten der Werkzeugradiuskorrektur. |
| N60 Y35 | ; Kontur fräsen. |
| N70 X60 Y70 | |
| **N80 CT X80 Y55** | **; Kreisprogrammierung mit tangentialem Übergang.** |
| **N90 X90 Y35** | |
| N100 G1 X100 | |
| N110 Y25 | |
| N120 X30 | |
| N130 G0 G40 X0 Y0 | ; Ausschalten der Werkzeugradiuskorrektur. |
| N140 Z100 | ; Werkzeug freifahren. |
| N140 M30 | |

### Beispiel 2: Drehen




| Programmcode | Kommentar |
|---|---|
| ... | |
| N110 G1 X23.293 Z0 F10 | |
| N115 X40 Z-30 F0.2 | |
| **N120 CT X58.146 Z-42** | **; Kreisprogrammierung mit tangentialem Übergang.** |
| N125 G1 X70 | |
| ... | |

## Weitere Informationen

### Splines

Bei Splines wird die Tangentialrichtung durch die Gerade durch die letzten beiden Punkte bestimmt. Diese Richtung ist bei A- und C-Splines bei aktivem ENAT oder EAUTO im Allgemeinen nicht mit der Richtung im Endpunkt des Splines identisch.

Der Übergang von B-Splines ist immer tangential, wobei die Tangentenrichtung wie bei A- oder C-Splines und aktivem ETAN definiert ist.

### Framewechsel

Findet zwischen dem Satz, der die Tangente definiert, und dem CT-Satz ein Framewechsel statt, so wird die Tangente diesem Wechsel unterworfen.

### Grenzfall

Verläuft die Verlängerung der Starttangente durch den Endpunkt, wird statt eines Kreises eine Gerade erzeugt (Grenzfall eines Kreises mit unendlichem Radius). In diesem Spezialfall darf TURN entweder nicht programmiert sein oder es muss TURN=0 gelten.

### Hinweis

Bei der Annäherung an diesen Grenzfall entstehen Kreise mit beliebig großem Radius, sodass bei TURN ungleich 0 die Bearbeitung in der Regel mit einem Alarm wegen der Verletzung der Softwarelimits abgebrochen werden wird.

### Lage der Kreisebene

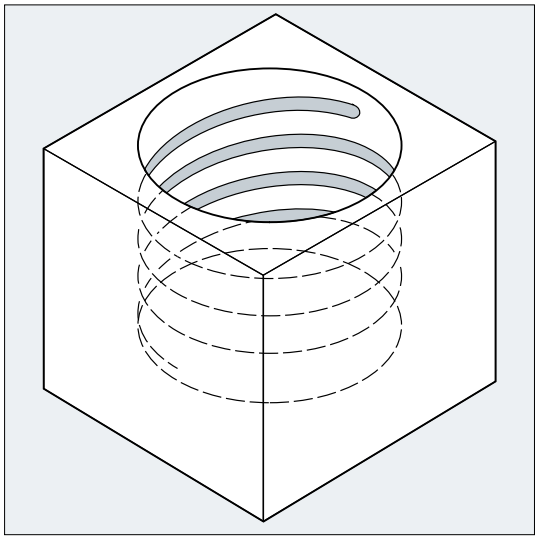
Die Lage der Kreisebene ist von der aktiven Ebene (G17-G19) abhängig.

Liegt die Tangente des Vorgängersatzes nicht in der aktiven Ebene, so wird deren Projektion in die aktive Ebene verwendet.

Haben Start- und Endpunkt nicht die gleiche Positionskomponente senkrecht zur aktiven Ebene, wird statt eines Kreises eine Helix erzeugt.

## 2.9.7 Schraubenlinien-Interpolation (G2/G3, TURN)

Die Schraubenlinieninterpolation (Helixinterpolation) ermöglicht zum Beispiel die Herstellung von Gewinden oder Schmiernuten.



Bei der Schraubenlinieninterpolation werden zwei Bewegungen überlagert und parallel ausgeführt:

- eine ebene Kreisbewegung, der
- eine senkrechte Linearbewegung überlagert wird.

### Syntax

```
G2/G3 X… Y… Z… I… J… K… TURN=
G2/G3 X… Y… Z… I… J… K… TURN=
G2/G3 AR=… I… J… K… TURN=
G2/G3 AR=… X… Y… Z… TURN=
G2/G3 AP… RP=… TURN=
```

### Bedeutung

| | |
|---|---|
| `G2`: | Fahren auf einer Kreisbahn im Uhrzeigersinn |
| `G3`: | Fahren auf einer Kreisbahn gegen den Uhrzeigersinn |
| `X Y Z`: | Endpunkt in kartesischen Koordinaten |

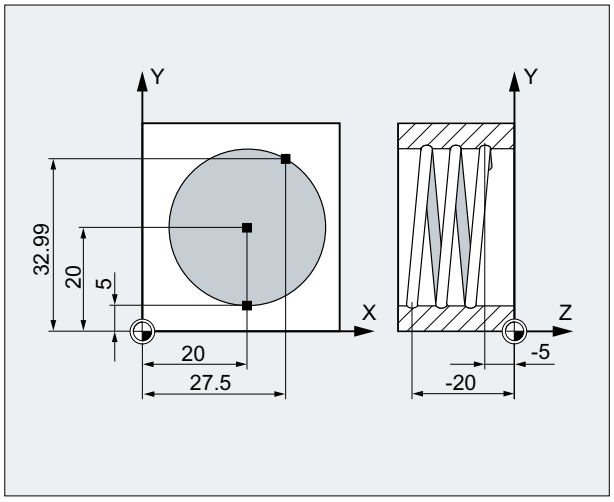
| | |
|---|---|
| I J K: | Kreismittelpunkt in kartesischen Koordinaten |
| AR: | Öffnungswinkel |
| TURN=: | Anzahl der zusätzlichen Kreisdurchläufe im Bereich von 0 bis 999 |
| AP=: | Polarwinkel |
| RP=: | Polarradius |

### Hinweis

G2 und G3 sind modal wirksam.

Die Kreisbewegung wird in den Achsen ausgeführt, die durch die Angabe der Arbeitsebene festgelegt sind.

## Beispiel




| Programmcode | Kommentar |
|---|---|
| N10 G17 G0 X27.5 Y32.99 Z3 | ; Anfahren der Startposition. |
| N20 G1 Z-5 F50 | ; Zustellen des Werkzeugs. |
| N30 G3 X20 Y5 Z-20 I=AC(20) J=AC(20) TURN=2 | ; Schraubenlinie mit den Angaben: Ab Startposition 2 Vollkreise ausführen, dann Endpunkt anfahren. |
| N40 M30 | ; Programmende. |

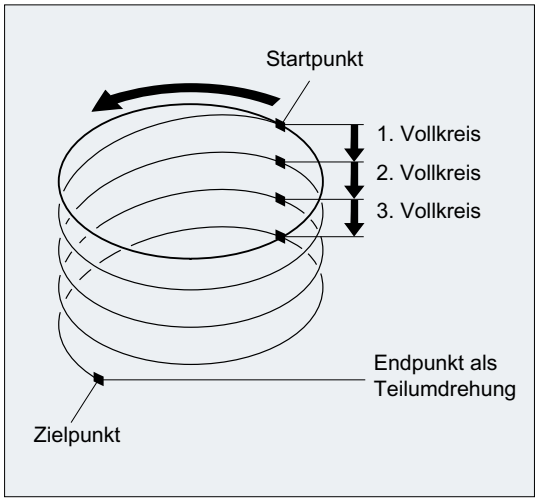
## Weitere Informationen

### Bewegungsfolge

1. Startpunkt anfahren
2. Mit TURN= programmierte Vollkreise ausführen.

3. Kreisendpunkt anfahren, z. B. als Teilumdrehung.
4. Punkt 2 und 3 über die Zustelltiefe ausführen.

Aus der Anzahl der Vollkreise plus programmierten Kreisendpunkt (ausgeführt über der Zustelltiefe), ergibt sich die Steigung, mit der die Schraubenlinie gefertigt werden soll.




#### Programmierung des Endpunkts Schraubenlinieninterpolation

Für detaillierte Erklärungen der Interpolationsparameter siehe Kreisinterpolation.

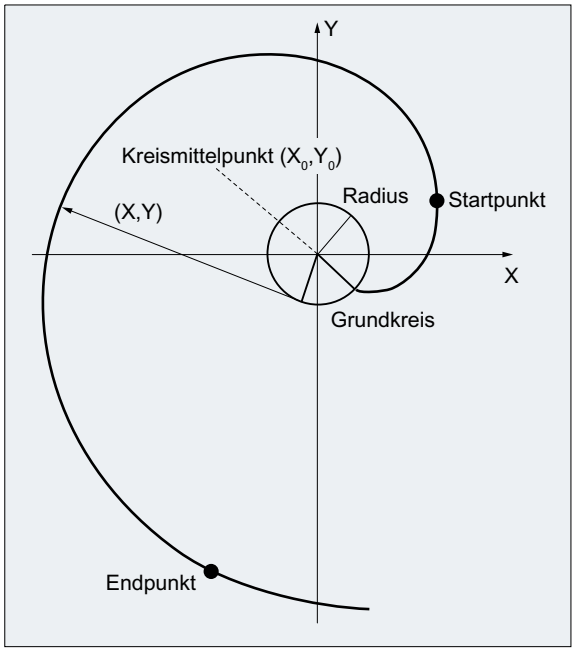
#### Programmierter Vorschub

Bei der Schraubenlinieninterpolation empfiehlt sich die Angabe eines programmierten Vorschub-Overrides (`CFC`). Mit `FGROUP` kann festgelegt werden, welche Achsen mit programmiertem Vorschub verfahren sollen. Mehr Informationen siehe Kapitel Bahnverhalten.

## 2.9.8 Evolventen-Interpolation (INVCW, INVCCW)

Die Evolvente des Kreises ist eine Kurve, die vom Endpunkt eines fest gespannten, von einem Kreis abgewickelten Fadens beschrieben wird.

Die Evolventen-Interpolation ermöglicht Bahnkurven entlang einer Evolvente. Sie wird in der Ebene ausgeführt, in welcher der Grundkreis definiert ist und verläuft vom programmierten Startpunkt zum programmierten Endpunkt.




Die Programmierung des Endpunkts kann auf zwei Arten erfolgen:

1. Direkt über kartesische Koordinaten
2. Indirekt durch Angabe eines Öffnungswinkels (vgl. hierzu auch die Programmierung des Öffnungswinkels bei der Kreisprogrammierung)

Liegen Start und Endpunkt nicht in der Ebene des Grundkreises, ergibt sich analog zur Schraubenlinien-Interpolation bei Kreisen eine Überlagerung zu einer Kurve im Raum.

Bei zusätzlicher Vorgabe von Bahnwegen senkrecht zur aktiven Ebene kann (vergleichbar zur Schraubenlinien-Interpolation bei Kreisen) eine Evolvente im Raum verfahren werden.

## Syntax

```
INVCW X... Y... Z... I... J... K... CR=...
INVCCW X... Y... Z... I... J... K... CR=...
INVCW I... J... K... CR=... AR=...
INVCCW I... J... K... CR=... AR=...
```

## Bedeutung

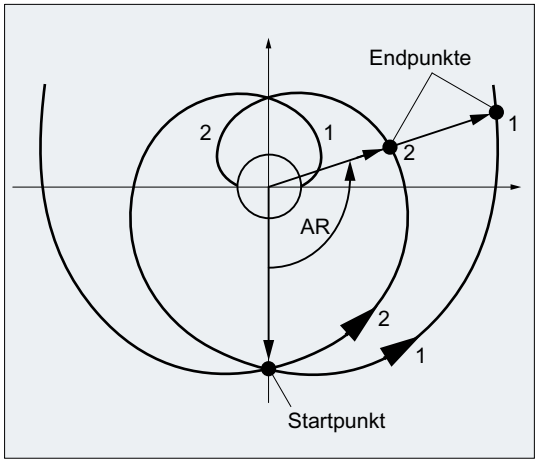
| | |
|---|---|
| `INVCW`: | Befehl zum Fahren auf einer Evolvente im Uhrzeigersinn |
| `INVCCW`: | Befehl zum Fahren auf einer Evolvente gegen den Uhrzeigersinn |
| `X... Y... Z...`: | Direkte Programmierung des Endpunkts in kartesischen Koordinaten |
| `I... J... K...`: | Interpolationsparameter zur Beschreibung des Mittelpunkts des Grundkreises in kartesischen Koordinaten<br>**Hinweis:**<br>Die Koordinatenangaben beziehen sich auf den Startpunkt der Evolvente. |
| `CR=...`: | Radius des Grundkreises |

| | | |
|---|---|---|
| AR=... : | Indirekte Programmierung des Endpunkts durch Angabe eines Öffnungswinkels (Drehwinkels)<br>Der Ursprung des Öffnungswinkels ist die Gerade vom Kreismittelpunkt zum Startpunkt. | |
| | AR > 0: | Die Bahn auf der Evolventen bewegt sich **vom Grundkreis weg**. |
| | AR < 0: | Die Bahn auf der Evolventen bewegt sich **zum Grundkreis hin**.<br>Für AR < 0 ist der maximale Drehwinkel dadurch beschränkt, dass der Endpunkt immer außerhalb des Grundkreises liegen muss. |

### Indirekte Programmierung des Endpunkts durch Angabe eines Öffnungswinkels

| ACHTUNG |
|---|
| **Öffnungswinkel undefiniert** |
| Bei der indirekten Programmierung des Endpunkts durch Angabe des Öffnungswinkels AR ist das Vorzeichen des Winkels zu beachten, da ein Vorzeichenwechsel eine andere Evolvente und damit eine andere Bahn zur Folge hätte. |

Dies soll anhand des folgenden Beispiels verdeutlicht werden:

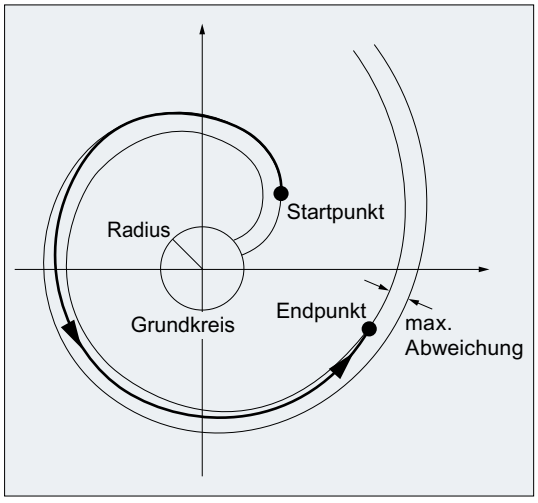



Für Evolvente 1 und 2 stimmen die Angaben von Radius und Mittelpunkt des Grundkreises, sowie des Startpunkts und des Drehsinns (INVCW / INVCCW) überein. Der einzige Unterschied besteht im Vorzeichen des Öffnungswinkels:

- Mit AR > 0 bewegt sich die Bahn auf der Evolvente 1 und es wird Endpunkt 1 angefahren.
- Mit AR < 0 bewegt sich die Bahn auf der Evolvente 2 und es wird Endpunkt 2 angefahren.

## Randbedingungen

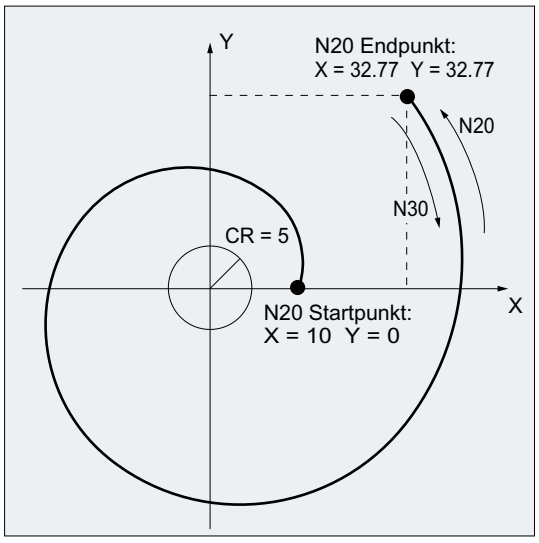
- Sowohl der Startpunkt als auch der Endpunkt müssen außerhalb der Fläche des Grundkreises der Evolvente (Kreis mit Radius CR um den durch I, J, K festgelegten Mittelpunkt) liegen. Trifft diese Bedingung nicht zu, wird ein Alarm generiert und die Programmverarbeitung abgebrochen.
- Die beiden Möglichkeiten zur Programmierung des Endpunkts (direkt über kartesische Koordinaten oder indirekt über die Angabe eines Öffnungswinkels) schließen sich gegenseitig aus. In einem Satz darf daher nur eine der beiden Programmiermöglichkeiten verwendet werden.
- Falls der programmierte Endpunkt nicht exakt auf der durch den Startpunkt und Grundkreis festgelegten Evolventen liegt, wird zwischen den beiden Evolventen, die durch den Startpunkt bzw. den Endpunkt definiert sind, interpoliert (siehe folgende Abbildung).




Die maximale Abweichung des Endpunkts wird durch ein Maschinendatum festgelegt (→ Maschinenhersteller!). Wenn die Abweichung des programmierten Endpunkts in radialer Richtung größer ist als der durch dieses MD festgelegte Wert, dann wird ein Alarm generiert und die Programmverarbeitung abgebrochen.

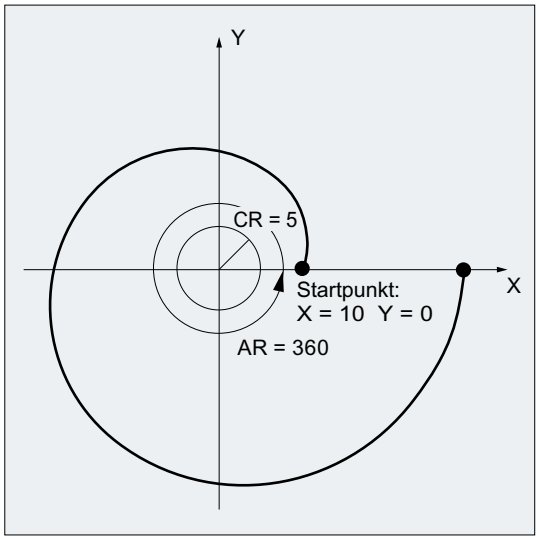
## Beispiele

### Beispiel 1: Linksdrehende Evolvente vom Startpunkt zum programmierten Endpunkt und als rechtsdrehende Evolvente wieder zurück




| **Programmcode** | **Kommentar** |
|---|---|
| `N10 G1 X10 Y0 F5000` | `; Anfahren der Startposition.` |
| `N15 G17` | `; Anwahl der X/Y-Ebene als Arbeitsebene.` |
| `N20 INVCCW X32.77 Y32.77 CR=5 I-10 J0` | `; Evolvente gegen den Uhrzeigersinn, Endpunkt in kartesischen Koordinaten.` |
| `N30 INVCW X10 Y0 CR=5 I-32.77 J-32.77` | `; Evolvente im Uhrzeigersinn, Startpunkt ist Endpunkt aus N20, neuer Endpunkt ist Startpunkt aus N20, neuer Kreismittelpunkt bezieht sich auf neuen Startpunkt und ist gleich dem alten Kreismittelpunkt.` |
| `...` | |

**Beispiel 2: Linksdrehende Evolvente mit indirekter Programmierung des Endpunkts durch Angabe eines Öffnungswinkels**




| Programmcode | Kommentar |
|---|---|
| `N10 G1 X10 Y0 F5000` | `; Anfahren der Startposition.` |
| `N15 G17` | `; Anwahl der X/Y-Ebene als Arbeitsebene.` |
| `N20 INVCCW CR=5 I-10 J0 AR=360` | `; Evolvente gegen den Uhrzeigersinn und weg vom Grundkreis (da positive Winkelangabe) mit einer vollen Umdrehung (360 Grad).` |
| `...` | |

## 2.9.9 Konturzüge

### 2.9.9.1 Konturzug-Programmierung

#### Funktion

Die Konturzug-Programmierung dient der schnellen Eingabe einfacher Konturen.

Programmierbar sind Konturzüge mit 1, 2, 3 oder mehr Punkten mit den Übergangselementen Fase oder Rundung durch Angabe von kartesischen Koordinaten und / oder Winkeln (ANG bzw. ANG1 und ANG2).

In den Sätzen, die Konturzüge beschreiben, können beliebige weitere NC-Adressen verwendet werden wie z. B. Adressbuchstaben für weitere Achsen (Einzelachsen oder Achse

senkrecht zur Bearbeitungsebene), Hilfsfunktionsangaben, G-Befehle, Geschwindigkeiten usw.

**Hinweis**

**Konturrechner**

Die Konturzug-Programmierung kann auf einfache Art und Weise auch mithilfe des Konturrechners erfolgen. Dabei handelt es sich um ein Werkzeug der Bedienoberfläche, das die Programmierung und grafische Darstellung einfacher und komplexer Werkstückkonturen ermöglicht. Die über den Konturrechner programmierten Konturen werden in das Teileprogramm übernommen.

**Weitere Informationen:** Bedienhandbuch

## Parametrierung

Die Bezeichner für Winkel, Radius und Fase werden über Maschinendaten definiert:

MD10652 $MN_CONTOUR_DEF_ANGLE_NAME (Name des Winkels für Konturzüge)

MD10654 $MN_RADIUS_NAME (Name des Radius für Konturzüge)

MD10656 $MN_CHAMFER_NAME (Name der Fase für Konturzüge)

**Hinweis**

Siehe Angaben des Maschinenherstellers.

### 2.9.9.2 Konturzüge: Eine Gerade

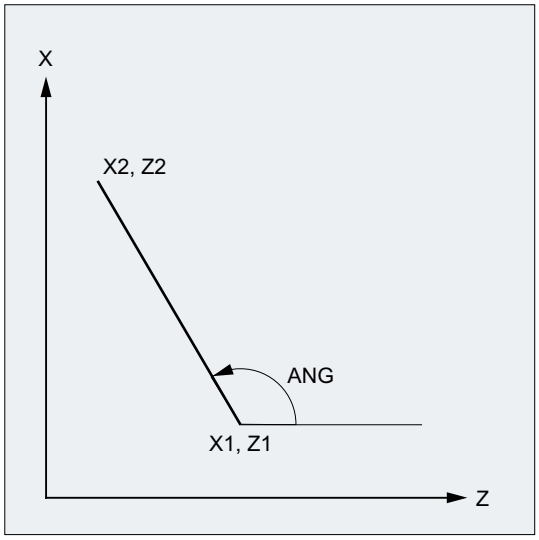
**Hinweis**

In der folgenden Beschreibung wird von davon ausgegangen, dass:

- G18 aktiv ist (⇒ aktive Arbeitsebene ist die Z/X-Ebene).
  (Die Programmierung von Konturzügen ist jedoch ohne Einschränkungen auch bei G17 oder G19 möglich.)
- für Winkel, Radius und Fase folgende Bezeichner definiert sind:
  - ANG (Winkel)
  - RND (Radius)
  - CHR (Fase

Der Endpunkt der Geraden wird definiert durch folgende Angaben:

- Winkel ANG
- **Eine** kartesische Endpunkt-Koordinate (X2 oder Z2)




| | |
|---|---|
| ANG: | Winkel der Geraden |
| X1, Z1: | Anfangskoordinaten |
| X2, Z2: | Endpunktkoordinaten der Geraden |

## Syntax

```
X... ANG=...
Z... ANG=...
```

## Bedeutung

| | |
|---|---|
| `X...` : | Endpunktkoordinate in X-Richtung |
| `Z...` : | Endpunktkoordinate in Z-Richtung |
| `ANG`: | Bezeichner zur Winkel-Programmierung<br>Der angegebene Wert (Winkel) bezieht sich auf die Abszisse der aktiven Arbeitsebene (Z-Achse bei `G18`). |

## Beispiel

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 X5 Z70 F1000 G18` | `; Anfahren der Startposition` |
| `N20 X88.8 ANG=110` | `; Gerade mit Winkelangabe` |
| `N30 ...` | |

Bzw.:

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 X5 Z70 F1000 G18` | `; Anfahren der Startposition` |
| `N20 Z39.5 ANG=110` | `; Gerade mit Winkelangabe` |
| `N30 ...` | |

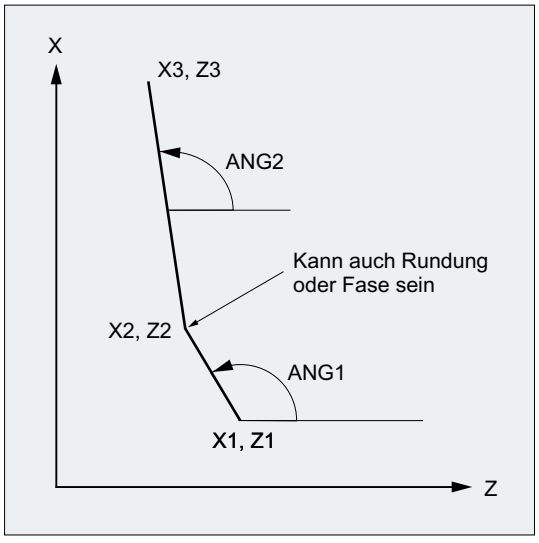
### 2.9.9.3 Konturzüge: Zwei Geraden

**Hinweis**

In der folgenden Beschreibung wird von davon ausgegangen, dass:

- G18 aktiv ist (⇒ aktive Arbeitsebene ist die Z/X-Ebene).
  (Die Programmierung von Konturzügen ist jedoch ohne Einschränkungen auch bei G17 oder G19 möglich.)
- für Winkel, Radius und Fase folgende Bezeichner definiert sind:
  – ANG (Winkel)
  – RND (Radius)
  – CHR (Fase)

Der Endpunkt der ersten Geraden kann durch Angabe der kartesischen Koordinaten oder durch Angabe der Winkel der beiden Geraden programmiert werden. Der Endpunkt der zweiten Geraden muss immer kartesisch programmiert werden. Der Schnittpunkt der beiden Geraden kann als Ecke, Rundung oder als Fase ausgeführt werden.




| | |
|---|---|
| ANG1: | Winkel der ersten Geraden |
| ANG2: | Winkel der zweiten Geraden |
| X1, Z1: | Anfangskoordinaten der ersten Geraden |
| X2, Z2: | Endpunktkoordinaten der ersten Geraden bzw.<br>Anfangskoordinaten der zweiten Geraden |
| X3, Z3: | Endpunktkoordinaten der zweiten Geraden |

## Syntax

**Programmierung des Endpunkts der ersten Geraden durch Angabe der Winkel**

- Ecke als Übergang zwischen den Geraden:

```
ANG=…
X… Z… ANG=…
```

- Rundung als Übergang zwischen den Geraden:

```
ANG=… RND=...
X… Z… ANG=…
```

- Fase als Übergang zwischen den Geraden:

```
ANG=… CHR=...
X… Z… ANG=…
```

**Programmierung des Endpunkts der ersten Geraden durch Angabe der Koordinaten**

- Ecke als Übergang zwischen den Geraden:

```
X… Z…
X… Z…
```

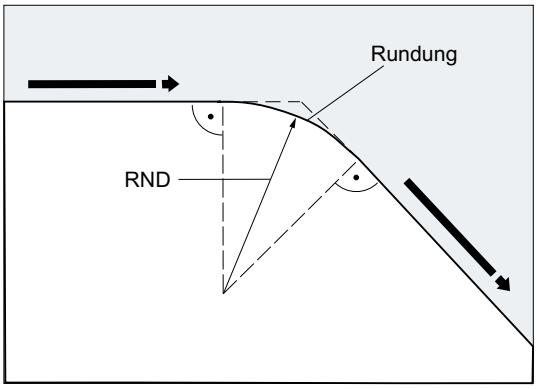
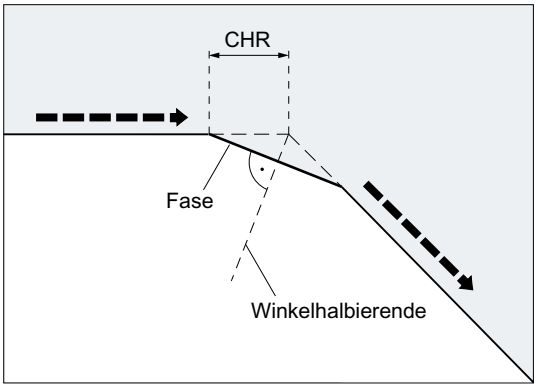
- Rundung als Übergang zwischen den Geraden:

```
X… Z… RND=...
X… Z…
```

- Fase als Übergang zwischen den Geraden:

```
X… Z… CHR=...
X… Z…
```

## Bedeutung

| | |
|---|---|
| `ANG=...` : | Bezeichner zur Winkel-Programmierung<br>Der angegebene Wert (Winkel) bezieht sich auf die Abszisse der aktiven Arbeitsebene (Z-Achse bei `G18`). |
| `RND=...` : | Bezeichner zur Programmierung einer Rundung<br>Der angegebene Wert entspricht dem Radius der Rundung:<br>  |
| `CHR=...` : | Bezeichner zur Programmierung einer Fase<br>Der angegebene Wert entspricht der Breite der Fase in der Bewegungsrichtung:<br>  |
| `X...` : | Koordinaten in X-Richtung |
| `Z...` : | Koordinaten in Z-Richtung |

### Hinweis

Weiterführende Informationen zur Programmierung einer Fase oder Rundung siehe " Fase, Rundung (CHF, CHR, RND, RNDM, FRC, FRCM) (Seite 248) ".

## Beispiel

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 X10 Z80 F1000 G18` | `; Anfahren der Startposition.` |

| **Programmcode** | **Kommentar** |
|---|---|
| N20 ANG=148.65 CHR=5.5 | ; Gerade mit Winkel- und Fasenangabe. |
| N30 X85 Z40 ANG=100 | ; Gerade mit Winkel- und Endpunktangabe. |
| N40 ... | |

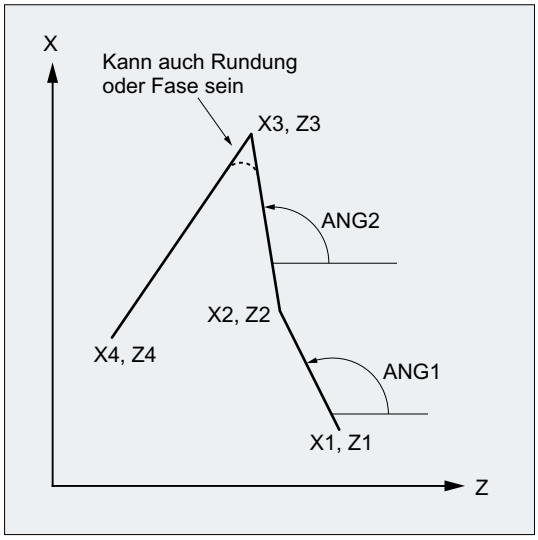
### 2.9.9.4 Konturzüge: Drei Geraden

**Hinweis**

In der folgenden Beschreibung wird von davon ausgegangen, dass:

- G18 aktiv ist (⇒ aktive Arbeitsebene ist die Z/X-Ebene).
  (Die Programmierung von Konturzügen ist jedoch ohne Einschränkungen auch bei G17 oder G19 möglich.)
- für Winkel, Radius und Fase folgende Bezeichner definiert sind:
  – ANG (Winkel)
  – RND (Radius)
  – CHR (Fase)

Der Endpunkt der ersten Geraden kann durch Angabe der kartesischen Koordinaten oder durch Angabe der Winkel der beiden Geraden programmiert werden. Der Endpunkt der zweiten und dritten Geraden muss immer kartesisch programmiert werden. Der Schnittpunkt der Geraden kann als Ecke, Rundung oder als Fase ausgeführt werden.




| | |
|---|---|
| ANG1: | Winkel der ersten Geraden |
| ANG2: | Winkel der zweiten Geraden |
| X1, Z1: | Anfangskoordinaten der ersten Geraden |
| X2, Z2: | Endpunktkoordinaten der ersten Geraden bzw.<br>Anfangskoordinaten der zweiten Geraden |
| X3, Z3: | Endpunktkoordinaten der zweiten Geraden bzw.<br>Anfangskoordinaten der dritten Geraden |
| X4, Z4: | Endpunktkoordinaten der dritten Geraden |

**Hinweis**

Die hier für einen 3-Punkt-Konturzug erläuterte Programmierung kann für Konturzüge mit mehr als drei Punkten beliebig fortgesetzt werden.

## Syntax

### Programmierung des Endpunkts der ersten Geraden durch Angabe der Winkel

- Ecke als Übergang zwischen den Geraden:

```
ANG=…
X… Z… ANG=…
X… Z…
```

- Rundung als Übergang zwischen den Geraden:

```
ANG=… RND=...
X… Z… ANG=… RND=...
X… Z…
```

- Fase als Übergang zwischen den Geraden:

```
ANG=… CHR=...
X… Z… ANG=… CHR=...
X… Z…
```

**Programmierung des Endpunkts der ersten Geraden durch Angabe der Koordinaten**

- Ecke als Übergang zwischen den Geraden:

```
X… Z…
X… Z…
X… Z…
```

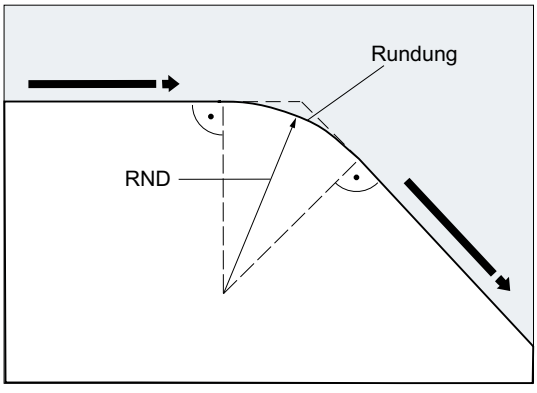
- Rundung als Übergang zwischen den Geraden:

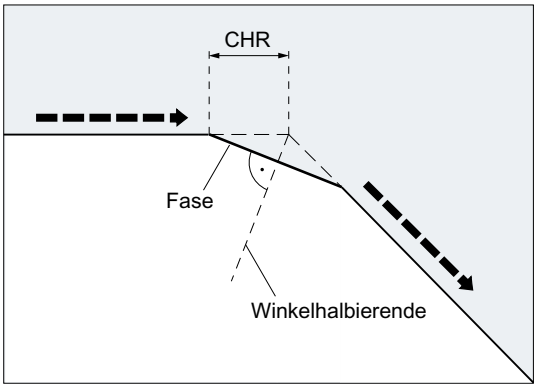
```
X… Z… RND=...
X… Z… RND=...
X… Z…
```

- Fase als Übergang zwischen den Geraden:

```
X… Z… CHR=...
X… Z… CHR=...
X… Z…
```

## Bedeutung

| | |
|---|---|
| `ANG=...` : | Bezeichner zur Winkel-Programmierung<br>Der angegebene Wert (Winkel) bezieht sich auf die Abszisse der aktiven Arbeitsebene (Z-Achse bei `G18`). |
| `RND=...` : | Bezeichner zur Programmierung einer Rundung<br>Der angegebene Wert entspricht dem Radius der Rundung:<br>  |

| | |
|---|---|
| CHR=... : | Bezeichner zur Programmierung einer Fase<br>Der angegebene Wert entspricht der Breite der Fase in der Bewegungsrichtung:<br>  <br> |
| X... : | Koordinaten in X-Richtung |
| Z... : | Koordinaten in Z-Richtung |

**Hinweis**

Weiterführende Informationen zur Programmierung einer Fase oder Rundung siehe " Fase, Rundung (CHF, CHR, RND, RNDM, FRC, FRCM) (Seite 248) ".

## Beispiel

| Programmcode | Kommentar |
|---|---|
| N10 X10 Z100 F1000 G18 | ; Anfahren der Startposition |
| N20 ANG=140 CHR=7.5 | ; Gerade mit Winkel- u. Fasenangabe |
| N30 X80 Z70 ANG=95.824 RND=10 | ; Gerade auf Zwischenpunkt mit Winkel- u. Rundungs-angabe |
| N40 X70 Z50 | ; Gerade auf Endpunkt |

### 2.9.9.5 Konturzüge: Endpunktprogrammierung mit Winkel

## Funktion

Erscheint in einem NC-Satz der Adressbuchstabe A, so dürfen zusätzlich keine, ein oder beide Achsen der aktiven Ebene programmiert sein.

**Anzahl der programmierten Achsen**

- Ist **keine Achse** der aktiven Ebene programmiert, so handelt es sich entweder um den ersten oder um den zweiten Satz eines Konturzugs, der aus zwei Sätzen besteht.
  Ist es der zweite Satz eines solchen Konturzugs, so bedeutet das, dass Start- und Endpunkt in der aktiven Ebene identisch sind. Der Konturzug besteht dann allenfalls aus einer Bewegung senkrecht zur aktiven Ebene.
- Ist **genau eine Achse** der aktiven Ebene programmiert, so handelt es sich entweder um eine einzelne Gerade, deren Endpunkt eindeutig aus dem Winkel und der programmierten kartesischen Koordinate bestimmt ist, oder um den zweiten Satz eines aus zwei Sätzen bestehenden Konturzugs. Im zweiten Fall wird die fehlende Koordinate gleich der letzten erreichten (modalen) Position gesetzt.
- Sind **zwei Achsen** der aktiven Ebene programmiert, handelt es sich um den zweiten Satz eines Konturzugs, der aus zwei Sätzen besteht. Ging dem aktuellen Satz kein Satz mit Winkelprogrammierung ohne programmierte Achsen der aktiven Ebene voraus, so ist ein solcher Satz nicht zulässig.

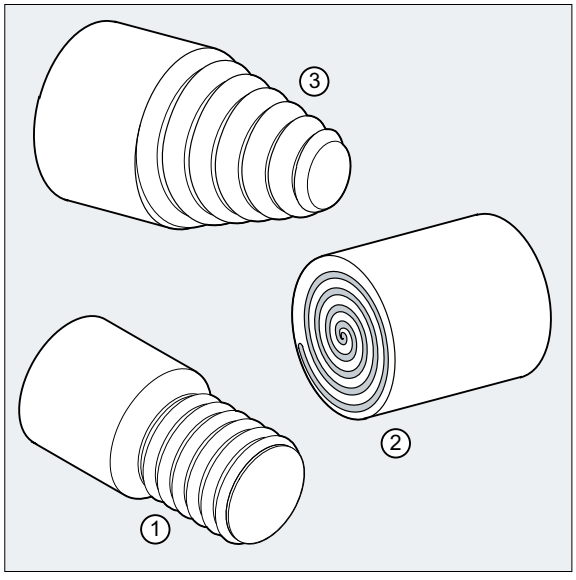
Der Winkel A darf nur bei Linear- oder Spline-Interpolation programmiert werden.

## 2.9.10 Gewindeschneiden

### 2.9.10.1 Gewindeschneiden mit konstanter Steigung (G33, SF)

Mit `G33` lassen sich Gewinde mit konstanter Steigung fertigen:

- Zylindergewinde ①
- Plangewinde ②
- Kegelgewinde ③

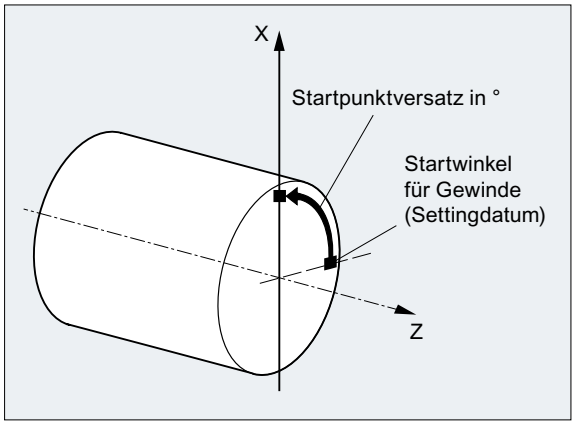


**Hinweis**

Technische Voraussetzung für das Gewindeschneiden mit G33 ist eine drehzahlgeregelte Spindel mit Wegmesssystem.

### Mehrgängige Gewinde

Mehrgängige Gewinde (Gewinde mit versetzten Schnitten) können durch die Angabe eines Startpunktversatzes hergestellt werden. Die Programmierung erfolgt im G33-Satz unter der Adresse SF.

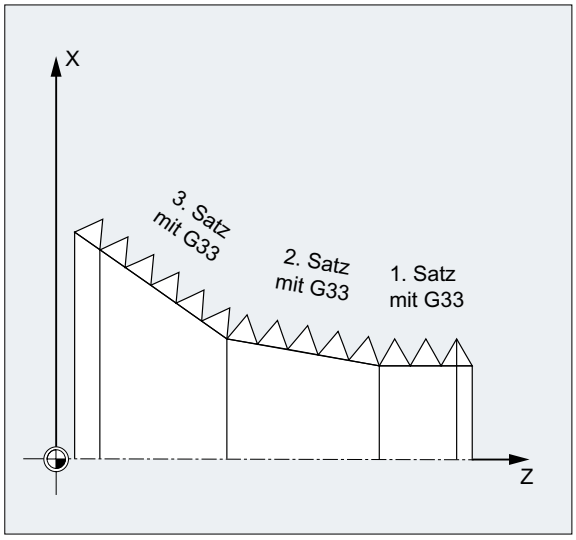



**Hinweis**

Falls kein Startpunktversatz angegeben ist, wird der in den Settingdaten festgelegte "Startwinkel für Gewinde" verwendet.

### Gewindekette

Durch mehrere, nacheinander programmierte G33-Sätze kann eine Gewindekette gefertigt werden:




**Hinweis**

Mit Bahnsteuerbetrieb `G64` werden die Sätze durch vorausschauende Geschwindigkeitsführung so miteinander verbunden, dass keine Geschwindigkeitssprünge entstehen.

### Drehrichtung des Gewindes

Die Drehrichtung des Gewindes wird durch die Drehrichtung der Spindel bestimmt:

- Rechtslauf mit `M3` erzeugt Rechtsgewinde
- Linkslauf mit `M4` erzeugt Linksgewinde

## Syntax

Zylindergewinde:

```
G33 Z… K…
G33 Z… K… SF=…
```

Plangewinde:

```
G33 X… I…
G33 X… I… SF=…
```

Kegelgewinde:

```
G33 X… Z… K…
G33 X… Z… K… SF=…
G33 X… Z… I…
G33 X… Z… I… SF=…
```

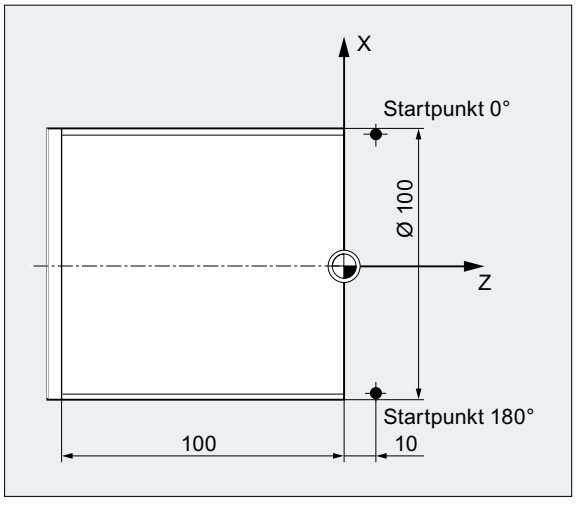
## Bedeutung

| | |
|---|---|
| `G33`: | Befehl zum Gewindeschneiden mit konstanter Steigung |
| `X... Y... Z...`: | Endpunkt(e) in kartesischen Koordinaten |
| `I...`: | Gewindesteigung in X-Richtung |

| | | |
|---|---|---|
| J...: | Gewindesteigung in Y-Richtung | |
| K...: | Gewindesteigung in Z-Richtung | |
| Z: | Längsachse | |
| X: | Planachse | |
| Z... K...: | Gewindelänge und Gewindesteigung für Zylindergewinde | |
| X... I...: | Gewindedurchmesser und Gewindesteigung für Plangewinde | |
| I... oder K...: | Gewindesteigung für Kegelgewinde<br>Die Angabe (I... oder K...) richtet sich nach dem Kegelwinkel: | |
| | < 45°: | Die Gewindesteigung wird mit K... angegeben (Gewindesteigung in Längsrichtung). |
| | > 45°: | Die Gewindesteigung wird mit I... angegeben (Gewindesteigung in Planrichtung). |
| | = 45°: | Die Gewindesteigung kann mit I... oder K... angegeben werden. |
| SF=...: | Startpunktversatz (nur notwendig bei mehrgängigen Gewinden!)<br>Der Startpunktversatz wird als absolute Winkelposition angegeben. | |
| | Wertebereich: | 0.0000 bis 359.999 Grad |

## Beispiele

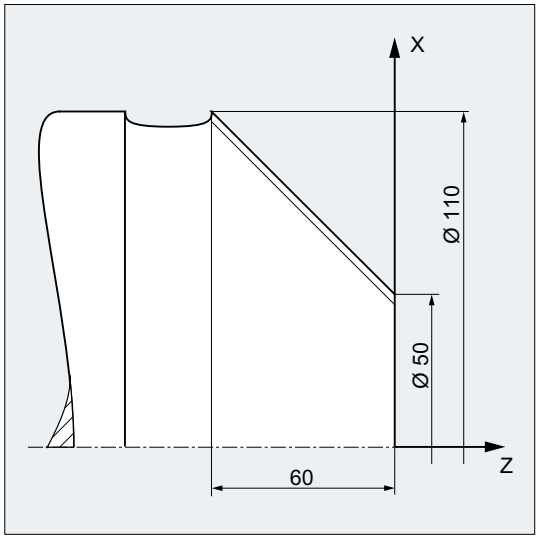
### Beispiel 1: Zweigängiges Zylindergewinde mit Startpunktversatz 180°




| Programmcode | Kommentar |
|---|---|
| N10 G1 G54 X99 Z10 S500 F100 M3 | ; Nullpunktverschiebung, Startpunkt anfahren, Spindel einschalten. |
| N20 G33 Z-100 K4 | ; Zylindergewinde: Endpunkt in Z |
| N30 G0 X102 | ; Rückzug auf Startposition. |
| N40 G0 Z10 | |
| N50 G1 X99 | |
| N60 G33 Z-100 K4 SF=180 | ; 2. Schnitt: Startpunktversatz 180° |
| N70 G0 X110 | ; Werkzeug wegfahren. |

| Programmcode | Kommentar |
|---|---|
| N80 G0 Z10 | |
| N90 M30 | ; Programmende. |

### Beispiel 2: Kegelgewinde mit Winkel kleiner 45°

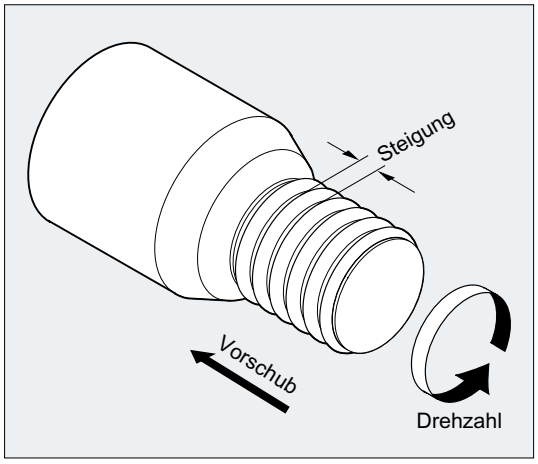



| Programmcode | Kommentar |
|---|---|
| N10 G1 X50 Z0 S500 F100 M3 | ; Startpunkt anfahren, Spindel einschalten. |
| N20 G33 X110 Z-60 K4 | ; Kegelgewinde: Endpunkt in X und Z, Angabe der Gewindesteigung mit K... in Z-Richtung (da Kegelwinkel < 45°). |
| N30 G0 Z0 M30 | ; Wegfahren, Programmende. |

## Weitere Informationen

### Vorschub beim Gewindeschneiden mit G33

Die Steuerung errechnet aus der programmierten Spindeldrehzahl und der Gewindesteigung den notwendigen Vorschub, mit dem der Drehstahl über die Gewindelänge in Längs- und / oder Planrichtung verfahren wird. Der Vorschub `F` wird bei `G33` nicht berücksichtigt, die Begrenzung auf maximale Achsgeschwindigkeit (Eilgang) wird von der Steuerung überwacht.



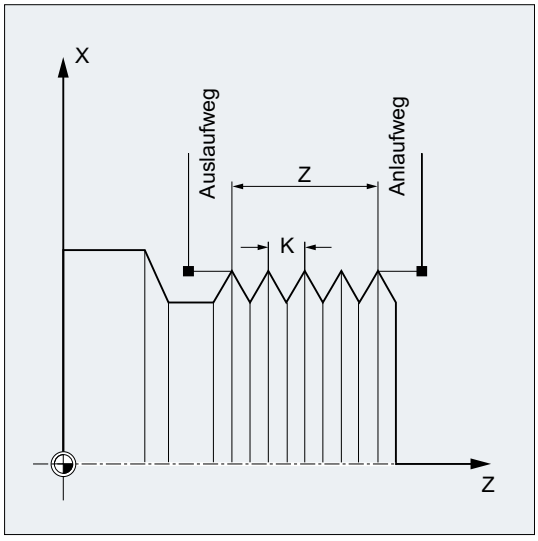

### Zylindergewinde

Das Zylindergewinde wird beschrieben durch:

- Gewindelänge
- Gewindesteigung

Die Gewindelänge wird mit einer der kartesischen Koordinaten X, Y oder Z im Absolut- oder Kettenmaß eingegeben (bei Drehmaschinen vorzugsweise in Z-Richtung). Zusätzlich sind Anlauf- und Auslaufwege zu berücksichtigen, auf denen der Vorschub hochgefahren bzw. reduziert wird.

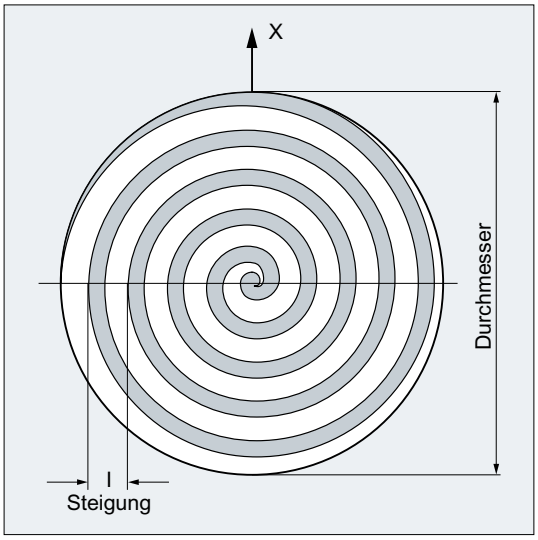
Die Gewindesteigung wird unter den Adressen `I`, `J`, `K` eingegeben (bei Drehmaschinen vorzugsweise mit `K`).




### Plangewinde

Das Plangewinde wird beschrieben durch:

- Gewindedurchmesser (vorzugsweise in X-Richtung)
- Gewindesteigung (vorzugsweise mit `I`)



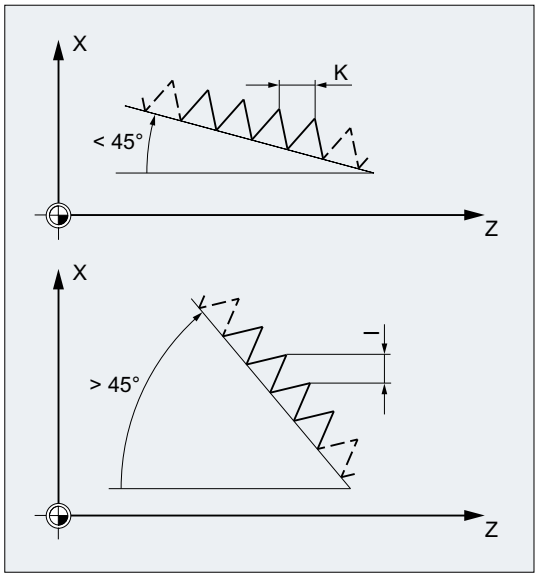

**Kegelgewinde**

Das Kegelgewinde wird beschrieben durch:

- Endpunkt in Längs- und Planrichtung (Kegelkontur)
- Gewindesteigung

Die Kegelkontur wird in kartesischen Koordinaten X, Y, Z im Bezugs- oder Kettenmaß eingegeben, bei der Bearbeitung auf Drehmaschinen vorzugsweise in X- und Z-Richtung. Zusätzlich sind Anlauf- und Auslaufwege zu berücksichtigen, auf denen der Vorschub hochgefahren bzw. reduziert wird.

Die Angabe für die Steigung richtet sich nach dem Kegelwinkel (Winkel zwischen Längsachse und Kegelmantel):



### 2.9.10.2 Gewindeschneiden mit zu- oder abnehmender Steigung (G34, G35)

Mit den Befehlen G34 und G35 wurde die G33-Funktionalität um die Möglichkeit erweitert, unter der Adresse F zusätzlich eine Änderung der Gewindesteigung zu programmieren. Im Falle von G34 führt das zu einer linearen Zunahme, im Falle von G35 zu einer linearen Abnahme der Gewindesteigung. Die Befehle G34 und G35 können damit zur Herstellung von selbstscherenden Gewinden eingesetzt werden.

## Syntax

Zylindergewinde mit zunehmender Steigung:
```
G34 Z… K… F...
```

Zylindergewinde mit abnehmender Steigung:
```
G35 Z… K… F...
```

Plangewinde mit zunehmender Steigung:
```
G34 X… I… F...
```

Plangewinde mit abnehmender Steigung:
```
G35 X… I… F...
```

Kegelgewinde mit zunehmender Steigung:
```
G34 X… Z… K… F...
G34 X… Z… I… F...
```

Kegelgewinde mit abnehmender Steigung:
```
G35 X… Z… K… F...
G35 X… Z… I… F...
```

## Bedeutung

| | | |
|---|---|---|
| `G34`: | Befehl zum Gewindeschneiden mit linear **zu**nehmender Steigung | |
| `G35`: | Befehl zum Gewindeschneiden mit linear **ab**nehmender Steigung | |
| `X... Y... Z...`: | Endpunkt(e) in kartesischen Koordinaten | |
| `I...`: | Gewindesteigung in X-Richtung | |
| `J...`: | Gewindesteigung in Y-Richtung | |
| `K...`: | Gewindesteigung in Z-Richtung | |
| `F...`: | Gewindesteigungsänderung<br>Ist die Anfangs- und Endsteigung eines Gewindes bekannt, dann kann die zu programmierende Gewindesteigungsänderung nach folgender Gleichung berechnet werden:<br>$F = \frac{k_e^2 - k_a^2}{2 * l_G}$ [ mm/U² ]<br>Dabei bedeuten: | |
| | $k_e$: | Gewindeendsteigung (Gewindesteigung der Achszielpunktkoordinate) [mm/U] |
| | $k_a$: | Gewindeanfangssteigung (unter I, J oder K programmiert) [mm/U] |
| | $l_G$: | Gewindelänge [mm] |

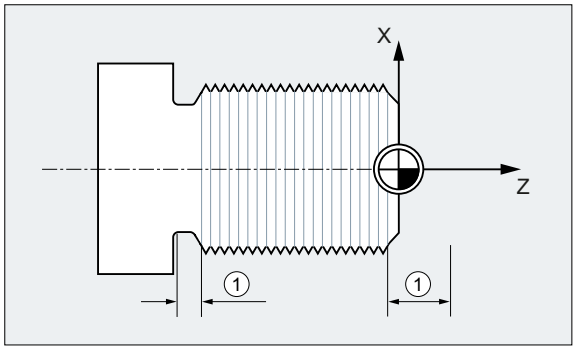
## Beispiel

| Programmcode | Kommentar |
|---|---|
| N1608 M3 S10 | ; Spindel ein. |
| N1609 G0 G64 Z40 X216 | ; Startpunkt anfahren. |
| N1610 G33 Z0 K100 SF=R14 | ; Gewindeschneiden mit konstanter Steigung (100 mm/U) |
| N1611 G35 Z-200 K100 F17.045455 | ; Steigungsabnahme: 17.0454 mm/U2 |
| | Steigung am Satzende: 50mm/U |
| N1612 G33 Z-240 K50 | ; Gewindesatz ohne Ruck fahren. |
| N1613 G0 X218 | |
| N1614 G0 Z40 | |
| N1615 M17 | |

### 2.9.10.3 Programmierter Einlauf- und Auslaufweg bei G33, G34 und G35 (DITS, DITE)

Mit den Adressen DITS und DITE kann im Teileprogramm der Einlauf- und Auslaufweg des Gewindes vorgegeben werden.

Die Gewindeachse wird innerhalb des vorgegebenen Wegs beschleunigt bzw. gebremst.




① Ein- bzw. Auslaufweg, je nach Bearbeitungsrichtung

#### Kurzer Einlaufweg

Durch den Bund am Gewindeeinlauf ist wenig Platz für die Werkzeug-Startrampe.
Diese muss deshalb über DITS kürzer vorgegeben werden.

#### Kurzer Auslaufweg

Durch den Bund am Gewindeauslauf ist wenig Platz für die Werkzeug-Bremsrampe, wodurch Kollisionsgefahr zwischen Werkstück und Schneide besteht. Die Bremsrampe kann über DITE kürzer vorgegeben werden. Wegen der Trägheit der Mechanik kann es dennoch zur Kollision kommen.

Abhilfe: Gewinde kürzer programmieren, Spindeldrehzahl reduzieren.

**Hinweis**

DITE wirkt am Gewindeende als Überschleifabstand. Damit wird eine stoßfreie Änderung der Achsbewegung erreicht.

### Auswirkungen

Der programmierte Einlauf- und Auslaufweg wirkt auf die Bahn ausschließlich beschleunigungssteigernd. Wird einer der beiden Wege größer vorgegeben, als er für die Gewindeachse mit aktiver Beschleunigung benötigt wird, wird die Gewindeachse mit maximaler Beschleunigung beschleunigt bzw. gebremst.

## Syntax

```
DITS=<Wert> DITE=<Wert>
```

## Bedeutung

| | |
|---|---|
| DITS: | Gewinde-Einlaufweg festlegen |
| DITE: | Gewinde-Auslaufweg festlegen |
| <Wert>: | Unter DITS und DITE werden ausschließlich Wege programmiert, keine Positionen!<br>Der programmierte Ein-/Auslaufweg wird entsprechend der aktuellen Maßeinstellung (Inch, metrisch) behandelt. |

## Beispiel

| Programmcode | Kommentar |
|---|---|
| ... | |
| N40 G90 G0 Z100 X10 SOFT M3 S500 | |
| N50 G33 Z50 K5 SF=180 DITS=1 DITE=3 | ; Überschleifbeginn bei Z=53. |
| N60 G0 X20 | |

## Weitere Informationen

### SD42010 $SC_THREAD_RAMP_DISP

Mit dem Einwechseln eines Satzes mit DITS und/oder DITE in den Hauptlauf wird der programmierte Ein- bzw. Auslaufweg in das Settingdatum SD42010 $SC_THREAD_RAMP_DISP übernommen:

- SD42010 $SC_THREAD_RAMP_DISP[ **0** ] = programmierter Wert von DITS
- SD42010 $SC_THREAD_RAMP_DISP[ **1** ] = programmierter Wert von DITE

Wird vor oder im ersten Gewindesatz kein Ein- bzw. Auslaufweg programmiert, wird jeweils der aktuelle Wert des Settingdatums verwendet.

#### Verhalten nach Kanal-/BAG-/Programmende-Reset

Die durch DITS und/oder DITE überschriebenen Werte von SD42010 bleiben auch nach einem Kanal-/BAG-/Programmende-Reset wirksam.

#### Verhalten nach Warmstart

Bei einem Warmstart wird das Settingdatum auf die Werte zurückgesetzt, die vor dem Überschreiben durch DITS und/oder DITE wirksam waren (Standardverhalten).

Sollen die mit DITS und DITE programmierten Werte dagegen auch nach einem Warmstart wirksam sein, muss das Settingdatum SD42010 $SC_THREAD_RAMP_DISP im Maschinendatum MD10710 $MN_PROG_SD_RESET_SAVE_TAB aufgelistet sein:

MD10710 $MN_PROG_SD_RESET_SAVE_TAB[<n>] = 42010

#### Verhalten bei sehr kleinem Ein- und/oder Auslaufweg

Bei sehr kleinem Ein- und/oder Auslaufweg wird die Gewindeachse stärker beschleunigt, als es die Projektierung vorsieht. Die Achse wird dann beschleunigungsmäßig überlastet.

Für den Gewindeeinlauf wird dann der Alarm 22280 "Programmierter Einlaufweg zu kurz" gemeldet (bei entsprechender Projektierung im MD11411 $MN_ENABLE_ALARM_MASK). Der Alarm ist rein informativ und hat keine Auswirkungen auf die Teileprogrammabarbeitung.

### 2.9.10.4 Schnellrückzug während Gewindeschneiden (LFON, LFOF, DILF, ALF, LFTXT, LFWP, LFPOS, POLF, POLFMASK, POLFMLIN)

Die Funktion "Schnellrückzug während Gewindeschneiden (G33)" ermöglicht eine zerstörungsfreie Unterbrechung des Gewindeschneidens in folgenden Situationen:

- NC-Stop ausgelöst über DB21, ... DBX7.3
- NC-Stop implizit ausgelöst über Alarm
- Schalten eines schnellen Eingangs (Seite 552)

Die Rückzugsbewegung ist programmierbar über:

- Rückzugweg und Rückzugsrichtung (relativ)
- Rückzugsposition (absolut)

**Hinweis**

**NC-Stop-Signale**

Folgende NC-Stop-Signale lösen keinen Schnellrückzug während des Gewindeschneidens aus:

- DB21, ... DBX3.4 (NC-Stop Achsen plus Spindeln)
- DB21, ... DBX7.2 (NC-Stop an Satzgrenze)

**Gewindebohren**

Die Funktion "Schnellrückzug" ist beim Gewinde**bohren** (G331 / G332) **nicht** anwendbar.

## Syntax

Schnellrückzug freigeben, Rückzugsbewegung über Rückzugweg und Rückzugsrichtung:

```
G33 ... LFON DILF=<Wert> LFTXT/LFWP ALF=<Wert>
```

Schnellrückzug freigeben, Rückzugsbewegung über Rückzugsposition:

```
POLF[<Achsbezeichner>]=<Wert> LFPOS
POLFMASK/POLFMLIN(<Achsname1>,<Achsname2>,...)
G33 ... LFON
```

Schnellrückzug während Gewindeschneiden sperren:
`LFOF`

## Bedeutung

| | | |
|---|---|---|
| `LFON`: | Schnellrückzug während Gewindeschneiden (`G33`) freigeben | |
| `LFOF`: | Schnellrückzug während Gewindeschneiden (`G33`) sperren | |
| `DILF=` : | Länge des Rückzugswegs festlegen | |
| | Der durch MD-Projektierung (MD21200 $MC_LIFTFAST_DIST) voreingestellte Wert kann im Teileprogramm durch Programmierung von `DILF` verändert werden.<br>**Hinweis:**<br>Nach NC-RESET ist immer der projektierte MD-Wert aktiv. | |
| `LFTXT`<br>`LFWP`: | Die Rückzugsrichtung wird in Verbindung mit `ALF` mit den G-Befehlen `LFTXT` und `LFWP` gesteuert. | |
| | `LFTXT`: | Die Ebene, in welcher die Rückzugsbewegung ausgeführt wird, wird aus der Bahntangente und der Werkzeugrichtung errechnet (Standardeinstellung). |
| | `LFWP`: | Die Ebene, in welcher die Rückzugsbewegung ausgeführt wird, ist die aktive Arbeitsebene. |
| `ALF=` : | In der Ebene der Rückzugsbewegung wird mit `ALF` die Richtung in diskreten Schritten von Grad programmiert. | |
| | Bei `LFTXT` ist für `ALF=1` der Rückzug in Werkzeugrichtung festgelegt.<br>Bei `LFWP` ergibt sich die Richtung in der Arbeitsebene nach folgender Zuordnung:<br>• `G17` (X/Y-Ebene)<br>`ALF=1` ; Rückzug in X-Richtung<br>`ALF=3` ; Rückzug in Y-Richtung<br>• `G18` (Z/X-Ebene)<br>`ALF=1` ; Rückzug in Z-Richtung<br>`ALF=3` ; Rückzug in X-Richtung<br>• `G19` (Y/Z-Ebene)<br>`ALF=1` ; Rückzug in Y-Richtung<br>`ALF=3` ; Rückzug in Z-Richtung | |
| | **Weitere Informationen** zu den Programmiermöglichkeiten mit `ALF` siehe Kapitel "Verfahrrichtung beim Schnellabheben von der Kontur (Seite 554)". | |
| `LFPOS`: | Rückzug der mit `POLFMASK` oder `POLFMLIN` bekannt gemachten Achse auf die mit `POLF` programmierte absolute Achsposition | |
| `POLFMASK`: | Freigabe der Achsen `(<Achsname1>,<Achsname1>,...)` für den unabhängigen Rückzug auf Absolutposition | |

| | | |
|---|---|---|
| POLFMLIN: | Freigabe der Achsen für den Rückzug auf Absolutposition im linearen Zusammenhang<br>**Hinweis:**<br>Der lineare Zusammenhang kann je nach dynamischen Verhalten aller beteiligten Achsen bis zum Erreichen der Abhebeposition nicht immer hergestellt werden. | |
| POLF[]: | Absolute Rückzugsposition für die im Index angegebene Geometrieachse bzw. Maschinenachse festlegen | |
| | Wirksamkeit: | modal |
| | =<Wert>: | Bei Geometrieachsen wird der zugewiesene Wert als Position im Werkstückkoordinatensystem (WKS) interpretiert, bei Maschinenachsen als Position im Maschinenkoordinatensystem (MKS).<br>Die Wertzuweisung ist auch als Kettenmaßangabe programmierbar:<br>=IC<Wert> |
| <Achsbezeichner>: | Bezeichner einer Geometrie- oder Maschinenachse | |

**Hinweis**

LFON bzw. LFOF können immer programmiert werden, die Auswertung erfolgt jedoch ausschließlich beim Gewindeschneiden (G33).

**Hinweis**

POLF mit POLFMASK/POLFMLIN sind nicht auf den Einsatz bei Gewindeschneiden beschränkt.

## Beispiele

### Beispiel 1: Schnellrückzug während Gewindeschneiden freigeben

```
Programmcode                                Kommentar
N55 M3 S500 G90 G18                         ; Aktive Bearbeitungsebene
...                                         ; Anfahren der Startposition
N65 MSG ("Gewindeschneiden")                ; Zustellen des Werkzeugs
MM_THREAD:
N67 $AC_LIFTFAST=0                          ; Vor Beginn des Gewindes zurücksetzen.
N68 G0 Z5
N68 X10
N70 G33 Z30 K5 LFON DILF=10 LFWP ALF=7      ; Schnellrückzug während Gewindeschnei-
                                            den freigeben.
                                            Rückzugsweg=10mm
                                            Rückzugsebene: Z/X (wegen G18)
                                            Rückzugsrichtung: -X
                                            (mit ALF=3: Rückzugsrichtung +X)
N71 G33 Z55 X15
N72 G1                                      ; Gewindeschneiden abwählen.
```

| Programmcode | Kommentar |
|---|---|
| N69 IF $AC_LIFTFAST GOTOB MM_THREAD | ; Wenn Gewindeschneiden unterbrochen wurde. |
| N90 MSG("") | |
| ... | |
| N70 M30 | |

### Beispiel 2: Schnellrückzug vor Gewindebohren ausschalten

| Programmcode | Kommentar |
|---|---|
| N55 M3 S500 G90 G0 X0 Z0 | |
| ... | |
| N87 MSG ("Gewindebohren") | |
| N88 LFOF | ; Schnellrückzug vor Gewindebohren ausschalten. |
| N89 CYCLE... | ; Gewindebohrzyklus mit G33. |
| N90 MSG ("") | |
| ... | |
| N99 M30 | |

### Beispiel 3: Schnellrückzug auf absolute Rückzugsposition

Bei einem Stopp wird die Bahninterpolation von X unterdrückt und stattdessen eine Bewegung mit max. Geschwindigkeit auf die Position POLF[X] interpoliert. Die Bewegung der anderen Achsen wird weiterhin durch die programmierte Kontur bzw. die Gewindesteigung und die Spindeldrehzahl bestimmt.

| Programmcode | Kommentar |
|---|---|
| N10 G0 G90 X200 Z0 S200 M3 | |
| N20 G0 G90 X170 | |
| N22 POLF[X]=210 LFPOS | |
| N23 POLFMASK(X) | ; Aktivieren (Freigeben) des Schnellabhebens von der Achse X. |
| N25 G33 X100 I10 LFON | |
| N30 X135 Z-45 K10 | |
| N40 X155 Z-128 K10 | |
| N50 X145 Z-168 K10 | |
| N55 X210 I10 | |
| N60 G0 Z0 LFOF | |
| N70 POLFMASK() | ; Abheben für alle Achsen sperren. |
| M30 | |

### 2.9.10.5 Ballige Gewinde (G335, G336)

Mit den G-Befehlen G335 und G336 besteht die Möglichkeit, ballige (= von der zylindrischen Form abweichende) Gewinde zu drehen. Anwendung ist die Bearbeitung extrem großer Bauteile, die durch ihr Eigengewicht in der Maschine durchhängen. Achsparallele Gewinde würden dazu führen, dass der Gewindegang in der Mitte des Bauteils zu gering ist. Mit den balligen Gewinden kann dies ausgeglichen werden.

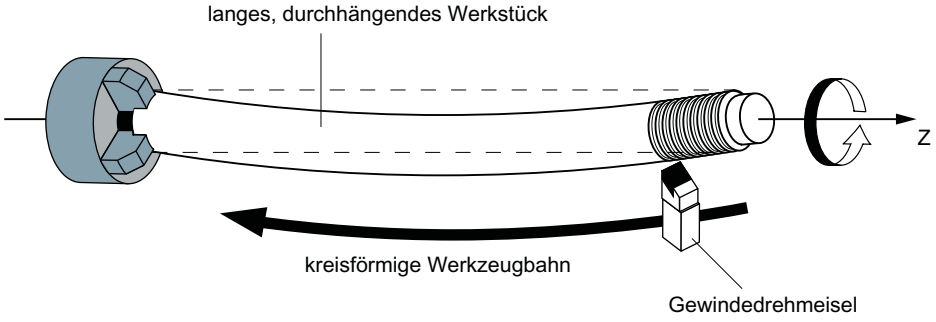



Bild 2-6 Drehen eines balligen Gewindes

## Programmierung

Das Drehen balliger Gewinde wird mit `G335` oder `G336` programmiert:

| | |
|---|---|
| `G335`: | Drehen eines balligen Gewindes auf einer **im Uhrzeigersinn** verlaufenden kreisförmigen Werkzeugbahn |
| `G336`: | Drehen eines balligen Gewindes auf einer **gegen den Uhrzeigersinn** verlaufenden kreisförmigen Werkzeugbahn |

Die Programmierung erfolgt zunächst wie bei einem linearen Gewinde über die Angabe der axialen Satzendpunkte und der Steigung über die Parameter `I`, `J`, und `K` (siehe "Gewindeschneiden mit konstanter Steigung (G33, SF) (Seite 224)").

Zusätzlich wird noch ein Kreisbogen mit angegeben. Dieser kann wie bei `G2`/`G3` über die Mittelpunkt-, Radius-, Öffnungswinkel- oder Zwischenpunktangabe programmiert werden (siehe "Kreisinterpolation (Seite 191)"). Bei Programmierung des balligen Gewindes mit Mittelpunktprogrammierung ist dabei Folgendes zu beachten: Da `I`, `J`, und `K` bei Gewindeschneiden für die Steigung benutzt werden, müssen die Kreisparameter bei Mittelpunktprogrammierung mit `IR=...`, `JR=...` und `KR=...` programmiert werden.

| | |
|---|---|
| `IR=...`: | Kartesische Koordinate für den Kreismittelpunkt in X-Richtung |
| `JR=...`: | Kartesische Koordinate für den Kreismittelpunkt in Y-Richtung |
| `KR=...`: | Kartesische Koordinate für den Kreismittelpunkt in Z-Richtung |

**Hinweis**

IR, JR und KR sind die Standardwerte der über Maschinendatum (MD10651 $MN_IPO_PARAM_THREAD_NAME_TAB) einstellbaren Namen der Interpolationsparameter für ballige Gewinde.

Abweichungen von diesen Standardwerten sind den Angaben des Maschinenherstellers zu entnehmen!

Optional kann noch ein Startpunktversatz `SF` mit angegeben werden (siehe "Gewindeschneiden mit konstanter Steigung (G33, SF) (Seite 224)").

**Syntax**

Die Syntax zur Programmierung eines balligen Gewindes hat also folgende allgemeine Form:
`G335/G336 <Achszielpunktkoordinate(n)> <Steigung> <Kreisbogen> [<Startpunktversatz>]`

## Beispiele

### Beispiel 1: Balliges Gewinde im Uhrzeigersinn mit End- und Mittelpunktprogrammierung

| Programmcode | Kommentar |
|---|---|
| `N5 G0 G18 X50 Z50` | `; Anfangspunkt anfahren.` |
| `N10 G335 Z100 K=3.5 KR=25 IR=-20 SF=90` | `; Balliges Gewinde im Uhrzeigersinn drehen.` |

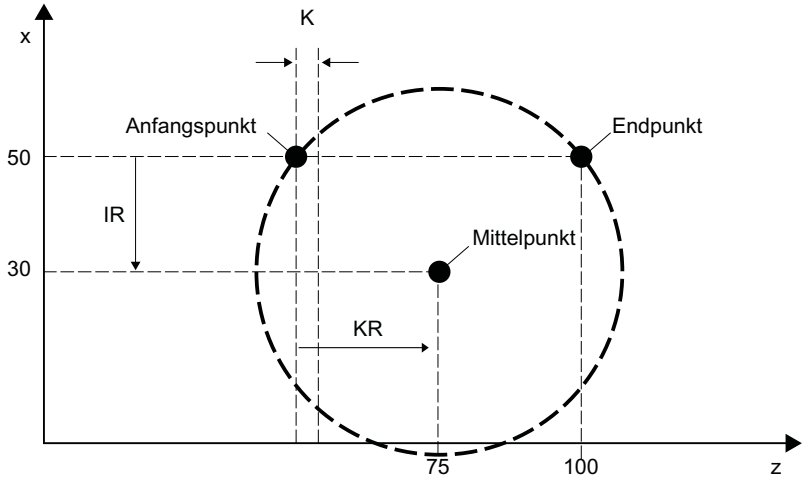



Bild 2-7 Balliges Gewinde im Uhrzeigersinn mit End- und Mittelpunktprogrammierung

### Beispiel 2: Balliges Gewinde gegen den Uhrzeigersinn mit End- und Mittelpunktprogrammierung

| Programmcode | Kommentar |
|---|---|
| `N5 G0 G18 X50 Z50` | `; Anfangspunkt anfahren.` |

| Programmcode | Kommentar |
|---|---|
| N10 **G336 Z100** K=3.5 **KR=25 IR=20** SF=90 | ; Balliges Gewinde gegen den Uhrzeigersinn drehen. |

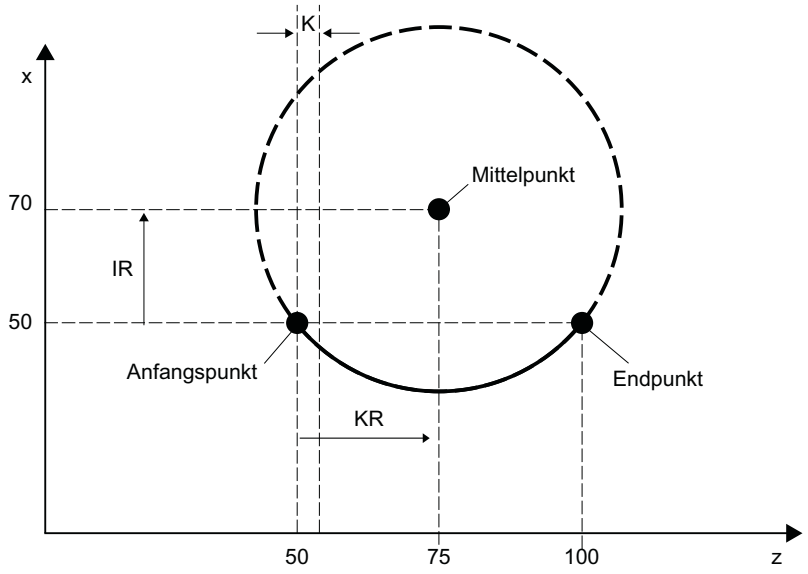



Bild 2-8 Balliges Gewinde gegen den Uhrzeigersinn mit End- und Mittelpunktprogrammierung

### Beispiel 3: Balliges Gewinde im Uhrzeigersinn mit Endpunkt- und Radiusprogrammierung

| Programmcode |
|---|
| N5 G0 G18 X50 Z50 |
| N10 **G335 Z100** K=3.5 **CR=32** SF=90 |

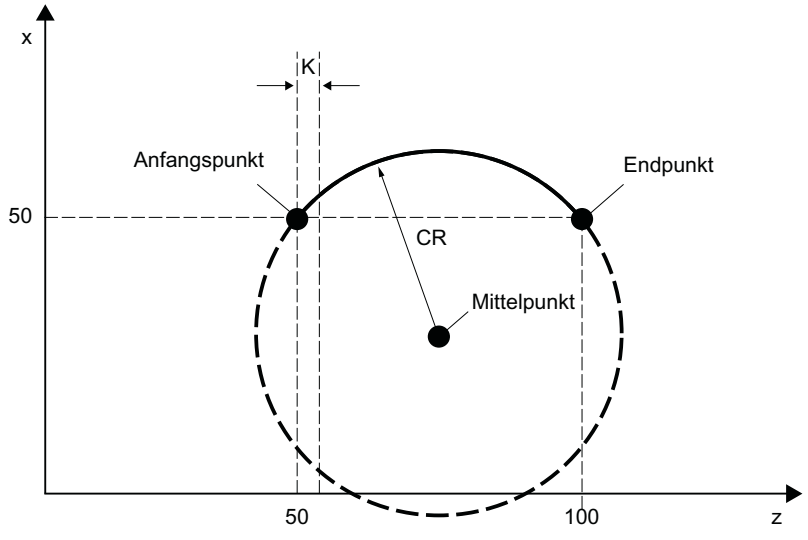



Bild 2-9 Balliges Gewinde im Uhrzeigersinn mit Endpunkt- und Radiusprogrammierung

### Beispiel 4: Balliges Gewinde im Uhrzeigersinn mit Endpunkt- und Öffnungswinkelprogrammierung

**Programmcode**

```
N5 G0 G18 X50 Z50
N10 G335 Z100 K=3.5 AR=102.75 SF=90
```

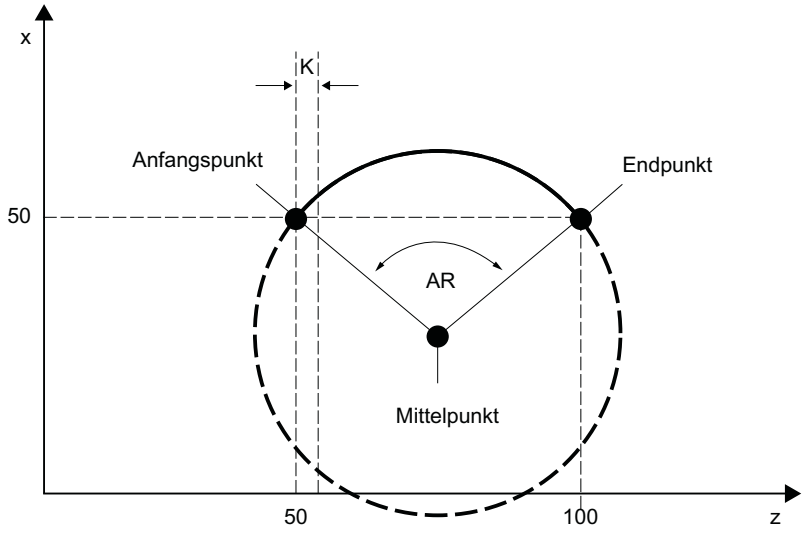



Bild 2-10 Balliges Gewinde im Uhrzeigersinn mit Endpunkt- und Öffnungswinkelprogrammierung

### Beispiel 5: Balliges Gewinde im Uhrzeigersinn mit Mittelpunkt- und Öffnungswinkelprogrammierung

**Programmcode**

```
N5 G0 G18 X50 Z50
N10 G335 K=3.5 KR=25 IR=-20 AR=102.75 SF=90
```

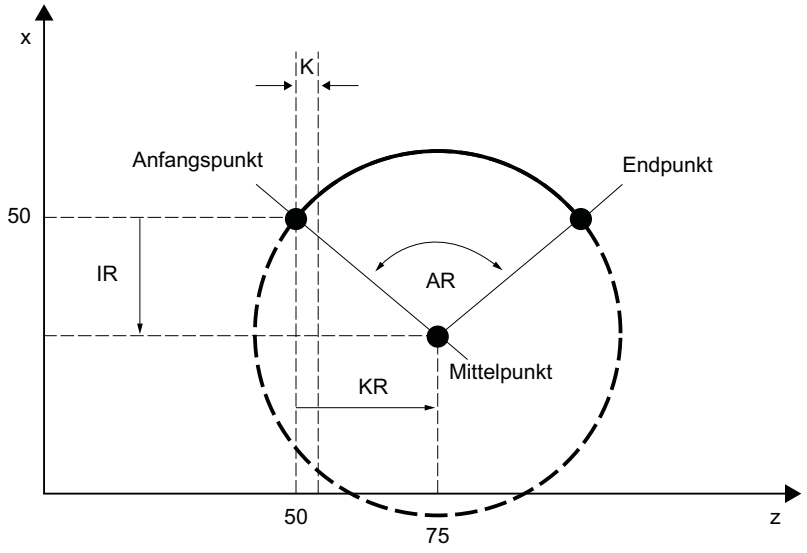


Bild 2-11 Balliges Gewinde im Uhrzeigersinn mit Mittelpunkt- und Öffnungswinkelprogrammierung

### Beispiel 6: Balliges Gewinde im Uhrzeigersinn mit End- und Zwischenpunktprogrammierung

**Programmcode**

```
N5 G0 G18 X50 Z50
N10 G335 Z100 K=3.5 I1=60 K1=64
```

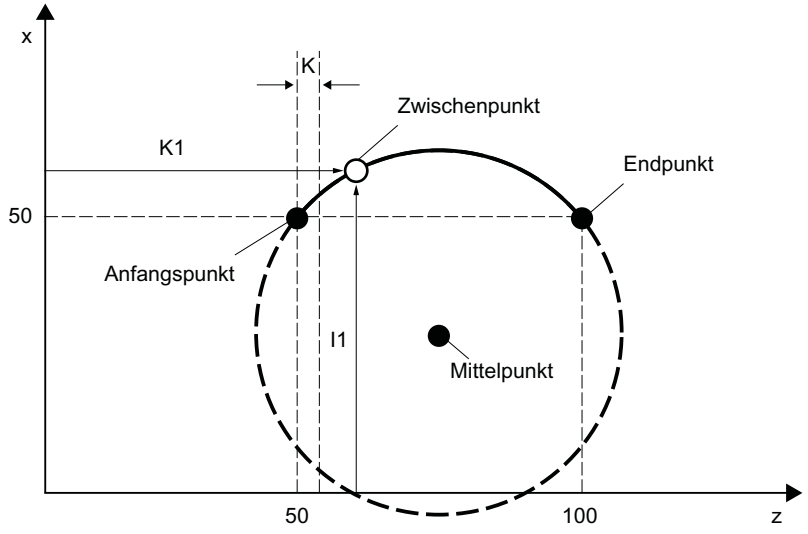
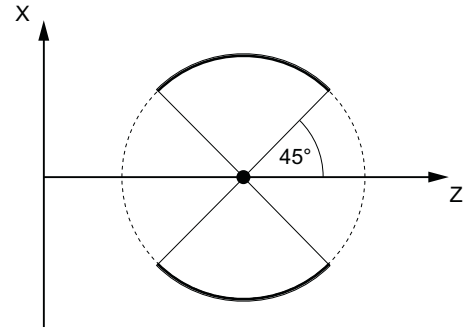



Bild 2-12 Balliges Gewinde im Uhrzeigersinn mit End- und Zwischenpunktprogrammierung

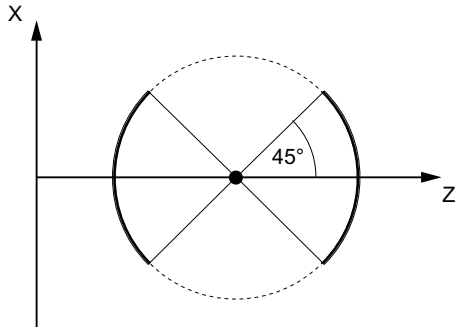
## Weitere Informationen

### Erlaubte Kreisbogenbereiche

Der unter G335/G336 programmierte Kreisbogen muss in einem Bereich liegen, in dem die spezifizierte Gewindehauptachse (I, J oder K) über den gesamten Kreisbogen den Hauptachsanteil am Kreisbogen hat:




Erlaubte Bereiche für die **Z**-Achse (Steigung mit K programmiert)




Erlaubte Bereiche für die **X**-Achse (Steigung mit I programmiert)

Ein Wechsel der Gewindehauptachse, wie im folgenden Bild dargestellt, ist **nicht** erlaubt:

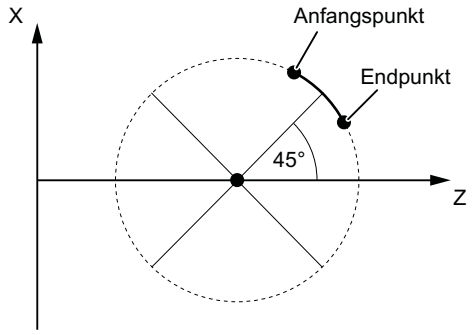



Bild 2-13 Ballige Gewinde: Nicht erlaubter Bereich

#### Frames

Auch bei aktiven Frames ist G335 und G336 möglich. Es ist allerdings darauf zu achten, dass im Basiskoordinatensystem (BKS) die erlaubten Kreisbogenbereiche eingehalten werden.

#### Randbedingungen zur Kreisprogrammierung

Für die Kreisprogrammierung unter G335/G336 gelten die bei der Kreisprogrammierung mit G2/G3 beschriebenen Randbedingungen (siehe "Kreisinterpolation (Seite 191)").

## 2.9.11 Gewindebohren ohne Ausgleichsfutter

### 2.9.11.1 Gewindebohren ohne Ausgleichsfutter und Rückzugsbewegung (G331, G332)

Beim Gewindebohren ohne Ausgleichsfutter werden mit den Befehlen `G331` und `G332` folgenden Verfahrbewegungen ausgeführt:

- `G331`: Gewindebohren in Bohrrichtung bis zum Gewindeendpunkt
- `G332`: Rückzugsbewegung zum Gewindebohrsatz `G331` mit automatischer Spindeldrehrichtungsumkehr

#### Syntax

```
G331 <Achse> <Gewindesteigung>
G331 <Achse> <Gewindesteigung> S...
G332 <Achse> <Gewindesteigung>
```

#### Bedeutung

| | | |
|---|---|---|
| `G331`: | Gewindebohren<br>Die Gewindebohrung wird beschrieben durch die Verfahrbewegung der Achse (Bohrtiefe) und die Gewindesteigung. | |
| | Wirksamkeit: | modal |

| | | |
|---|---|---|
| G332: | Rückzugsbewegung des Gewindebohrens<br>Die Rückzugsbewegung muss mit derselben Steigung ausgeführt werden wie das Gewindebohren (G331). Die Richtungsumkehr der Spindel erfolgt automatisch. | |
| | Wirksamkeit: | modal |
| <Achse>: | Verfahrweg/position der Geometrieachse (X, Y oder Z) zum Gewindeendpunkt, z. B. Z50 | |
| <Gewindesteigung>: | Gewindesteigung I (X), J (Y) oder K (Z):<br>• Positive Steigung: Rechtsgewinde, z. B. K1,25<br>• Negative Steigung: Linksgewinde, z. B. K-1,25 | |
| | Wertebereich: | ±0.001 bis ±2000.00 mm/Umdrehung |
| S...: | Spindeldrehzahl<br>Ist keine Spindeldrehzahl angegeben, wird mit der zuletzt wirksamen Spindeldrehzahl verfahren. | |

**Hinweis**

**Zweiter Getriebestufen-Datensatz**

Um beim Gewindebohren eine effektive Anpassung von Spindeldrehzahl und Motormoment zu erreichen und schneller beschleunigen zu können, kann in achsspezifischen Maschinendaten abweichend vom ersten Getriebestufen-Datensatz und auch unabhängig von diesen Drehzahlschaltschwellen ein zweiter Getriebestufen-Datensatz für zwei weitere projektierbare Schaltschwellen (Maximaldrehzahl und Minimaldrehzahl) voreingestellt sein. Die Vorgaben des Maschinenherstellers sind zu beachten.

**Weitere Informationen** siehe Funktionshandbuch "Achsen und Spindeln".

## Beispiele


- Beispiel: Gewindebohren mit G331 / G332 (Seite 244)
- Beispiel: Programmierte Bohrdrehzahl in der aktuellen Getriebestufe ausgeben (Seite 245)
- Beispiel: Anwendung des zweiten Getriebestufen-Datensatzes (Seite 245)
- Beispiel: Keine Drehzahlprogrammierung, Überwachung der Getriebestufe (Seite 246)
- Beispiel: Getriebestufenwechsel nicht möglich, Überwachung der Getriebestufe (Seite 246)
- Beispiel: Programmierung ohne SPOS (Seite 247)


### 2.9.11.2 Beispiel: Gewindebohren mit G331 / G332

| Programmcode | Kommentar |
|---|---|
| N10 SPOS[n]=0 | ; Spindel: Lageregelbetrieb |
| | ; Startposition 0 Grad |
| N20 G0 X0 Y0 Z2 | ; Achsen: Startposition anfahren |

| Programmcode | Kommentar |
|---|---|
| N30 G331 Z-50 K-4 S200 | ; Gewindebohren in Z, |
| | ; Steigung K-4 negativ => |
| | ; Spindeldrehrichtung: Linkslauf, |
| | ; Spindeldrehzahl 200 U/min |
| N40 G332 Z3 K-4 | ; Rückzugsbewegung in Z, |
| | ; Steigung K-4 negativ (Linkslauf), |
| | ; autom. Drehrichtungsumkehr => |
| | ; Spindeldrehrichtung Rechtslauf |
| N50 G1 F1000 X100 Y100 Z100 S300 M3 | ; Spindel im Spindelbetrieb. |

### 2.9.11.3 Beispiel: Programmierte Bohrdrehzahl in der aktuellen Getriebestufe ausgeben

| Programmcode | Kommentar |
|---|---|
| N05 M40 S500 | ; programmierte Spindeldrehzahl: 500 U/min => |
| | ; Getriebestufe 1 (20 bis 1028 U/min) |
| ... | |
| N55 SPOS=0 | ; Spindel positionieren |
| N60 G331 Z-10 K5 S800 | ; Gewindebohren |
| | ; Spindeldrehzahl 800 U/min => Getriebestufe 1 |

Die zur programmierten Spindeldrehzahl `S500` passende Getriebestufe bei `M40` wird aus dem ersten Getriebestufen-Datensatz ermittelt. Die programmierte Bohrdrehzahl `S800` wird in der aktuellen Getriebestufe ausgegeben und ist gegebenenfalls auf die Maximaldrehzahl der Getriebestufe begrenzt. Ein automatischer Getriebestufenwechsel nach erfolgtem `SPOS` ist nicht möglich. Die Voraussetzung für den automatischen Getriebestufenwechsel ist der Drehzahlsteuerbetrieb der Spindel.

**Hinweis**

Soll bei einer Spindeldrehzahl von 800 U/min die Getriebestufe 2 angewählt werden, müssen die Schaltschwellen für die Maximal- und Minimaldrehzahl hierfür in den betreffenden Maschinendaten des zweiten Getriebestufen-Datensatzes projektiert sein (siehe nachfolgende Beispiele).

### 2.9.11.4 Beispiel: Anwendung des zweiten Getriebestufen-Datensatzes

Die Schaltschwellen des zweiten Getriebestufen-Datensatzes für die Maximaldrehzahl und Minimaldrehzahl werden bei `G331`/`G332` und Programmierung eines `S`-Werts für die aktive Masterspindel ausgewertet. Automatischer Getriebestufenwechsel `M40` muss aktiv sein. Die so ermittelte Getriebestufe wird mit der aktiven Getriebestufe verglichen. Besteht zwischen beiden ein Unterschied, wird ein Getriebestufenwechsel ausgeführt.

| Programmcode | Kommentar |
|---|---|
| N05 M40 S500 | ; programmierte Spindeldrehzahl: 500 U/min |
| ... | |

| Programmcode | Kommentar |
|---|---|
| N50 G331 S800 | ; Masterspindel: Getriebestufe 2 wird angewählt |
| N55 SPOS=0 | ; Spindel positionieren |
| N60 G331 Z-10 K5 | ; Gewindebohren |
| | ; Spindel-Beschleunigung aus Getriebestufen-Datensatz 2 |

### 2.9.11.5 Beispiel: Keine Drehzahlprogrammierung, Überwachung der Getriebestufe

Wird bei Anwendung des zweiten Getriebestufen-Datensatzes mit G331 keine Drehzahl programmiert, wird das Gewinde mit der zuletzt programmierten Drehzahl gefertigt. Ein Getriebestufenwechsel erfolgt nicht. In diesem Fall wird jedoch überwacht, dass die zuletzt programmierte Drehzahl im vorgegebenen Drehzahlbereich (Schaltschwellen für die Maximal- und Minimaldrehzahl) der aktiven Getriebestufe liegt. Anderenfalls wird Alarm 16748 angezeigt.

| Programmcode | Kommentar |
|---|---|
| N05 M40 S800 | ; programmierte Spindeldrehzahl: 800 U/min |
| ... | |
| N55 SPOS=0 | ; Spindel positionieren |
| N60 G331 Z-10 K5 | ; Gewindebohren |
| | ; Überwachung der Spindeldrehzahl 800 U/min |
| | ; Getriebestufen 1 ist aktiv |
| | ; Getriebestufe 2 müsste aktiv sein => Alarm 16748 |

### 2.9.11.6 Beispiel: Getriebestufenwechsel nicht möglich, Überwachung der Getriebestufe

Wird bei Anwendung des zweiten Getriebestufen-Datensatzes im G331-Satz zusätzlich zur Geometrie die Spindeldrehzahl programmiert, dann kann, falls die Drehzahl nicht im vorgegebenen Drehzahlbereich (Schaltschwellen für die Maximal- und Minimaldrehzahl) der aktiven Getriebestufe liegt, kein Getriebestufenwechsel durchgeführt werden, weil dann die Bahnbewegung von Spindel und Zustellachse(n) nicht eingehalten werden würde.

Wie im vorhergehenden Beispiel werden im G331-Satz die Drehzahl und die Getriebestufe überwacht und gegebenenfalls der Alarm 16748 angezeigt.

| Programmcode | Kommentar |
|---|---|
| N05 M40 S500 | ; programmierte Spindeldrehzahl: 500 U/min => |
| | ; Getriebestufe 1 |
| ... | |
| N55 SPOS=0 | ; Spindel positionieren |
| N60 G331 Z-10 K5 S800 | ; Gewindebohren |
| | ; Getriebestufenwechsel nicht möglich, |
| | ; Überwachung der Spindeldrehzahl 800 U/min |
| | ; mit Getriebestufen-Datensatz 1: Getriebestufe 2 |
| | ; müsste aktiv sein => Alarm 16748 |

### 2.9.11.7 Beispiel: Programmierung ohne SPOS

| Programmcode | Kommentar |
|---|---|
| N05 M40 S500 | ; programmierte Spindeldrehzahl: 500 U/min => |
| | ; Getriebestufe 1 (20 bis 1028 U/min) |
| ... | |
| N50 G331 S800 | ; Masterspindel: Getriebestufe 2 wird angewählt |
| N60 G331 Z-10 K5 | ; Gewindebohren |
| | ; Spindel-Beschleunigung aus Getriebestufen-Datensatz 2 |

Die Gewindeinterpolation für die Spindel beginnt von der aktuellen Position, die von dem zuvor abgearbeiteten Teileprogrammbereich abhängig ist, z. B. wenn ein Getriebestufenwechsel ausgeführt wurde. Eine Nachbearbeitung des Gewindes ist deshalb ggf. nicht möglich.

**Hinweis**

Es ist darauf zu achten, dass bei der Bearbeitung mit mehreren Spindeln die Bohrspindel auch Masterspindel sein muss. Durch die Programmierung von `SETMS(<Spindelnummer>)` kann die Bohrspindel zur Masterspindel gemacht werden.

## 2.9.12 Gewindebohren mit Ausgleichsfutter

### 2.9.12.1 Gewindebohren mit Ausgleichsfutter und Rückzugsbewegung (G63)

Beim Gewindebohren mit Ausgleichsfutter werden mit dem Befehl `G63` folgenden Verfahrbewegungen ausgeführt:

- `G63`: Gewindebohren in Bohrrichtung bis zum Gewindeendpunkt
- `G63`: Rückzugsbewegung mit programmierter Spindeldrehrichtungsumkehr

**Hinweis**

Nach einem `G63` -Satz wird wieder die zuletzt wirksame Interpolationsart `G0`, `G1`, `G2` aktiv.

### Syntax

```
G63 <Achse> <Drehrichtung> <Drehzahl> <Vorschub>
```

### Bedeutung

| | | |
|---|---|---|
| `G63`: | Gewindebohren mit Ausgleichsfutter | |
| | Wirksamkeit: | satzweise |
| `<Achse>`: | Verfahrweg/position der Geometrieachse (X, Y oder Z) zum Gewindeendpunkt, z. B. Z50 | |

| | |
|---|---|
| <Drehrichtung>: | Spindeldrehrichtung:<br>• M3: Drehrichtung im Uhrzeigersinn, Rechtsgewinde<br>• M4: Drehrichtung gegen den Uhrzeigersinn, Linksgewinde |
| <Drehzahl>: | Maximale während des Gewindebohrens zulässige Spindeldrehzahl, z. B. S100 |
| <Vorschub>: | Vorschub der Bohrachse F, mit F = Spindeldrehzahl * Gewindesteigung |

### Beispiel

Bohren eines M5-Gewindes:

- Gewindesteigung nach Norm: 0,8 mm/U
- Spindeldrehzahl S: 200 U/min
- Vorschub F = 200 U/min * 0,8 mm/U = 160 mm/min.

| **Programmcode** | **Kommentar** |
|---|---|
| N10 G1 X0 Y0 Z2 F1000 S200 M3 | ; Startpunkt anfahren |
| | ; Spindel im Uhrzeigersinn, 200 U/min |
| N20 G63 Z-50 F160 | ; Gewindebohren mit Ausgleichsfutter |
| | ; Bohrtiefe: absolut Z=50mm |
| | ; Vorschub: 160 mm/min |
| N30 G63 Z3 M4 | ; Rückzugsbewegung: absolut Z=3mm |
| | ; Drehrichtungsumkehr |
| | ; Spindel gegen den Uhrzeigersinn, 200 U/min |

## 2.9.13 Fase, Rundung (CHF, CHR, RND, RNDM, FRC, FRCM)

Konturecken innerhalb der aktiven Arbeitsebene können als Rundung oder Fase ausgeführt werden.

Zur Optimierung der Oberflächengüte kann für das Anfasen/Verrunden ein eigener Vorschub programmiert werden. Wird kein Vorschub programmiert, wirkt der normale Bahnvorschub F.

Mit der Funktion "Modales Verrunden" können mehrere Konturecken hintereinander gleichartig verrundet werden.

### Syntax

Konturecke anfasen:
```
G... X... Z... CHR/CHF=<Wert> FRC/FRCM=<Wert>
G... X... Z...
```

Konturecke verrunden:
```
G... X... Z... RND=<Wert> FRC=<Wert>
G... X... Z...
```

Modales Verrunden:

```
G... X... Z... RNDM=<Wert> FRCM=<Wert>
...
RNDM=0
```

**Hinweis**

Die Technologie (Vorschub, Vorschubtyp, M-Befehle ...) für das Anfasen/Verrunden wird abhängig von der Einstellung von Bit 0 im Maschinendatum MD20201 $MC_CHFRND_MODE_MASK (Verhalten Fase/Rundung) entweder vom Vorgänger- oder vom Nachfolgersatz abgeleitet. Empfohlene Einstellung ist die Ableitung aus dem Vorgängersatz (Bit 0 = 1).

## Bedeutung

| | | |
|---|---|---|
| CHF=… : | Konturecke anfasen | |
| | <Wert>: | Länge der Fase (Maßeinheit entsprechend G70/G71) |
| CHR=… : | Konturecke anfasen | |
| | <Wert>: | Breite der Fase in der ursprünglichen Bewegungsrichtung (Maßeinheit entsprechend G70/G71) |
| RND=… : | Konturecke verrunden | |
| | <Wert>: | Radius der Rundung (Maßeinheit entsprechend G70/G71) |
| RNDM=… : | Modales Verrunden (mehrere aufeinanderfolgende Konturecken gleichartig verrunden) | |
| | <Wert>: | Radius der Rundungen (Maßeinheit entsprechend G70/G71) |
| | | Mit RNDM=0 wird das modale Verrunden ausgeschaltet. |
| FRC=… : | Satzweise wirksamer Vorschub für Anfasen/Verrunden | |
| | <Wert>: | Vorschubgeschwindigkeit in mm/min (bei aktivem G94) bzw. mm/Umdr (bei aktivem G95) |
| FRCM=… : | Modal wirksamer Vorschub für Anfasen/Verrunden | |
| | <Wert>: | Vorschubgeschwindigkeit in mm/min (bei aktivem G94) bzw. mm/Umdr (bei aktivem G95) |
| | | Mit FRCM=0 wird der modal wirksame Vorschub für Anfasen/Verrunden ausgeschaltet und der unter F programmierte Vorschub ist aktiv. |

**Hinweis**

**Fase/Rundung zu groß**

Sind die programmierten Werte für Fase (CHF/CHR) oder Rundung (RND/RNDM) für die beteiligten Konturelemente zu groß, werden Fase oder Rundung automatisch angepasst:

1. Falls MD11411 $MN_ENABLE_ALARM_MASK Bit 4 gesetzt ist, wird Alarm 10833 "Fase oder Rundung muss verkürzt werden" ausgegeben (Cancel-Alarm).
2. Die Fase/Rundung wird soweit verkleinert, dass sie in die Konturecke passt. Dabei entsteht mindestens ein Satz ohne Bewegung. An diesem Satz kommt die Bewegung notwendig zum Stehen.

**Hinweis**

**Fase/Rundung nicht möglich**

Keine Fase/Rundung wird eingefügt, wenn:

- keine Geraden- oder Kreiskontur in der Ebene vorhanden ist.
- eine Bewegung außerhalb der Ebene stattfindet.
- ein Wechsel der Ebene vorgenommen wird.
- eine im Maschinendatum festgelegte Anzahl von Sätzen, die keine Informationen zum Verfahren enthalten (z. B. nur Befehlsausgaben), überschritten wird.

**Hinweis**

**FRC/FRCM**

FRC/FRCM wirkt nicht, wenn eine Fase mit G0 verfahren wird; die Programmierung ist entsprechend dem F-Wert ohne Fehlermeldung möglich.

FRC ist nur wirksam, wenn im Satz eine Fase/Rundung programmiert ist bzw. RNDM aktiviert wurde.

FRC überschreibt im aktuellen Satz den F- bzw. FRCM-Wert.

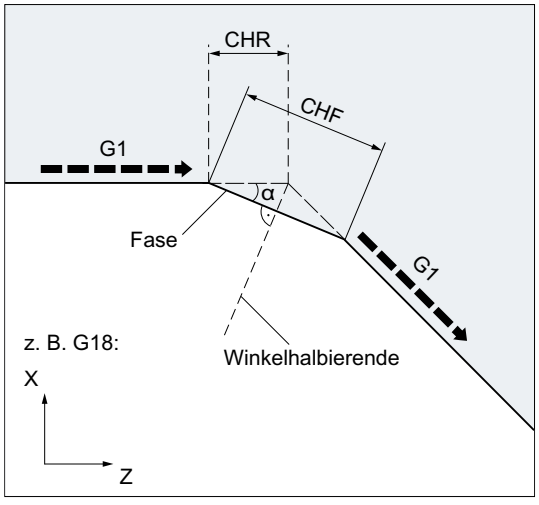
Der unter FRC programmierte Vorschub muss größer Null sein.

FRCM=0 aktiviert für das Anfasen/Verrunden den unter F programmierten Vorschub.

Ist FRCM programmiert, muss äquivalent zu F der FRCM-Wert bei Wechsel G94 ↔ G95 usw. neu programmiert werden. Wird nur F neu programmiert und ist vor dem Wechsel des Vorschubtyps FRCM > 0, dann erfolgt eine Fehlermeldung.

## Beispiele

**Beispiel 1: Anfasen zwischen zwei Geraden**




- MD20201 Bit 0 = 1 (Ableitung aus dem Vorgängersatz)
- G71 ist aktiv.
- Die Breite der Fase in der Bewegungsrichtung (CHR) soll 2 mm, der Vorschub für das Anfasen 100 mm/min betragen.

Die Programmierung kann auf zwei Arten erfolgen:

- Programmierung mit CHR

**Programmcode**

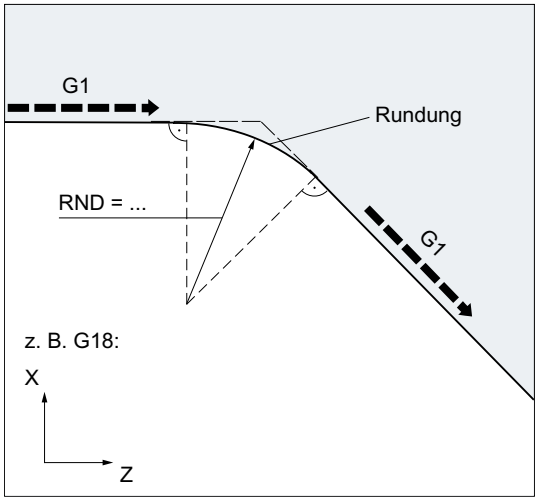
```
...
N30 G1 Z… CHR=2 FRC=100
N40 G1 X…
...
```

- Programmierung mit CHF

**Programmcode**

```
...
N30 G1 Z… CHF=2(cosα*2) FRC=100
N40 G1 X…
...
```

### Beispiel 2: Verrunden zwischen zwei Geraden



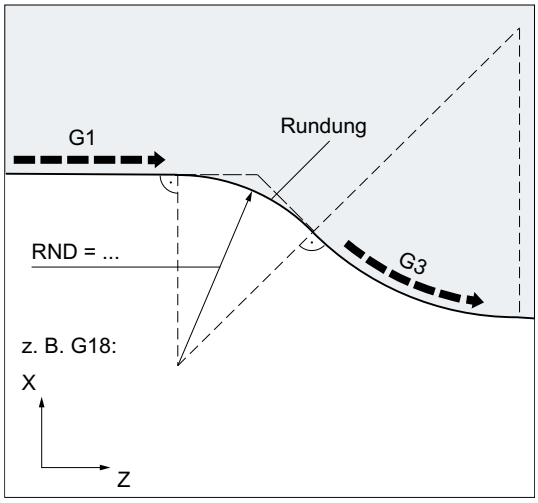

- MD20201 Bit 0 = 1 (Ableitung aus dem Vorgängersatz)
- G71 ist aktiv.
- Der Radius der Rundung soll 2 mm, der Vorschub für das Verrunden 50 mm/min betragen.

**Programmcode**

```
...
N30 G1 Z… RND=2 FRC=50
N40 G1 X…
...
```

### Beispiel 3: Verrunden zwischen Gerade und Kreis

Zwischen Linear- und Kreiskonturen in beliebigen Kombinationen kann durch die Funktion RND mit tangentialem Anschluss ein Kreiskonturelement eingefügt werden.




- MD20201 Bit 0 = 1 (Ableitung aus dem Vorgängersatz)
- G71 ist aktiv.
- Der Radius der Rundung soll 2 mm, der Vorschub für das Verrunden 50 mm/min betragen.

| Programmcode |
|---|
| ... |
| N30 G1 Z… RND=2 FRC=50 |
| N40 G3 X… Z… I… K… |
| ... |

### Beispiel 4: Modales Verrunden zum Entgraten scharfer Werkstückkanten

| Programmcode | Kommentar |
|---|---|
| ... | |
| N30 G1 X… Z… RNDM=2 FRCM=50 | ; Modales Verrunden einschalten. |
| | Radius der Rundung: 2mm |
| | Vorschub für das Verrunden: 50 mm/min |
| N40... | |
| N120 RNDM=0 | ; Modales Verrunden ausschalten. |
| ... | |

### Beispiel 5: Technologie vom Nachfolgesatz oder Vorgängersatz übernehmen

- MD20201 Bit 0 = 0: Ableitung aus dem Nachfolgesatz (Standardeinstellung!)

| Programmcode | Kommentar |
|---|---|
| N10 G0 X0 Y0 G17 F100 G94 | |
| N20 G1 X10 CHF=2 | ; Fase N20-N30 mit F=100 mm/min |
| N30 Y10 CHF=4 | ; Fase N30-N40 mit FRC=200 mm/min |
| N40 X20 CHF=3 FRC=200 | ; Fase N40-N60 mit FRCM=50 mm/min |
| N50 RNDM=2 FRCM=50 | |

| **Programmcode** | **Kommentar** |
|---|---|
| N60 Y20 | ; Modale Rundung N60-N70 mit FRCM=50 mm/min |
| N70 X30 | ; Modale Rundung N70-N80 mit FRCM=50 mm/min |
| N80 Y30 CHF=3 FRC=100 | ; Fase N80-N90 mit FRC=100 mm/min |
| N90 X40 | ; Modale Rundung N90-N100 mit F=100 mm/min (Abwahl FRCM) |
| N100 Y40 FRCM=0 | ; Modale Rundung N100-N120 mit G95 FRC=1 mm/Umdr |
| N110 S1000 M3 | |
| N120 X50 G95 F3 FRC=1 | |
| ... | |
| M02 | |

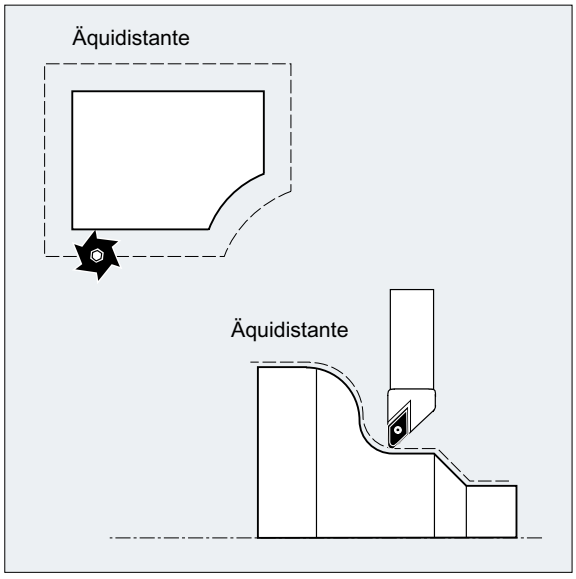
- MD20201 Bit 0 = 1: Ableitung aus dem Vorgängersatz (empfohlenen Einstellung!)

| **Programmcode** | **Kommentar** |
|---|---|
| N10 G0 X0 Y0 G17 F100 G94 | |
| N20 G1 X10 CHF=2 | ; Fase N20-N30 mit F=100 mm/min |
| N30 Y10 CHF=4 FRC=120 | ; Fase N30-N40 mit FRC=120 mm/min |
| N40 X20 CHF=3 FRC=200 | ; Fase N40-N60 mit FRC=200 mm/min |
| N50 RNDM=2 FRCM=50 | |
| N60 Y20 | ; Modale Rundung N60-N70 mit FRCM=50 mm/min |
| N70 X30 | ; Modale Rundung N70-N80 mit FRCM=50 mm/min |
| N80 Y30 CHF=3 FRC=100 | ; Fase N80-N90 mit FRC=100 mm/min |
| N90 X40 | ; Modale Rundung N90-N100 mit FRCM=50 mm/min |
| N100 Y40 FRCM=0 | ; Modale Rundung N100-N120 mit F=100 mm/min |
| N110 S1000 M3 | |
| N120 X50 CHF=4 G95 F3 FRC=1 | ; Fase N120-N130 mit G95 FRC=1 mm/Umdr |
| N130 Y50 | ; Modale Rundung N130-N140 mit F=3 mm/Umdr |
| N140 X60 | |
| ... | |
| M02 | |

# 2.10 Werkzeugradiuskorrekturen

## 2.10.1 Werkzeugradiuskorrektur (G40, G41, G42, OFFN)

Bei eingeschalteter Werkzeugradiuskorrektur (WRK) errechnet die Steuerung automatisch für unterschiedliche Werkzeuge die jeweils äquidistanten Werkzeugwege.




### Syntax

| | |
|---|---|
| G0/G1 X... Y… Z... **G41**/**G42** [**OFFN=<Wert>**] | |
| ... | |
| **G40** X... Y… Z... | |

### Bedeutung

| | |
|---|---|
| G41: | WRK mit Bearbeitungsrichtung **links** von der Kontur einschalten |
| G42: | WRK mit Bearbeitungsrichtung **rechts** von der Kontur einschalten |
| OFFN=<Wert>: | Aufmaß zur programmierten Kontur (Offset Kontur normal) (optional)<br>Z. B. zum Erzeugen von äquidistanten Bahnen für das Grobschlichten. |
| G40: | WRK ausschalten |

**Hinweis**

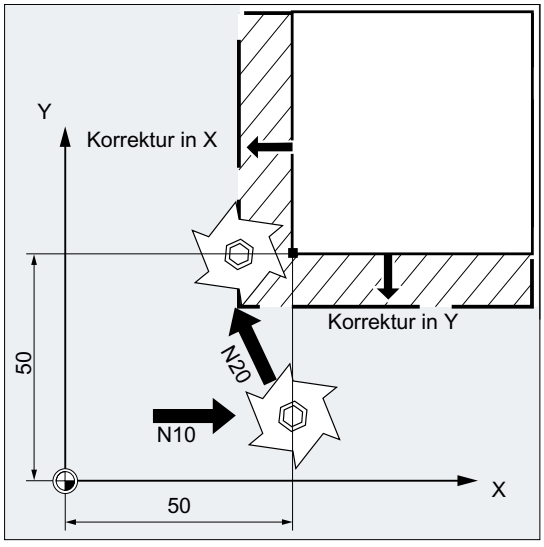
Im NC-Satz mit G40/G41/G42 muss G0 oder G1 aktiv sein und mindestens eine Achse der gewählten Arbeitsebene angegeben werden.

Wird beim Einschalten nur eine Achse angegeben, dann wird die letzte Position der zweiten Achse automatisch ergänzt und in **beiden** Achsen verfahren.

Die beiden Achsen müssen als Geometrieachsen im Kanal aktiv sein. Dies kann durch GEOAX-Programmierung sichergestellt werden.

## Beispiele

### Beispiel 1: Fräsen

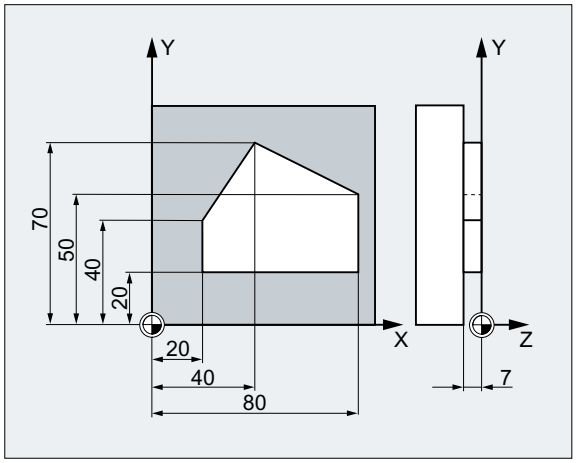



| Programmcode | Kommentar |
|---|---|
| N10 G0 X50 T1 D1 | ; Nur die Werkzeuglängenkorrektur wird eingeschaltet. X50 wird unkorrigiert angefahren. |
| N20 G1 G41 Y50 F200 | ; Die Radiuskorrektur wird eingeschaltet, der Punkt X50/Y50 wird korrigiert angefahren. |
| N30 Y100 | |
| … | |

### Beispiel 2: "Klassische" Vorgehensweise am Beispiel Fräsen

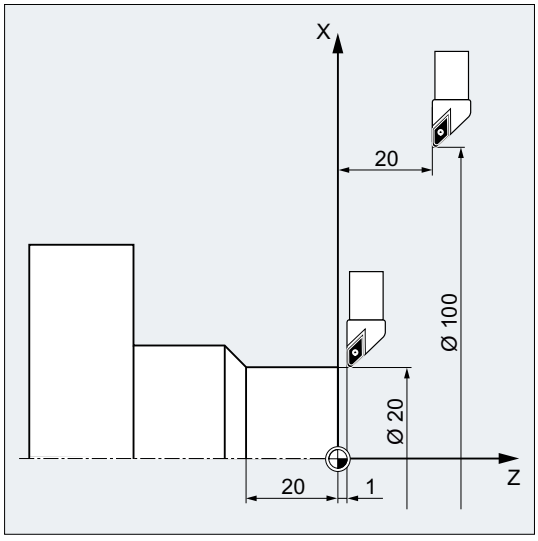
"Klassische" Vorgehensweise:

1. Werkzeugaufruf
2. Werkzeug einwechseln.
3. Arbeitsebene und Werkzeugradiuskorrektur einschalten.



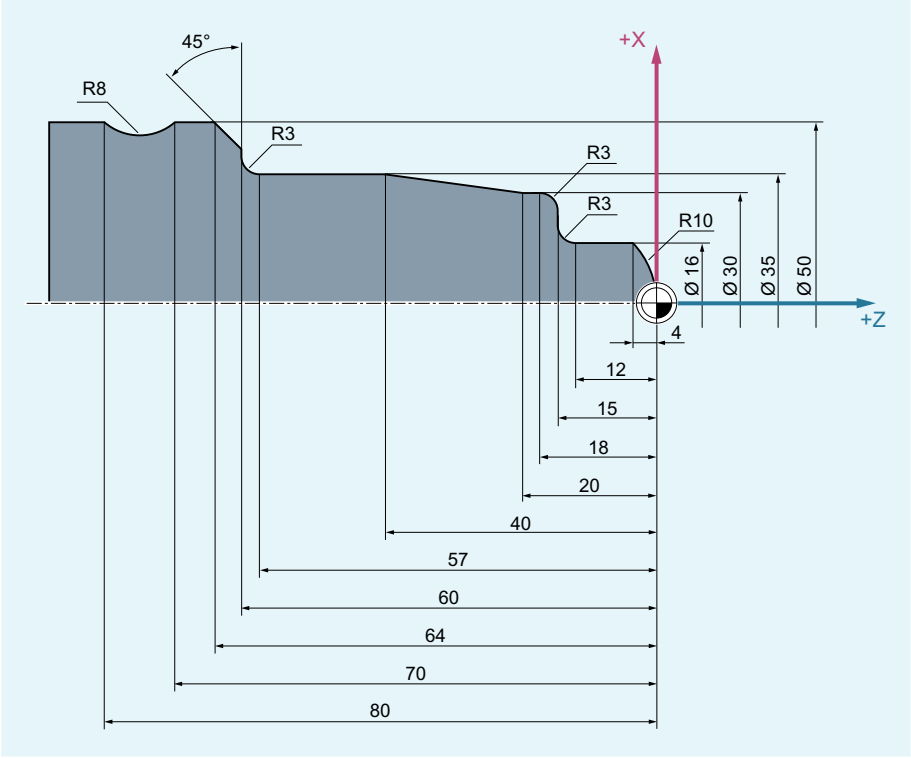

| **Programmcode** | **Kommentar** |
|---|---|
| N10 G0 Z100 | ; Freifahren zum Werkzeugwechsel. |
| N20 G17 T1 M6 | ; Werkzeugwechsel |
| N30 G0 X0 Y0 Z1 M3 S300 D1 | ; Werkzeugkorrekturwerte aufrufen, Längenkorrektur anwählen. |
| N40 Z-7 F500 | ; Werkzeug zustellen. |
| N50 G41 X20 Y20 | ; Werkzeugradiuskorrektur einschalten, Werkzeug arbeitet links von der Kontur. |
| N60 Y40 | ; Kontur fräsen. |
| N70 X40 Y70 | |
| N80 X80 Y50 | |
| N90 Y20 | |
| N100 X20 | |
| N110 G40 G0 Z100 M30 | ; Werkzeug freifahren, Programm-Ende. |

### Beispiel 3: Drehen




| Programmcode | Kommentar |
|---|---|
| … | |
| N20 T1 D1 | ; Nur die Werkzeuglängenkorrektur wird eingeschaltet. |
| N30 G0 X100 Z20 | ; X100 Z20 wird unkorrigiert angefahren. |
| N40 G42 X20 Z1 | ; Die Radiuskorrektur wird eingeschaltet, der Punkt X20/Z1 wird korrigiert angefahren. |
| N50 G1 Z-20 F0.2 | |
| … | |

### Beispiel 4: Drehen




| **Programmcode** | **Kommentar** |
|---|---|
| N5 G0 G53 X280 Z380 D0 | ; Startpunkt |
| N10 TRANS X0 Z250 | ; Nullpunktverschiebung |
| N15 LIMS=4000 | ; Drehzahlbegrenzung (G96) |
| N20 G96 S250 M3 | ; konstanten Vorschub anwählen |
| N25 G90 T1 D1 M8 | ; Werkzeuganwahl und Korrektur anwählen |
| N30 G0 G42 X-1.5 Z1 | ; Werkzeug anstellen mit Werkzeugradius-korrektur |
| N35 G1 X0 Z0 F0.25 | |
| N40 G3 X16 Z-4 I0 K-10 | ; Radius 10 drehen |
| N45 G1 Z-12 | |
| N50 G2 X22 Z-15 CR=3 | ; Radius 3 drehen |
| N55 G1 X24 | |
| N60 G3 X30 Z-18 I0 K-3 | ; Radius 3 drehen |
| N65 G1 Z-20 | |
| N70 X35 Z-40 | |
| N75 Z-57 | |
| N80 G2 X41 Z-60 CR=3 | ; Radius 3 drehen |
| N85 G1 X46 | |
| N90 X52 Z-63 | |
| N95 G0 G40 G97 X100 Z50 M9 | ; Werkzeugradiuskorrektur abwählen und Werkzeugwechselpunkt anfahren |

| Programmcode | Kommentar |
|---|---|
| N100 T2 D2 | ; Werkzeug aufrufen und Korrektur anwählen |
| N105 G96 S210 M3 | ; konstante Schnittgeschwindigkeit anwählen |
| N110 G0 G42 X50 Z-60 M8 | ; Werkzeug anstellen mit Werkzeugradiuskorrektur |
| N115 G1 Z-70 F0.12 | ; Durchmesser 50 drehen |
| N120 G2 X50 Z-80 I6.245 K-5 | ; Radius 8 drehen |
| N125 G0 G40 X100 Z50 M9 | ; Werkzeug abheben und Werkzeugradiuskorrektur abwählen |
| N130 G0 G53 X280 Z380 D0 M5 | ; Werkzeugwechselpunkt verfahren |
| N135 M30 | ; Programm-Ende |

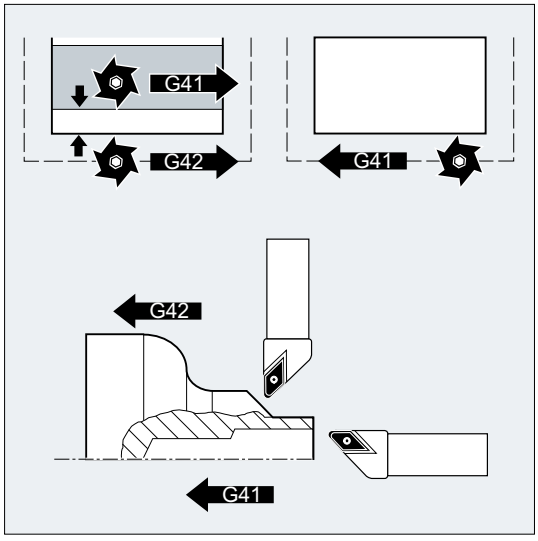
## Weitere Informationen

Für die Berechnung der Werkzeugwege benötigt die Steuerung folgende Informationen:

- Werkzeugnr. (T...), Schneidennr. (D...)
- Bearbeitungsrichtung (G41/G42)
- Arbeitsebene (G17/G18/G19)

### Werkzeugnr. (T...), Schneidennr. (D...)

Aus den Fräserradien bzw. Schneidenradien und Angaben zur Schneidenlage wird der Abstand zwischen Werkzeugbahn und Werkstückkontur berechnet.



### Bearbeitungsrichtung (G41/G42)

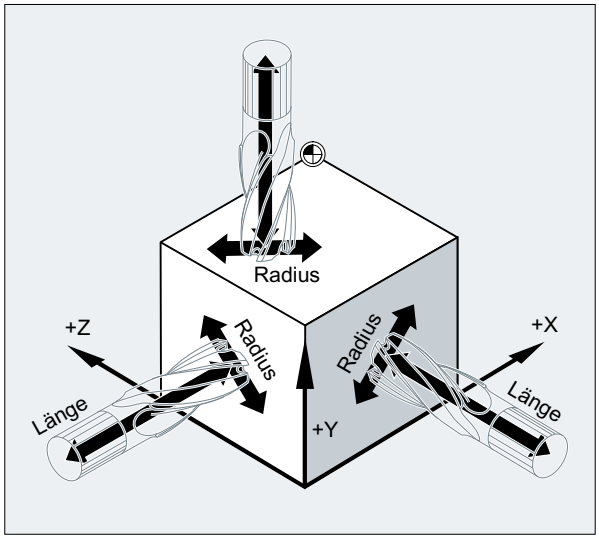
Hieraus erkennt die Steuerung die Richtung, in die die Werkzeugbahn verschoben werden soll.

**Hinweis**

Ein negativer Korrekturwert ist gleichbedeutend mit einem Wechsel der Korrekturseite (`G41` ↔ `G42`).

### Arbeitsebene (G17/G18/G19)

Hieraus erkennt die Steuerung die Ebene und damit die Achsrichtungen, in denen korrigiert wird.




Beispiel: Fräswerkzeug

| **Programmcode** | **Kommentar** |
|---|---|
| ... | |
| N10 G17 G41 … | ; Die Werkzeugradiuskorrektur erfolgt in der X/Y-Ebene, die Werkzeuglängenkorrektur in Z-Richtung. |
| ... | |

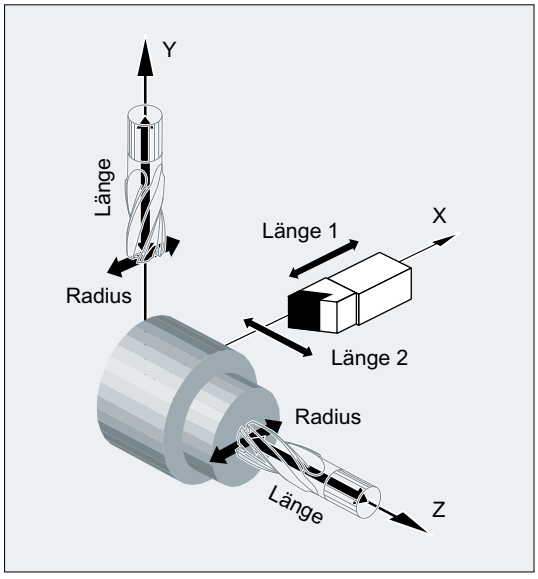
**Hinweis**

Bei 2-Achs-Maschinen ist die Werkzeugradiuskorrektur nur in "echten" Ebenen möglich, in der Regel bei `G18`.

### Werkzeuglängenkorrektur

Der bei Werkzeuganwahl der Durchmesserachse zugeordnete Verschleißparameter kann über ein Maschinendatum als Durchmesserwert definiert werden. Bei einem nachfolgenden Ebenenwechsel wird diese Zuordnung nicht automatisch verändert. Dafür muss das Werkzeug nach Ebenenwechsel neu angewählt werden.

Drehen:




Mit NORM und KONT kann die Werkzeugbahn beim Ein- und Ausschalten des Korrekturbetriebs festgelegt werden (siehe "Kontur anfahren und verlassen (NORM, KONT, KONTC, KONTT) (Seite 263)").

### Schnittpunkt

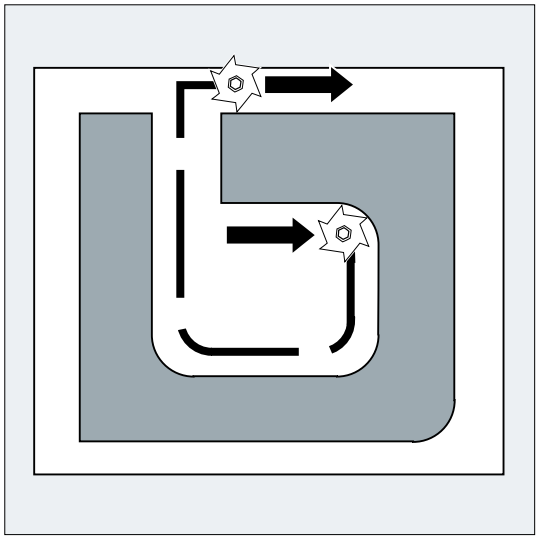
Die Auswahl des Schnittpunkts erfolgt über das Settingdatum:

SD42496 $SC_CUTCOM_CLSD_CONT (Verhalten der Werkzeugradiuskorrektur bei geschlossener Kontur)

| Wert | Bedeutung |
|---|---|
| FALSE | Ergeben sich bei einer (nahezu) geschlossenen Kontur, die aus zwei aufeinanderfolgenden Kreissätzen oder einem Kreis- und einem Linearsatz besteht, bei Korrektur an der Innenseite zwei Schnittpunkte, so wird entsprechend dem Standardverfahren der Schnittpunkt gewählt, der auf der ersten Teilkontur näher am Satzende liegt.<br>Eine Kontur wird dann als (nahezu) geschlossen betrachtet, wenn der Abstand zwischen dem Startpunkt des ersten Satzes und dem Endpunkt des zweiten Satzes kleiner ist als 10 % des wirksamen Korrekturradius, aber nicht größer als 1000 Weginkremente (entspricht 1 mm bei 3 Nachkommastellen). |
| TRUE | In der gleichen Situation wie oben beschrieben wird der Schnittpunkt gewählt, der auf der ersten Teilkontur näher am Satzanfang liegt. |

### Wechsel der Korrekturrichtung (G41 ↔ G42)

Ein Wechsel der Korrekturrichtung (G41 ↔ G42) kann ohne zwischengeschaltetes G40 programmiert werden.



**Wechsel der Arbeitsebene**

Ein Wechsel der Arbeitsebene (G17/G18/G19) ist bei eingeschaltetem G41/G42 **nicht** möglich.

**Wechsel des Werkzeugkorrekturdatensatzes (D…)**

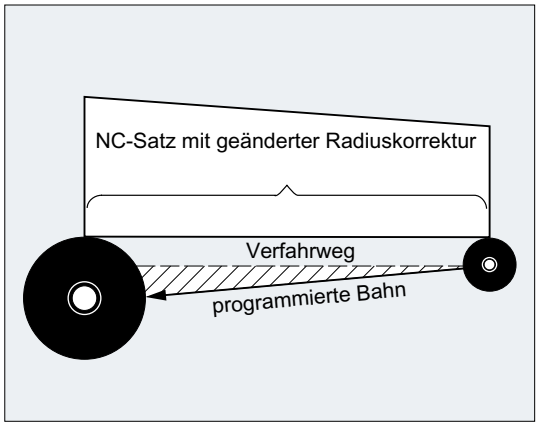
Der Werkzeugkorrekturdatensatz kann im Korrekturbetrieb gewechselt werden.

Ein veränderter Werkzeugradius gilt bereits ab dem Satz, in dem die neue D-Nummer steht.

**Hinweis**

Die Radiusänderung bzw. Ausgleichsbewegung erstreckt sich über den gesamten Satz und erreicht erst im programmierten Endpunkt den neuen äquidistanten Abstand.

Bei Linearbewegungen fährt das Werkzeug auf einer schräg liegenden Bahn zwischen Anfangs- und Endpunkt:




Bei Kreisinterpolationen entstehen Spiralbewegungen.

#### Änderung des Werkzeugradius

Die Änderung kann z. B. über Systemvariablen erfolgen. Für den Ablauf gilt das Gleiche wie beim Wechsel des Werkzeugkorrekturdatensatzes (D...).

**Hinweis**

Die geänderten Werte werden erst nach erneuter T- oder D-Programmierung wirksam. Die Änderung gilt erst im nächsten Satz.

#### Korrekturbetrieb

Der Korrekturbetrieb darf nur von einer bestimmten Anzahl aufeinanderfolgender Sätze oder M-Befehle unterbrochen werden, die keine Fahrbefehle bzw. Wegangaben in der Korrekturebene enthalten.

**Hinweis**

Die Anzahl aufeinanderfolgender Sätze oder M-Befehle ist über ein Maschinendatum einstellbar (siehe Angaben des Maschinenherstellers!).

**Hinweis**

Ein Satz mit Bahnweg Null zählt ebenfalls als Unterbrechung!

## 2.10.2 Kontur anfahren und verlassen (NORM, KONT, KONTC, KONTT)

#### Vorraussetzung

Die Befehle KONTC und KONTT stehen nur zur Verfügung, wenn in der Steuerung die Option "Polynom-Interpolation" freigegeben ist.

#### Funktion

Mit den Befehlen NORM, KONT, KONTC oder KONTT kann bei eingeschalteter Werkzeugradiuskorrektur (G41/G42) der An- und Abfahrweg des Werkzeugs an den gewünschten Konturverlauf oder an die Rohteilform angepasst werden.

Mit KONTC oder KONTT werden die Stetigkeitsbedingungen in allen drei Achsen eingehalten. Damit wird es zulässig, gleichzeitig eine Wegkomponente senkrecht zur Korrekturebene zu programmieren.

### Syntax

| | |
|---|---|
| G41/G42 **NORM**/**KONT**/**KONTC**/**KONTT** X... Y... Z... | |
| ... | |
| G40 X... Y... Z... | |

## Bedeutung

| | |
|---|---|
| NORM: | Direktes An-/Abfahren auf einer Geraden einschalten<br>Das Werkzeug wird senkrecht zum Konturpunkt ausgerichtet. |
| KONT: | An-/Abfahren mit Umfahren des Anfangs-/Endpunkts nach programmiertem Eckenverhalten G450 bzw. G451 einschalten |
| KONTC: | Krümmungsstetiges An-/Abfahren einschalten |
| KONTT: | Tangentenstetiges An-/Abfahren einschalten |

### Hinweis

Als Original-An-/Abfahrsätze für KONTC und KONTT sind nur G1-Sätze zulässig. Diese werden von der Steuerung durch Polynome für die entsprechende Anfahr-/Abfahrbahn ersetzt.

## Randbedingungen

KONTT und KONTC stehen bei den 3D-Varianten der Werkzeugradiuskorrektur (CUT3DC, CUT3DCC, CUT3DF) nicht zur Verfügung. Werden sie dennoch programmiert, wird steuerungsintern ohne Fehlermeldung auf NORM umgeschaltet.

## Beispiel

### KONTC

In der Kreismitte beginnend wird an den Vollkreis angefahren. Dabei sind im Satzendpunkt des Anfahrsatzes Richtung und Krümmungsradius gleich den Werten des folgenden Kreises. In den beiden An-/Abfahrsätzen wird gleichzeitig in Z-Richtung zugestellt. Das folgende Bild zeigt die senkrechte Projektion der Werkzeugbahnbahn:

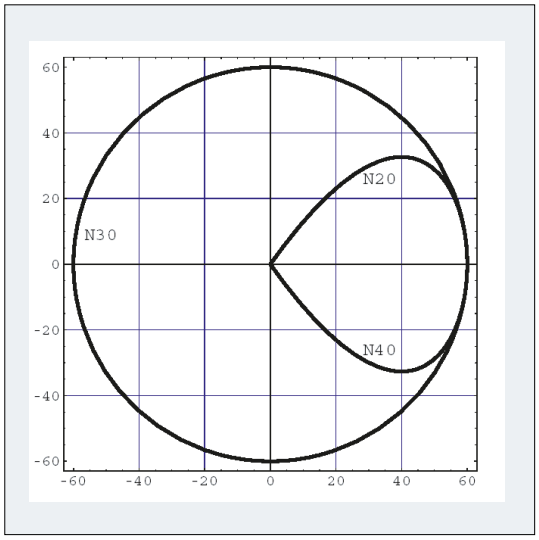



Bild 2-14 Senkrechte Projektion

Das zugehörige NC-Programmsegment sieht folgendermaßen aus:

| **Programmcode** | **Kommentar** |
|---|---|
| `$TC_DP1[1,1]=121` | `; Fräser` |
| `$TC_DP6[1,1]=10` | `; Radius 10 mm` |
| `N10 G1 X0 Y0 Z60 G64 T1 D1 F10000` | |
| `N20 G41 KONTC X70 Y0 Z0` | `; Anfahren` |
| `N30 G2 I-70` | `; Vollkreis` |
| `N40 G40 G1 X0 Y0 Z60` | `; Abfahren` |
| `N50 M30` | |

Gleichzeitig zur Anpassung der Krümmung an die Kreisbahn des Vollkreises wird von Z60 auf die Ebene des Kreises Z0 verfahren:

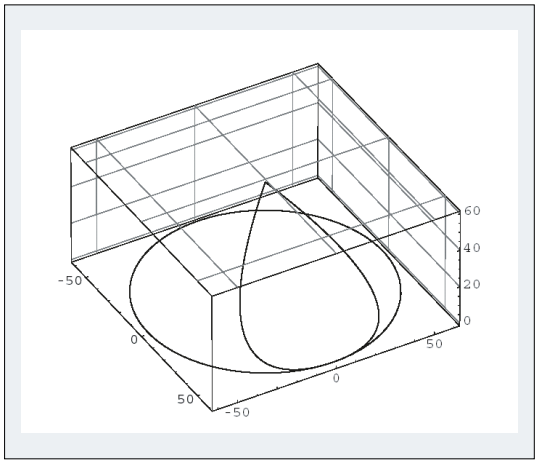
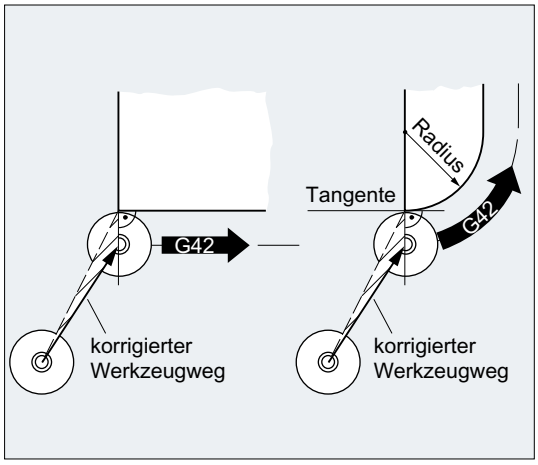



Bild 2-15 Räumliche Darstellung

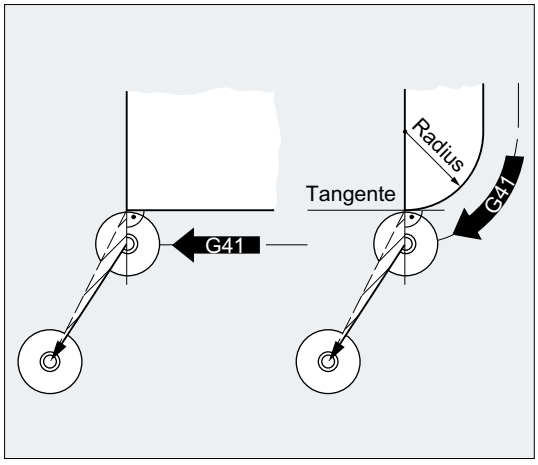
## Weitere Informationen

### An-/Abfahren mit NORM

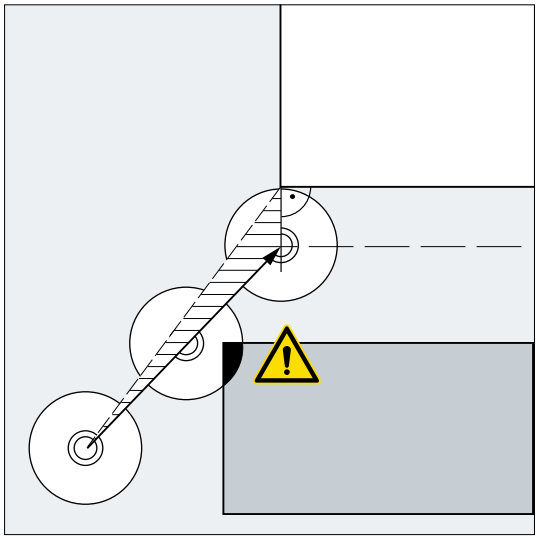
1. Anfahren:
   Bei eingeschaltetem `NORM` fährt das Werkzeug direkt auf einer Geraden auf die korrigierte Startposition (unabhängig von dem durch die programmierte Fahrbewegung vorgegebenen Anfahrwinkel) und wird senkrecht zur Bahntangente im Anfangspunkt ausgerichtet:




2. Abfahren:
   Das Werkzeug steht in senkrechter Position zum letzten korrigierten Bahn-Endpunkt und fährt dann (unabhängig von dem durch die programmierte Fahrbewegung vorgegebenen Anfahrwinkel) direkt auf einer Geraden zur nächsten, unkorrigierten Position, z. B. zum Werkzeugwechselpunkt:



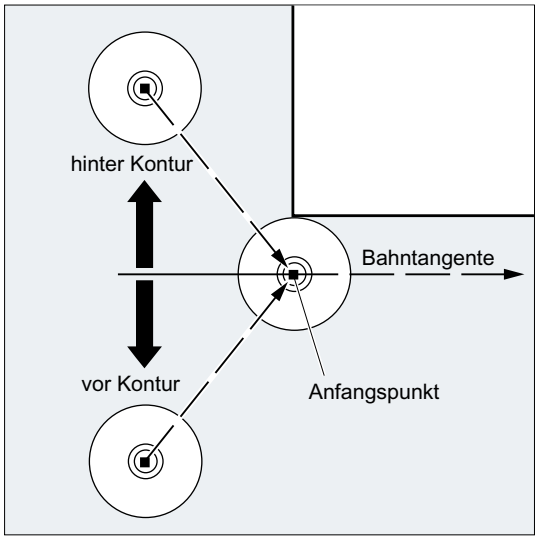

Veränderte An-/Abfahrwinkel stellen eine Kollisionsgefahr dar:



| ACHTUNG |
| --- |
| **Kollisionsgefahr**<br>Um eventuelle Kollisionen zu vermeiden, müssen veränderte An-/Abfahrwinkel bei der Programmierung berücksichtigt werden. |

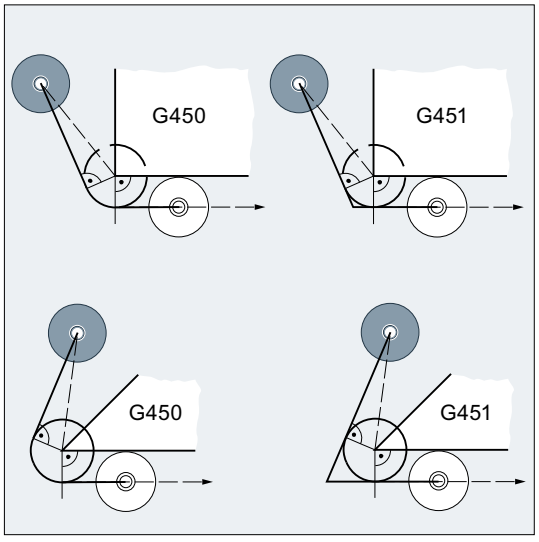
### An-/Abfahren mit KONT

Vor dem Anfahren kann sich das Werkzeug **vor** oder **hinter** der Kontur befinden. Als Trennlinie gilt dabei die Bahntangente im Anfangspunkt:

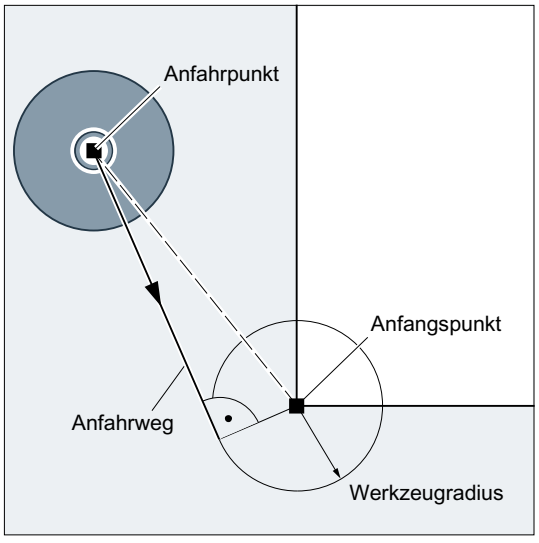


Entsprechend sind beim An-/Abfahren mit KONT zwei Fälle zu unterscheiden:

1. Das Werkzeug befindet sich vor der Kontur.
   → An-/Abfahrstrategie wie bei NORM.
2. Das Werkzeug befindet sich hinter der Kontur
   - Anfahren:
     Das Werkzeug umfährt den Anfangspunkt je nach programmiertem Eckenverhalten (G450/G451) auf einer Kreisbahn oder über den Schnittpunkt der Äquidistanten.
     Die Befehle G450/G451 gelten für den Übergang vom aktuellen Satz zum nächsten Satz:




In beiden Fällen (G450/G451) wird folgender Anfahrweg erzeugt:




Vom unkorrigierten Anfahrpunkt wird eine Gerade gezogen, die einen Kreis mit Kreisradius = Werkzeugradius tangiert. Der Kreismittelpunkt liegt im Anfangspunkt.

   - Abfahren:
     Für das Abfahren gilt, in umgekehrter Reihenfolge, das Gleiche wie für das Anfahren.

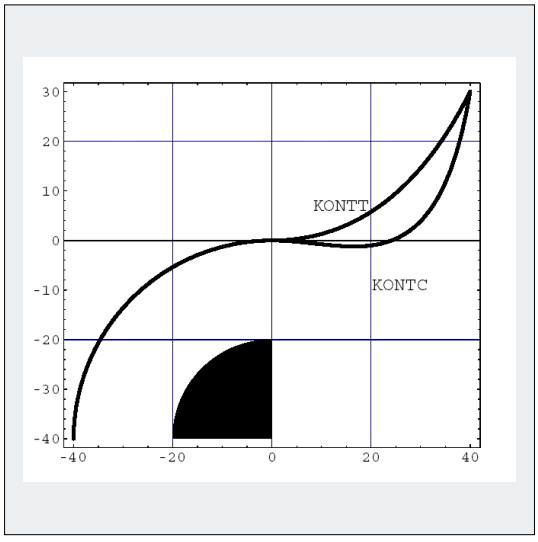
### An-/Abfahren mit KONTC

Der Konturpunkt wird krümmungsstetig angefahren/verlassen. Am Konturpunkt tritt kein Beschleunigungssprung auf. Die Bahn vom Ausgangspunkt zum Konturpunkt wird als Polynom interpoliert.

### An-/Abfahren mit KONTT

Der Konturpunkt wird tangentenstetig angefahren/verlassen. Am Konturpunkt kann ein Beschleunigungssprung auftreten. Die Bahn vom Ausgangspunkt zum Konturpunkt wird als Polynom interpoliert.

### Unterschied zwischen KONTC und KONTT

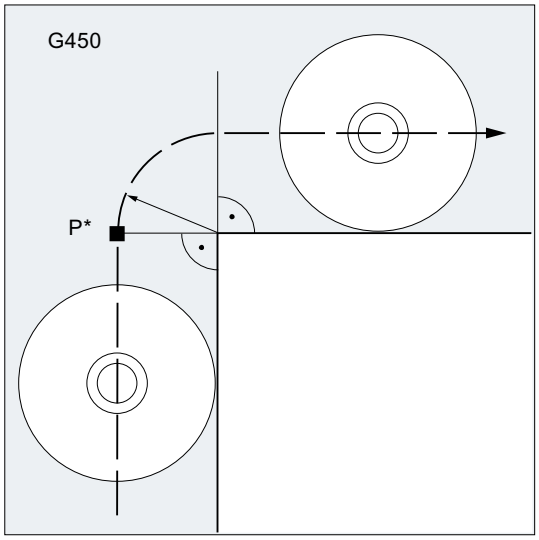



In diesem Bild ist das unterschiedliche An-/Abfahrverhalten bei `KONTT` und `KONTC` dargestellt. Ein Kreis mit dem Radius 20 mm um den Mittelpunkt bei X0 Y-40 wird mit einem Werkzeug mit 20 mm Radius an der Außenseite korrigiert. Es ergibt sich deshalb eine kreisförmige Bewegung des Werkzeugmittelpunkts mit dem Radius 40 mm. Der Endpunkt des Abfahrsatzes liegt bei X40 Y30. Der Übergang zwischen dem Kreissatz und dem Abfahrsatz liegt im Nullpunkt. Wegen der verlangten Krümmungsstetigkeit bei `KONTC` führt der Abfahrsatz zunächst eine Bewegung mit negativer Y-Komponente aus. Dies wird häufig unerwünscht sein. Der Abfahrsatz mit `KONTT` zeigt dieses Verhalten nicht. Allerdings tritt in diesem Fall am Satzübergang ein Beschleunigungssprung auf.

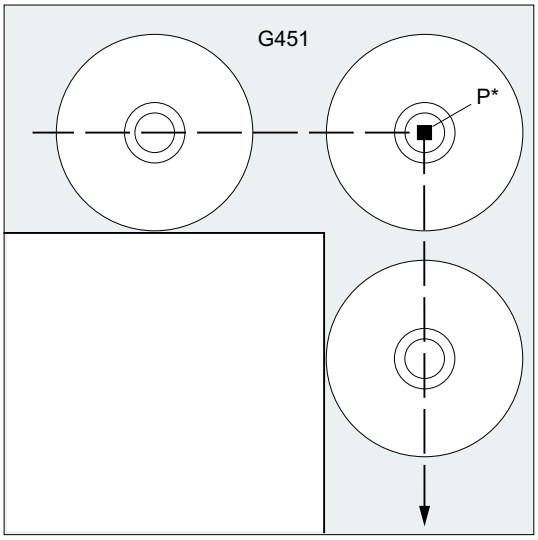
Ist der `KONTT`- bzw. `KONTC`-Satz nicht der Ab- sondern der Anfahrsatz, ergibt sich exakt die gleiche Kontur, die lediglich in umgekehrter Richtung durchlaufen wird.

## 2.10.3 Korrektur an den Außenecken (G450, G451, DISC)

Mit dem Befehl `G450` bzw. `G451` wird bei eingeschalteter Werkzeugradiuskorrektur (`G41`/`G42`) der Verlauf der korrigierten Werkzeugbahn beim Umfahren von Außenecken festgelegt:




Mit `G450` umfährt der Werkzeugmittelpunkt die Werkstückecke auf einem Kreisbogen mit Werkzeugradius.




Mit `G451` fährt der Werkzeugmittelpunkt den Schnittpunkt der beiden Äquidistanten an, die im Abstand Werkzeugradius zur programmierten Kontur liegen. `G451` gilt nur für Geraden und Kreise.

**Hinweis**

Mit `G450`/`G451` wird auch der Anfahrweg bei aktivem `KONT` und Anfahrpunkt hinter der Kontur festgelegt (siehe "Kontur anfahren und verlassen (NORM, KONT, KONTC, KONTT) (Seite 263)").

Mit dem Befehl `DISC` können die Übergangskreise bei `G450` verzerrt und damit scharfe Konturecken hergestellt werden.

### Syntax

```
G450 [DISC=<Wert>]
G451
```

## Bedeutung

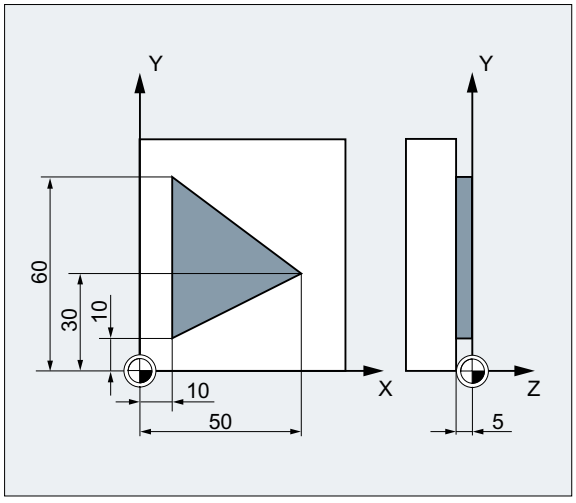
| | | | | |
|---|---|---|---|---|
| G450: | Mit G450 werden Werkstückecken auf einer Kreisbahn umfahren. | | | |
| DISC: | Flexible Programmierung der Kreisbahn bei G450 (optional) | | | |
| | <Wert>: | Typ: | INT | |
| | | Wertebereich: | 0, 1, 2, ... 100 | |
| | | Bedeutung: | 0 | Übergangskreis |
| | | | 100 | Schnittpunkt der Äquidistanten (theoretischer Wert) |
| G451: | Mit G451 wird bei Werkstückecken der Schnittpunkt der beiden Äquidistanten angefahren. Das Werkzeug schneidet in der Werkstückecke frei. | | | |

### Hinweis

DISC wirkt nur mit Aufruf von G450, kann allerdings in einem vorherigen Satz ohne G450 programmiert werden. Beide Befehle sind modal wirksam.

## Beispiel

Im folgenden Beispiel wird bei allen Außenecken ein Übergangsradius eingefügt (entsprechend der Programmierung des Eckenverhaltens in Satz N30). Hierdurch vermeidet man, dass das Werkzeug zum Richtungswechsel stehen bleiben muss und freischneidet.




| **Programmcode** | **Kommentar** |
|---|---|
| N10 G17 T1 G0 X35 Y0 Z0 F500 | ; Startbedingungen |
| N20 G1 Z-5 | ; Werkzeug zustellen. |
| N30 G41 KONT **G450** X10 Y10 | ; WRK mit An-/Abfahrmodus KONT und **Eckenverhalten G450 einschalten.** |
| N40 Y60 | ; Fräsen der Kontur. |
| N50 X50 Y30 | |
| N60 X10 Y10 | |

| Programmcode | Kommentar |
|---|---|
| N80 G40 X-20 Y50 | ; Korrekturbetrieb ausschalten, Wegfahren auf Übergangskreis. |
| N90 G0 Y100 | |
| N100 X200 M30 | |

## Weitere Informationen

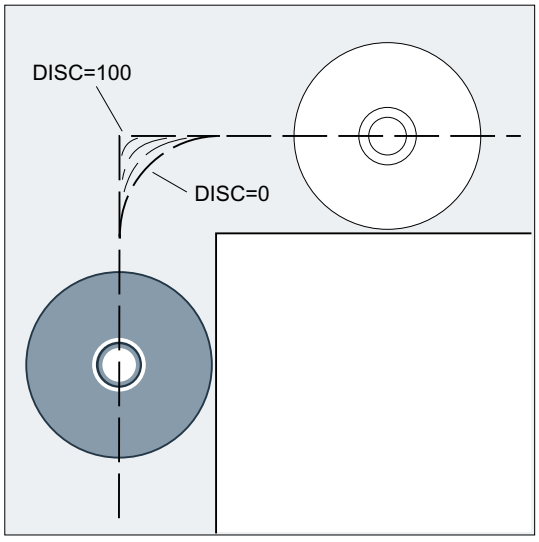
### G450/G451

Im Zwischenpunkt P* führt die Steuerung Anweisungen durch, wie z. B. Zustellbewegungen oder Schaltfunktionen. Diese Anweisungen werden in Sätzen programmiert, die zwischen den beiden Sätzen liegen, die die Ecke bilden.

Der Übergangskreis bei `G450` gehört datentechnisch zum anschließenden Fahrbefehl.

### DISC

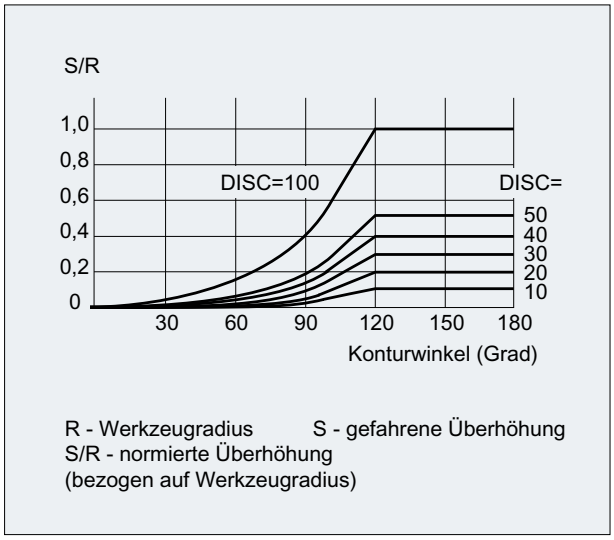
Bei Angabe von `DISC`-Werten größer 0 werden Zwischenkreise überhöht dargestellt, hierbei entstehen Übergangsellipsen bzw. Parabeln oder Hyperbeln:




Über Maschinendatum kann ein oberer Grenzwert festgelegt werden, in der Regel DISC=50.

### Fahrverhalten

Bei eingeschaltetem `G450` hebt das Werkzeug bei spitzen Konturwinkeln und hohen `DISC`-Werten an den Ecken von der Kontur ab. Bei Konturwinkel ab 120° wird die Kontur gleichmäßig umfahren:

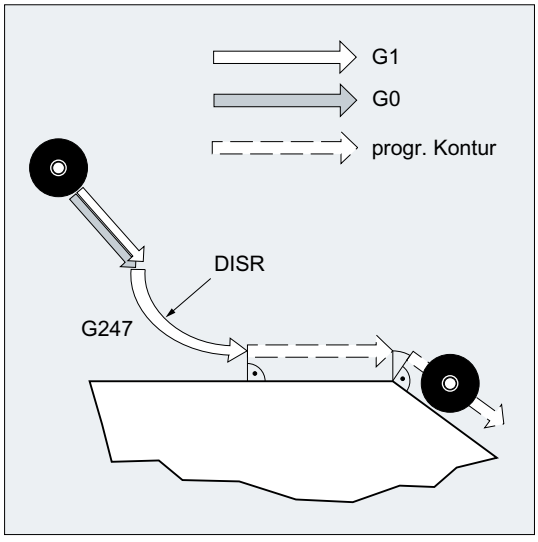



Bei eingeschaltetem `G451` können bei spitzen Konturwinkeln durch Abhebebewegungen überflüssige Leerwege des Werkzeugs entstehen. Über Maschinendatum lässt sich festlegen, dass in solchen Fällen automatisch auf Übergangskreis umgeschaltet wird.

## 2.10.4 Weiches An- und Abfahren

### 2.10.4.1 An- und Abfahren (G140 bis G143, G147, G148, G247, G248, G347, G348, G340, G341, DISR, DISCL, DISRP, FAD, PM, PR)

Die Funktion "Weiches An- und Abfahren (WAB)" dient dazu, im Startpunkt einer Kontur unabhängig von der Lage des Ausgangspunkts tangential anzufahren.




Die Funktion wird vorwiegend in Verbindung mit der Werkzeugradiuskorrektur eingesetzt.

Bei der Aktivierung der Funktion übernimmt die Steuerung die Aufgabe, Zwischenpunkte so zu berechnen, dass der Übergang in den Folgesatz (bzw. der Übergang vom Vorgängersatz beim Abfahren) entsprechend den angegebenen Parametern erfolgt.

Die Anfahrbewegung besteht aus maximal 4 Teilbewegungen. Der Startpunkt der Bewegung wird im Folgenden mit $P_0$, der Endpunkt mit $P_4$ bezeichnet. Dazwischen können maximal drei Zwischenpunkte $P_1$, $P_2$ und $P_3$ liegen. Die Punkte $P_0$, $P_3$ und $P_4$ sind immer definiert. Die Zwischenpunkte $P_1$ und $P_2$ können ja nach Parametrierung und geometrischen Verhältnissen entfallen. Beim Abfahren werden die Punkte in der umgekehrten Reihenfolge durchlaufen, d. h. beginnend mit $P_4$ und endend mit $P_0$.

## Syntax

**Weiches Anfahren:**

- mit einer Geraden:
  ```
  G147 G340/G341 ... DISR=..., DISCL=..., DISRP=... FAD=...
  ```
- mit einem Viertelkreis/Halbkreis:
  ```
  G247/G347 G340/G341 G140/G141/G142/G143 ... DISR=... DISCL=...
  DISRP=... FAD=...
  ```

**Weiches Abfahren:**

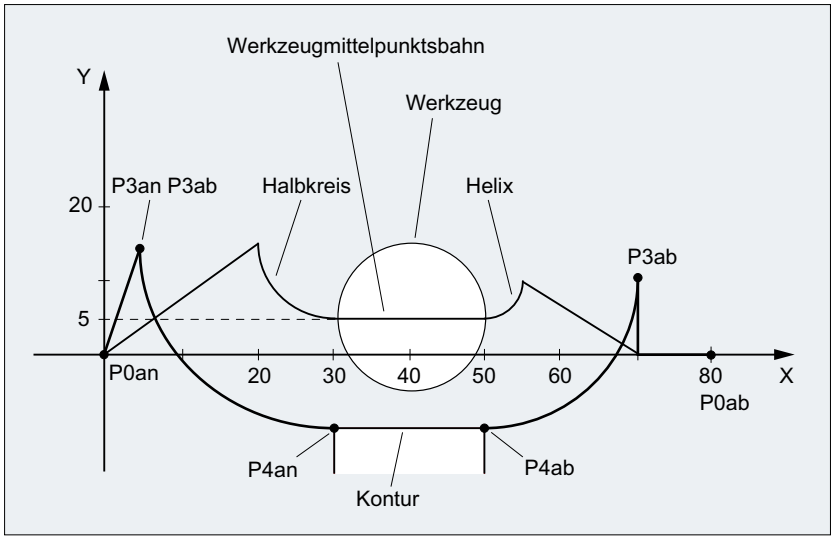
- mit einer Geraden:
  ```
  G148 G340/G341 ... DISR=..., DISCL=..., DISRP=... FAD=...
  ```
- mit einem Viertelkreis/Halbkreis:
  ```
  G248/G348 G340/G341 G140/G141/G142/G143 ... DISR=... DISCL=...
  DISRP=... FAD=...
  ```

## Bedeutung

| | |
|---|---|
| G147: | Anfahren mit einer Geraden |
| G148: | Abfahren mit einer Geraden |
| G247: | Anfahren mit einem Viertelkreis |
| G248: | Abfahren mit einem Viertelkreis |
| G347: | Anfahren mit einem Halbkreis |
| G348: | Abfahren mit einem Halbkreis |
| G340: | An- und Abfahren räumlich (Grundeinstellung) |
| G341: | An- und Abfahren in der Ebene |
| G140: | An- und Abfahrrichtung abhängig von der aktuellen Korrekturseite (Grundeinstellung) |
| G141: | Anfahren von links bzw. Abfahren nach links |
| G142: | Anfahren von rechts bzw. Abfahren nach rechts |
| G143: | An- bzw. Abfahrrichtung abhängig von der relativen Lage von Start- bzw. Endpunkt zur Tangentenrichtung |

| | |
|---|---|
| `DISR=...`: | 1. Beim An- und Abfahren mit Geraden (G147/G148): Abstand der Fräserkante vom Startpunkt der Kontur<br>2. Beim An- und Abfahren mit Kreisen (G247, G347/G248, G348): Radius der Werkzeugmittelpunktsbahn<br>**Achtung:**<br>Bei REPOS mit einem Halbkreis bezeichnet DISR den Kreisdurchmesser. |
| `DISCL=...`: | Abstand des Endpunkts der schnellen Zustellbewegung von der Bearbeitungsebene<br>DISCL=AC(...) Angabe der absoluten Lage des Endpunkts der schnellen Zustellbewegung |
| `DISCL=AC(...)`: | Angabe der absoluten Lage des Endpunkts der schnellen Zustellbewegung |
| `DISRP`: | Abstand des Punkts P1 (Rückzugsebene) von der Bearbeitungsebene |
| `DISRP=AC(...)`: | Angabe der absoluten Lage des Punkts P1 |
| `FAD=...`: | Geschwindigkeit der langsamen Zustellbewegung<br>Der programmierte Wert wirkt entsprechend dem aktiven Vorschubtyp (G-Gruppe 15). |
| `FAD=PM(...)`: | Der programmierte Wert wird unabhängig vom aktiven Vorschubtyp als Linearvorschub (wie G94) interpretiert. |
| `FAD=PR(...)`: | Der programmierte Wert wird unabhängig vom aktiven Vorschubtyp als Umdrehungsvorschub (wie G95) interpretiert. |

## Beispiel




- Weiches Anfahren (Satz N20 aktiviert)
- Anfahrbewegung mit Viertelkreis (G247)
- Anfahrrichtung nicht programmiert, es wirkt G140, d. h. WRK ist aktiv (G41)
- Konturoffset OFFN=5 (N10)

- Aktueller Werkzeugradius=10, damit ist der effektive Korrekturradius für WRK=15, der Radius der WAB-Kontur=25, so dass Radius der Werkzeugmittelpunktsbahn gleich DISR=10 wird
- Endpunkt des Kreises ergibt sich aus N30, da in N20 nur Z-Position programmiert ist
- Zustellbewegung
  - Von Z20 nach Z7 (DISCL=AC(7)) im Eilgang.
  - Anschließend nach Z0 mit FAD=200.
  - Anfahrkreis in X-Y-Ebene und Folgesätze mit F1500 (damit diese Geschwindigkeit in den Folgesätzen wirksam wird, muss der aktive G0 in N30 mit G1 überschrieben werden, andernfalls würde die Kontur mit G0 weiter bearbeitet werden).
- Weiches Abfahren (Satz N60 aktiviert)
- Abfahrbewegung mit Viertelkreis (G248) und Helix (G340)
- FAD nicht programmiert, da bei G340 ohne Bedeutung
- Z=2 im Startpunkt; Z=8 im Endpunkt, da DISCL=6
- Bei DISR=5 ist Radius der WAB-Kontur=20, der Radius der Werkzeugmittelpunktsbahn=5

Wegfahrbewegungen von Z8 nach Z20 und die Bewegung parallel zur X-Y Ebene zu X70 Y0.

| **Programmcode** | **Kommentar** |
|---|---|
| `$TC_DP1[1,1]=120` | `; Werkzeugdefinition T1/D1` |
| `$TC_DP6[1,1]=10` | `; Radius` |
| `N10 G0 X0 Y0 Z20 G64 D1 T1 OFFN=5` | `; (P0 an)` |
| `N20 G41 G247 G341 Z0 DISCL=AC(7) DISR=10 F1500 FAD=200` | `; Anfahren (P3an)` |
| `N30 G1 X30 Y-10` | `; (P4 an)` |
| `N40 X40 Z2` | |
| `N50 X50` | `; (P4 ab)` |
| `N60 G248 G340 X70 Y0 Z20 DISCL=6 DISR=5 G40 F10000` | `; Abfahren (P3 ab)` |
| `N70 X80 Y0` | `; (P0 ab)` |
| `N80 M30` | |

## Weitere Informationen

### Wahl der An- bzw. Abfahrkontur

Die Wahl der An- bzw. Abfahrkontur erfolgt mit dem entsprechenden G-Befehl aus der 2. G-Gruppe:

| | |
|---|---|
| G147: | Anfahren mit einer Geraden |
| G247: | Anfahren mit einem Viertelkreis |
| G347: | Anfahren mit einem Halbkreis |
| G148: | Abfahren mit einer Geraden |
| G248: | Abfahren mit einem Viertelkreis |
| G348: | Abfahren mit einem Halbkreis |

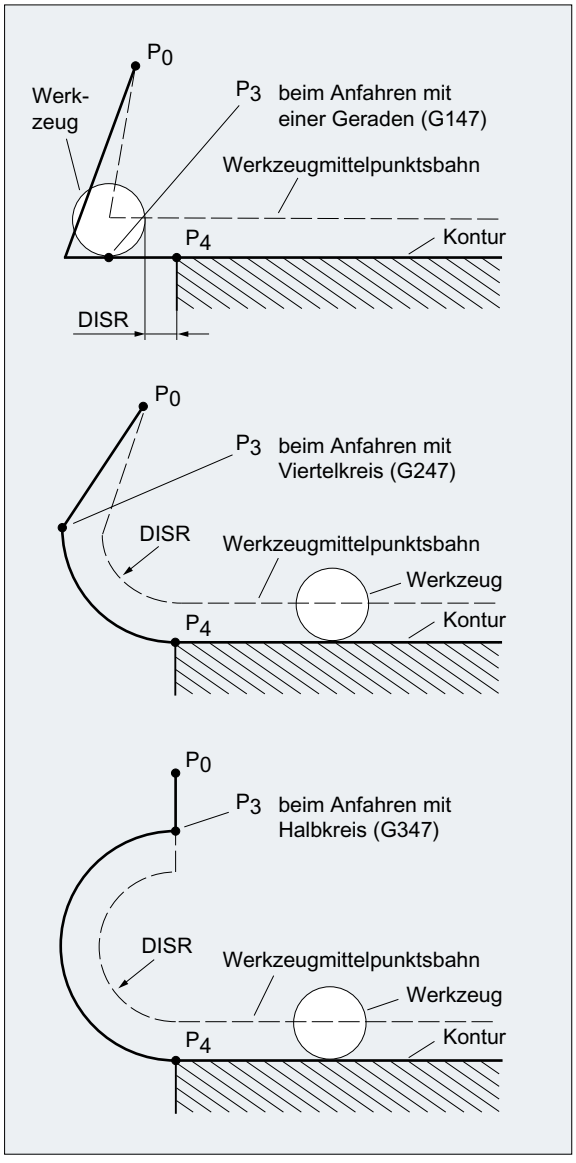



Bild 2-16 Anfahrbewegungen bei gleichzeitiger Aktivierung der Werkzeugradiuskorrektur

### Wahl der An- bzw. Abfahrrichtung

Bestimmung der An- und Abfahrrichtung mit Hilfe der Werkzeugradiuskorrektur (G140, Grundeinstellung) bei positivem Werkzeugradius:

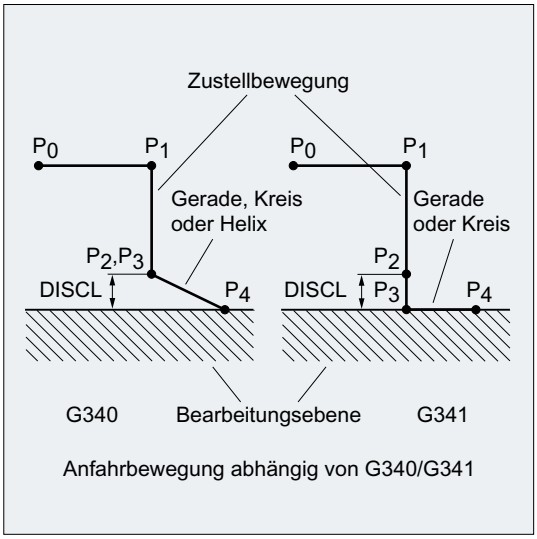
- G41 aktiv → Anfahren von links
- G42 aktiv → Anfahren von rechts

Weitere Anfahrmöglichkeiten sind mit G141, G142 und G143 gegeben.

Diese G-Befehle sind nur dann von Bedeutung, wenn die Anfahrkontur ein Viertel- oder Halbkreis ist.

### Aufteilung der Bewegung vom Start- zum Endpunkt (G340 und G341)

Die Bewegungen setzen sich in jedem Fall aus einer oder mehreren Geraden sowie abhängig vom G-Befehl zur Bestimmung der Anfahrkontur aus einer weiteren Geraden bzw. einem Viertel- oder Halbkreis zusammen. Die 2 Varianten der Wegaufteilung sind in folgendem Bild dargestellt:




Anfahrbewegung abhängig von G340/G341

| | |
|---|---|
| G340: | Anfahren mit einer Geraden vom Punkt $P_0$ zum Punkt $P_1$. Diese Gerade ist parallel zur Bearbeitungsebene, wenn der Parameter DISRP nicht programmiert wurde.<br>Senkrecht zur Bearbeitungsebene vom Punkt $P_1$ zum Punkt $P_3$ zustellen auf den durch den Parameter DISCL definierten Sicherheitsabstand zur Bearbeitungsebene.<br>Anfahren des Endpunkts P4 mit der durch den G-Befehl der zweiten Gruppe bestimmten Kurve (Gerade, Kreis, Helix). Ist G247 oder G347 aktiv (Viertel- oder Halbkreis) und der Startpunkt $P_3$ liegt nicht in der durch den Endpunkt $P_4$ definierten Bearbeitungsebene, wird statt eines Kreises eine Helix eingefügt. Der Punkt $P_2$ ist nicht definiert bzw. er fällt mit $P_3$ zusammen.<br>Die Kreisebene bzw. die Helixachse wird dabei durch die in im WAB-Satz aktive Ebene (G17/G18/G19) bestimmt, d. h. vom Folgesatz wird nicht die Starttangente selbst zur Bestimmung des Kreises verwendet, sondern deren Projektion in die aktive Ebene.<br>Die Bewegung vom Punkt $P_0$ zum Punkt $P_3$ erfolgt in zwei Geraden mit der Geschwindigkeit, die vor dem WAB-Satz wirksam war. |
| G341: | Anfahren mit einer Geraden vom Punkt $P_0$ zum Punkt $P_1$. Diese Gerade ist parallel zur Bearbeitungsebene, wenn der Parameter DISRP nicht programmiert wurde.<br>Senkrecht zur Bearbeitungsebene vom Punkt $P_1$ bis zu dem durch den Parameter DISCL definierten Sicherheitsabstand zur Bearbeitungsebene im Punkt $P_2$ zustellen.<br>Senkrecht zur Bearbeitungsebene vom Punkt $P_2$ zum Punkt $P_3$ zustellen. Anfahren des Endpunkts mit der durch den G-Befehl der zweiten Gruppe bestimmten Kurve. $P_3$ und $P_4$ liegen in der Bearbeitungsebene, so dass bei G247 bzw. G347 nie eine Helix, sondern immer ein Kreis eingefügt wird. |

In allen Fällen, in denen die Lage der aktiven Ebene G17/G18/G19 eingeht (Kreisebene, Helixachse, Zustellbewegungen senkrecht zur aktiven Ebene), wird ein eventuell aktiver drehender Frame berücksichtigt.

### Länge der Anfahrgerade bzw. Radius bei Anfahrkreisen (DISR)

- An-/Abfahren mit Geraden
  DISR gibt den Abstand der Fräserkante vom Startpunkt der Kontur an, d. h. die Länge der Geraden ergibt sich bei aktiver WRK als Summe von Werkzeugradius und programmiertem Wert von DISR. Der Werkzeugradius wird nur berücksichtigt, wenn er positiv ist.
  Die resultierende Geradenlänge muss positiv sein, d. h. es sind negative Werte für DISR zulässig, solange der Betrag von DISR kleiner als der Werkzeugradius ist.
- An-/Abfahren mit Kreisen
  DISR gibt den Radius der Werkzeugmittelpunktsbahn an. Ist WRK aktiviert, wird ein Kreis mit einem solchen Radius erzeugt, dass auch in diesem Fall die Werkzeugmittelpunktsbahn mit dem programmierten Radius resultiert.

### Abstand des Punktes P2 von der Bearbeitungsebene (DISCL)

Soll die Position des Punkts $P_2$ auf der Achse senkrecht zur Kreisebene absolut angegeben werden, ist der Wert in der Form `DISCL=AC(...)` zu programmieren.

Bei DISCL=0 gilt:

- Bei G340: Die gesamte Anfahrbewegung besteht nur noch aus zwei Sätzen ($P_1$, $P_2$ und $P_3$ fallen zusammen). Die Anfahrkontur wird von $P_1$ zu $P_4$ gebildet.
- Bei G341: Die gesamte Anfahrbewegung besteht aus drei Sätzen ($P_2$ und $P_3$ fallen zusammen). Liegen $P_0$ und $P_4$ in der gleichen Ebene entstehen nur zwei Sätze (Zustellbewegung von $P_1$ nach $P_3$ entfällt).
- Es wird überwacht, dass der durch DISCL definierte Punkt zwischen $P_1$ und $P_3$ liegt, d. h. bei allen Bewegungen, die eine Komponente senkrecht zur Bearbeitungsebene haben, muss diese Komponente das gleiche Vorzeichen haben.
- Bei der Erkennung der Richtungsumkehr wird eine durch das Maschinendatum MD20204 $MC_WAB_CLEARANCE_TOLERANCE definierte Toleranz zugelassen.

### Abstand des Punkts P1 (Rückzugsebene) von der Bearbeitungsebene (DISRP)

Soll die Position des Punkts $P_1$ auf der Achse senkrecht zur Bearbeitungsebene absolut angegeben werden, ist der Wert in der Form `DISRP=AC(...)` zu programmieren.

Ist dieser Parameter nicht programmiert, hat der Punkt $P_1$ den gleichen Abstand von der Bearbeitungsebene wie der Punkt $P_0$, d. h die Anfahrgerade $P_0 \rightarrow P_1$ ist parallel zur Bearbeitungsebene.

Es wird überwacht, dass der durch DISRP definierte Punkt zwischen $P_0$ und $P_2$ liegt, d. h. bei allen Bewegungen, die eine Komponente senkrecht zur Bearbeitungsebene haben (Zustellbewegungen, Anfahrbewegung von $P_3$ nach $P_4$) muss diese Komponente das gleiche Vorzeichen haben. Eine Richtungsumkehr ist nicht zugelassen. Gegebenenfalls wird ein Alarm ausgegeben.

Bei der Erkennung der Richtungsumkehr wird eine durch das Maschinendatum MD20204 $MC_WAB_CLEARANCE_TOLERANCE definierte Toleranz zugelassen. Liegt $P_1$ außerhalb des durch $P_0$ und $P_2$ definierten Bereichs, ist die Abweichung jedoch kleiner oder gleich dieser Toleranz, wird angenommen, dass $P_1$ in der durch $P_0$ bzw. $P_2$ definierten Ebene liegt.

### Programmierung des Endpunkts

Der Endpunkt wird in der Regel programmiert mit X... Y... Z...

Die Programmierung des Konturendpunkts beim Anfahren unterscheidet sich wesentlich von der beim Abfahren. Die beiden Fälle werden deshalb hier getrennt behandelt.

### Programmierung des Endpunktes P4 beim Anfahren

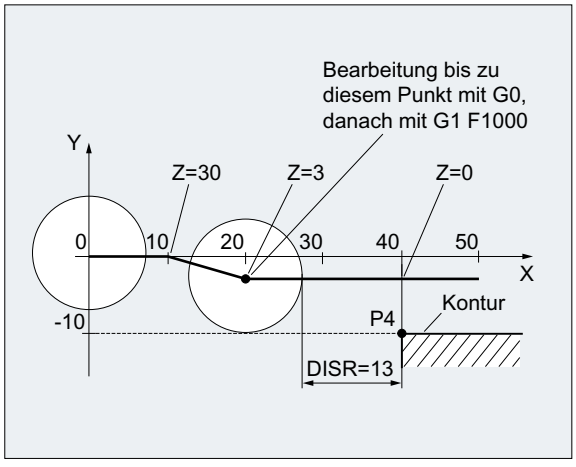
Der Endpunkt $P_4$ kann im WAB-Satz selbst programmiert sein. Alternativ dazu besteht die Möglichkeit, $P_4$ durch den Endpunkt des nächsten Verfahrsatzes zu bestimmen. Zwischen WAB-Satz und nächsten Verfahrsatz können weitere Sätze ohne Bewegung der Geometrieachsen eingefügt werden.

Beispiel:

| **Programmcode** | **Kommentar** |
|---|---|
| `$TC_DP1[1,1]=120` | `; Fräserwerkzeug T1/D1` |
| `$TC_DP6[1,1]=7` | `; Werkzeug mit 7 mm Radius` |
| `N10 G90 G0 X0 Y0 Z30 D1 T1` | |
| `N20 X10` | |
| `N30 G41 G147 DISCL=3 DISR=13 Z=0 F1000` | |
| `N40 G1 X40 Y-10` | |
| `N50 G1 X50` | |
| `...` | |

`N30/N40` kann ersetzt werden durch:
```
N30 G41 G147 DISCL=3 DISR=13 X40 Y-10 Z0 F1000
```

bzw.
```
N30 G41 G147 DISCL=3 DISR=13 F1000
N40 G1 X40 Y-10 Z0
```




### Programmierung des Endpunktes P0 beim Abfahren

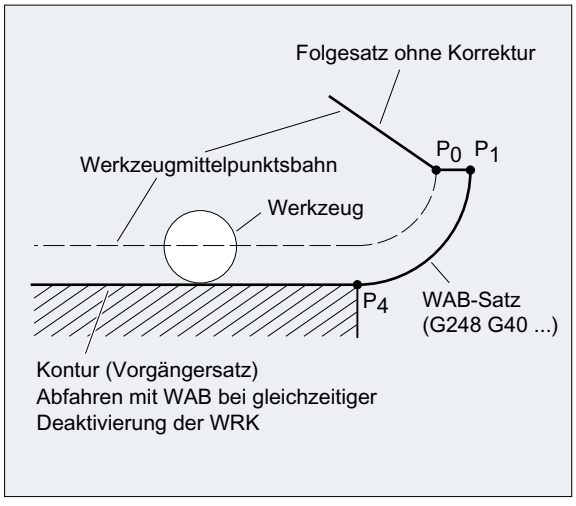
Beim Abfahren ist die Programmierung des Endpunkts der WAB-Kontur in einem Folgesatz nicht vorgesehen, d. h. die Endposition wird immer aus dem WAB-Satz selbst entnommen,

unabhängig davon wie viele Achsen programmiert wurden. Bei der Bestimmung des Endpunkts sind dabei die folgenden drei Fälle zu unterscheiden:

1. Im WAB-Satz ist keine Geometrieachse programmiert. Die Kontur endet in diesem Fall im Punkt $P_1$ (falls DISRP programmiert ist), im Punkt $P_2$ (falls DISCL aber nicht DISRP programmiert ist) oder im Punkt $P_3$ (falls weder DICLS noch DISRP programmiert ist). Die Position in den Achsen, die die Bearbeitungsebene bilden, ergibt sich aus der Wegfahrkontur (Endpunkt der Geraden bzw. des Kreises). Die Achskomponente senkrecht dazu wird durch DISCL bzw. DISPR definiert. Ist in diesem Fall sowohl DISCL=0 als auch DISRP=0, verläuft die Bewegung deshalb vollständig in der Ebene, d. h. die Punkt $P_0$ bis $P_3$ fallen zusammen.
2. Im WAB-Satz ist nur die Achse senkrecht zur Bearbeitungsebene programmiert. Die Kontur endet in diesem Fall im Punkt $P_0$. Falls DISRP programmiert ist (d. h. die beiden Punkte $P_0$ und $P_1$ fallen nicht zusammen), verläuft die Gerade $P_1 \rightarrow P_0$ senkrecht zur Bearbeitungsebene. Die Positionen der übrigen beiden Achsen ergeben sich wie unter 1.
3. Es ist mindestens eine Achse der Bearbeitungsebene programmiert. Die eventuell fehlende zweite Achse der Bearbeitungsebene wird aus ihrer letzten Position im Vorgängersatz modal ergänzt.

Die Position der Achse senkrecht zur Bearbeitungsebene wird – abhängig davon, ob diese Achse programmiert ist oder nicht – wie unter 1. oder 2. gebildet. Die so gebildete Position definiert den Endpunkt $P_0$. Ist der WAB-Abfahrsatz gleichzeitig der Deaktivierungssatz der Werkzeugradiuskorrektur, so wird in den ersten beiden Fällen eine zusätzliche Wegkomponente in der Bearbeitungsebene von $P_1$ nach $P_0$ derart eingefügt, dass sich bei der Deaktivierung der Werkzeugradiuskorrektur am Ende der Wegfahrkontur keine Bewegung ergibt, d. h. dieser Punkt definiert dann nicht eine Position auf einer zu korrigierenden Kontur, sondern den Werkzeugmittelpunkt. Im Fall drei muss die Abwahl der Werkzeugradiuskorrektur nicht gesondert behandelt werden, da der programmierte Punkt $P_0$ bereits unmittelbar die Position des Werkzeugmittelpunkts am Ende der Gesamtkontur definiert.

Das Verhalten in den Fällen 1 und 2, d. h. bei nicht explizit programmiertem Endpunkt in der Bearbeitungsebene bei gleichzeitiger Abwahl der Werkzeugradiuskorrektur, ist in folgendem Bild dargestellt:

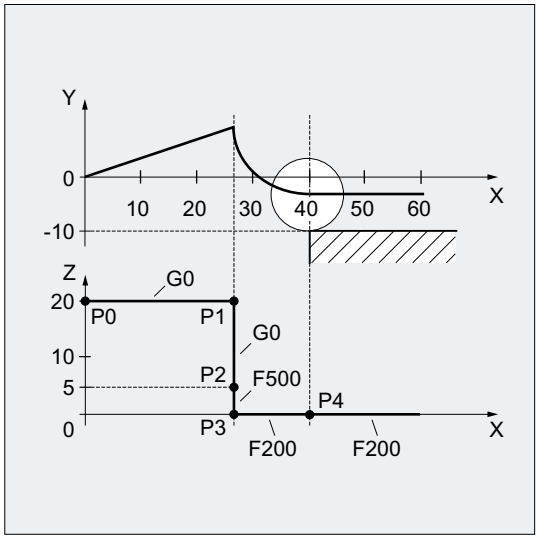


### An- bzw. Abfahrgeschwindigkeiten

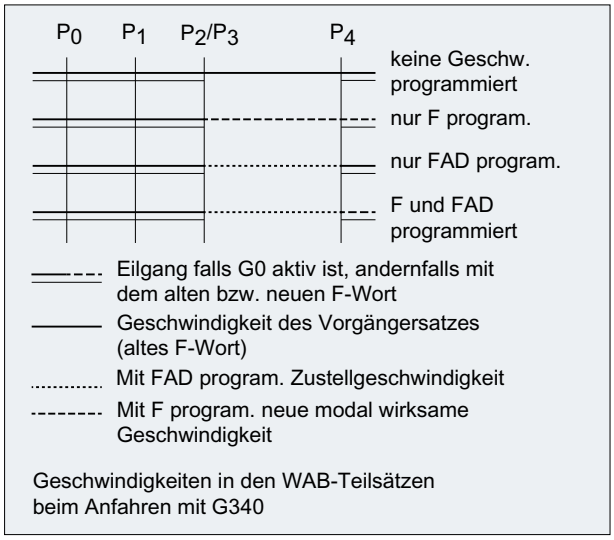
- Geschwindigkeit des Vorgängersatzes (G0)
  Mit dieser Geschwindigkeit werden alle Bewegungen von $P_0$ bis zu $P_2$ ausgeführt, d. h. die Bewegung parallel zur Bearbeitungsebene und der Teil der Zustellbewegung bis zum Sicherheitsabstand.
- Programmierung mit FAD
  Angabe der Vorschubgeschwindigkeit bei
  - G341: Zustellbewegung senkrecht zur Bearbeitungsebene von $P_2$ nach $P_3$
  - G340: von Punkt $P_2$ bzw. $P_3$ zum $P_4$
    Wird FAD nicht programmiert, wird dieser Teil der Kontur ebenfalls mit der modal wirksamen Geschwindigkeit des Vorgängersatzes, falls im WAB-Satz kein F-Wort programmiert ist, verfahren.
- Programmierter Vorschub F
  Dieser Vorschubwert ist ab $P_3$ bzw. $P_2$ wirksam, falls FAD nicht programmiert ist. Wird im WAB-Satz kein F-Wort programmiert, wirkt die Geschwindigkeit des Vorgängersatzes.

Beispiel:

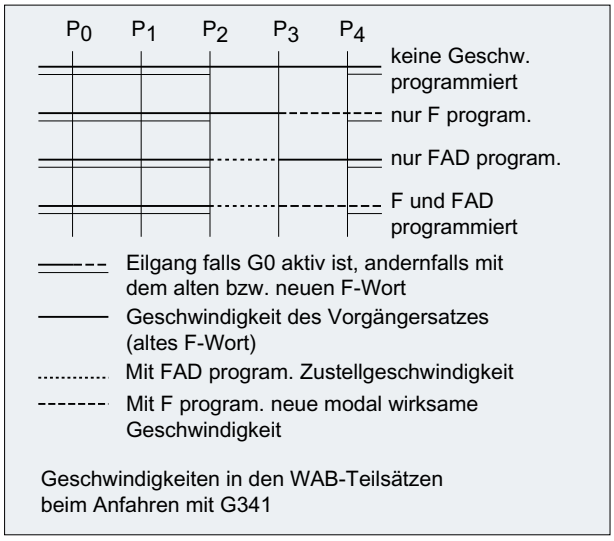
| **Programmcode** | **Kommentar** |
|---|---|
| `$TC_DP1[1,1]=120` | `; Fräserwerkzeug T1/D1` |
| `$TC_DP6[1,1]=7` | `; Werkzeug mit 7mm Radius` |
| `N10 G90 G0 X0 Y0 Z20 D1 T1` | |
| `N20 G41 G341 G247 DISCL=AC(5) DISR=13 FAD 500 X40 Y-10 Z=0 F200` | |
| `N30 X50` | |
| `N40 X60` | |
| `...` | |



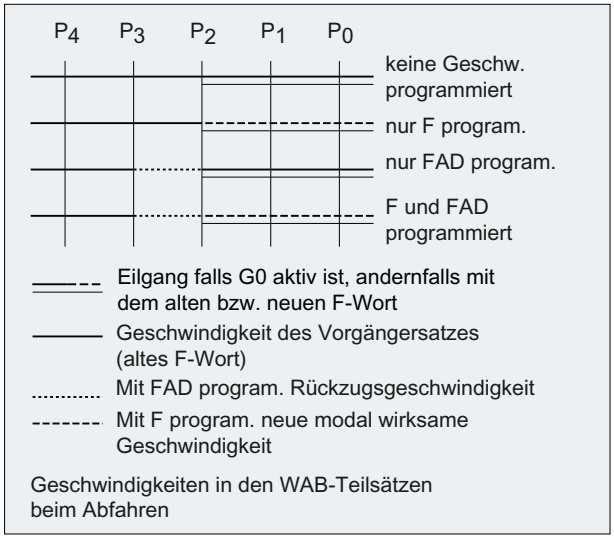

Beim Abfahren sind die Rollen von modal wirksamem Vorschub aus dem Vorgängersatz und dem im WAB-Satz programmierten Vorschubwert vertauscht, d. h. die eigentliche Abfahrkontur wird mit dem alten Vorschub verfahren, eine neu mit F-Wort programmierte Geschwindigkeit gilt entsprechend ab $P_2$ bis zum $P_0$.




Geschwindigkeiten in den WAB-Teilsätzen beim Anfahren mit G340




Geschwindigkeiten in den WAB-Teilsätzen beim Anfahren mit G341




Geschwindigkeiten in den WAB-Teilsätzen beim Abfahren

**Lesen von Positionen**

Die Punkte $P_3$ und $P_4$ können beim Anfahren als Systemvariable im WKS gelesen werden.

- $P_APR: Lesen von P
- $_3$ (Aufstartpunkt)
- $P_AEP: Lesen von P
- $_4$ (Konturanfangspunkt)
- $P_APDV: Lesen, ob $P_APR und $P_AEP gültige Werte enthalten

## 2.10.4.2 An- und Abfahren mit erweiterten Abfahrstrategien (G460, G461, G462)

In bestimmten geometrischen Sonderfällen werden gegenüber der bisherigen Realisierung mit eingeschalteter Kollisionsüberwachung für An- und Abfahrsatz, spezielle erweiterte An- und Abfahrstrategien beim Aktivieren bzw. Deaktivieren der Werkzeugradiuskorrektur benötigt. So kann z. B. eine Kollisionsüberwachung dazu führen, dass ein Abschnitt auf der Kontur nicht vollständig bearbeitet wird, siehe folgendes Bild:

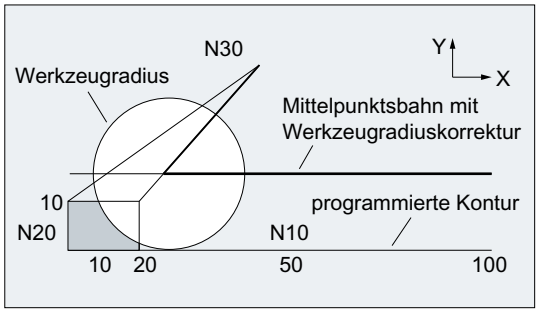



Bild 2-17 Abfahrverhalten bei G460

### Syntax

```
G460
G461
G462
```

### Bedeutung

| | |
|---|---|
| G460: | Wie bisher (Einschalten der Kollisionsüberwachung für An- und Abfahrsatz) |
| G461: | Einfügen eines Kreises im WRK-Satz, wenn kein Schnittpunkt möglich ist, dessen Mittelpunkt im Endpunkt des nicht korrigierten Satzes liegt, und dessen Radius gleich dem Werkzeugradius ist.<br>Bis zum Schnittpunkt wird mit **Hilfskreis** um den Konturendpunkt (also bis Konturende) bearbeitet. |
| G462: | Einfügen einer Geraden im WRK-Satz, wenn kein Schnittpunkt möglich ist, der Satz wird durch seine Endtangente verlängert (Standardeinstellung)<br>Bearbeitet wird bis zur **Verlängerung** des letzten Konturelements (also bis kurz vor Konturende). |

**Hinweis**

Das Anfahrverhalten ist symmetrisch zum Abfahrverhalten.

Das An- bzw. Abfahrverhalten wird vom Zustand des G-Befehls im An- bzw. Abfahrsatz bestimmt. Das Anfahrverhalten kann deshalb unabhängig vom Abfahrverhalten eingestellt werden.

## Beispiele

### Beispiel 1: Abfahrverhalten bei G460

Im Folgenden wird immer nur die Situation bei Deaktivieren der Werkzeugradiuskorrektur dargestellt. Das Verhalten beim Anfahren ist dazu völlig analog.

| Programmcode | Kommentar |
|---|---|
| G42 D1 T1 | ; Werkzeugradius 20mm |
| ... | |
| G1 X110 Y0 | |
| N10 X0 | |
| N20 Y10 | |
| N30 G40 X50 Y50 | |

### Beispiel 2: Anfahren bei G461

| Programmcode | Kommentar |
|---|---|
| N10 $TC_DP1[1,1]=120 | ; Werkzeugtyp Fräser |
| N20 $TC_DP6[1,1]=10 | ; Werkzeugradius |
| N30 X0 Y0 F10000 T1 D1 | |
| N40 Y20 | |
| N50 G42 X50 Y5 G461 | |
| N60 Y0 F600 | |
| N70 X30 | |
| N80 X20 Y-5 | |
| N90 X0 Y0 G40 | |
| N100 M30 | |

## Weitere Informationen

### G461

Wenn kein Schnittpunkt des letzten WRK-Satzes mit einem Vorgängersatz möglich ist, wird die Offsetkurve dieses Satzes mit einem Kreis verlängert, dessen Mittelpunkt im Endpunkt des nicht korrigierten Satzes liegt, und dessen Radius gleich dem Werkzeugradius ist.

Die Steuerung versucht, diesen Kreis mit einem der Vorgängersätze zu schneiden.

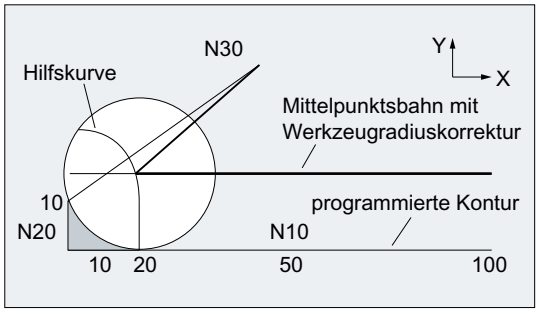



Bild 2-18 Abfahrverhalten bei G461

Kollisionsüberwachung CDON, CDOF

Dabei wird bei aktivem CDOF (siehe Abschnitt Kollisionsüberwachung, CDON, CDOF) die Suche abgebrochen, wenn ein Schnittpunkt gefunden wurde, d. h. es wird nicht überprüft, ob auch noch Schnittpunkte mit weiter in der Vergangenheit liegenden Sätzen existieren.

Bei aktivem CDON wird auch dann, wenn bereits ein Schnittpunkt gefunden wurde, nach weiteren Schnittpunkten gesucht.

Ein so gefundener Schnittpunkt ist der neue Endpunkt eines Vorgängersatzes und der Startpunkt des Deaktivierungssatzes. Der eingefügte Kreis dient nur zur Schnittpunktberechnung und hat selbst keine Verfahrbewegung zur Folge.

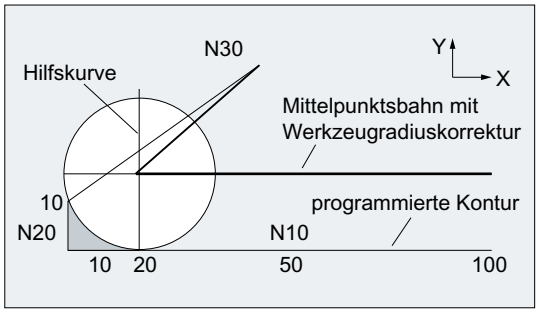
**Hinweis**

Wird kein Schnittpunkt gefunden, wird der Alarm 10751 (Kollisionsgefahr) ausgegeben.

### G462

Wenn kein Schnittpunkt des letzten WRK-Satzes mit einem Vorgängersatz möglich ist, wird beim Abfahren mit G462 (Grundstellung) im Endpunkt des letzen Satzes mit Werkzeugradiuskorrektur eine Gerade eingefügt (der Satz wird durch seine Endtangente verlängert).

Die Schnittpunktsuche verläuft dann identisch zu der bei G461.




Abfahrverhalten bei G462 (siehe Beispiel)

Bei G462 wird die im Beispielprogramm von N10 und N20 gebildete Ecke nicht soweit ausgeräumt, wie es mit dem verwendeten Werkzeug möglich wäre. Dieses Verhalten kann aber dennoch notwendig sein, wenn die Teilekontur (abweichend von der programmierten Kontur) im Beispiel links von N20 auch bei größeren Werten von y als 10 mm nicht verletzt werden darf.

**Eckenverhalten bei KONT**

Ist KONT aktiv (Kontur im Start- oder Endpunkt umfahren), wird unterschieden, ob der Endpunkt vor oder hinter der Kontur liegt.

- **Endpunkt vor der Kontur**
  Liegt der Endpunkt vor der Kontur, ist das Abfahrverhalten gleich wie bei NORM. Diese Eigenschaft ändert sich auch nicht, wenn der letzte Kontursatz bei G451 mit einer Geraden oder einem Kreis verlängert wird. Zusätzliche Umfahrungsstrategien, um eine Konturverletzung in der Nähe des Konturendpunktes zu vermeiden, sind deshalb nicht notwendig.
- **Endpunkt hinter der Kontur**
  Liegt der Endpunkt hinter der Kontur, wird immer abhängig von G450/G451 ein Kreis bzw. eine Gerade eingefügt. G460 - G462 hat dann keine Bedeutung. Hat der letzte Verfahrsatz in dieser Situation keinen Schnittpunkt mit einem Vorgängersatz, kann sich nun ein Schnittpunkt mit dem eingefügten Konturelement oder mit dem Geradenstück vom Endpunkt des Umfahrungskreises zum programmierten Endpunkt ergeben.
  Ist das eingefügte Konturelement ein Kreis (G450), und dieses bildet mit dem Vorgängersatz einen Schnittpunkt, ist dieser gleich dem Schnittpunkt, der sich auch bei NORM und G461 ergeben würde. Im allgemeinen bleibt jedoch ein zusätzliches Stück des Kreises zu verfahren. Für den linearen Teil des Abfahrsatzes ist keine Schnittpunktberechnung mehr notwendig.
  Im zweiten Fall, wenn kein Schnittpunkt des eingefügten Konturelements mit den Vorgängersätzen gefunden wird, wird auf den Schnittpunkt zwischen der Abfahrgeraden und einem Vorgängersatz verfahren.
  Es kann sich somit bei aktivem G461 bzw. G462 nur dann ein gegenüber G460 verändertes Verhalten ergeben, wenn entweder NORM aktiv ist, oder das Verhalten bei KONT geometrisch bedingt identisch zu dem bei NORM ist.

## 2.10.5 Kollisionsüberwachung ("Flaschenhalserkennung") ein-/ausschalten (CDON, CDOF, CDOF2)

Die Kollisionsüberwachung ("Flaschenhalserkennung") bei aktiver WRK wird im NC-Programm mit den Befehlen der G-Gruppe 23 ein- bzw. ausgeschaltet.

### Syntax

```
G41/G42 CDON
...
CDOF/CDOF2
```

## Bedeutung

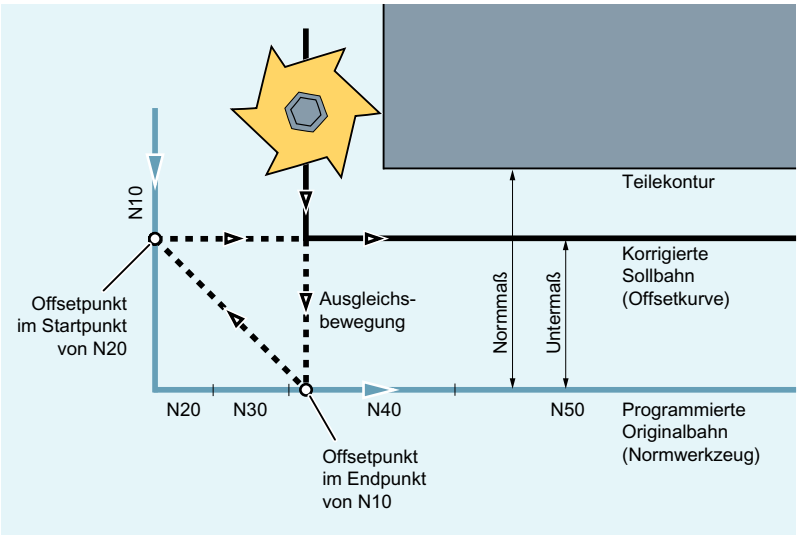
| | |
|---|---|
| CDON: | Kollisionsüberwachung ("Flaschenhalserkennung") einschalten<br>Mit CDON wird über eine einstellbare (MD20240) Anzahl an Sätzen hinweg überprüft, ob sich die Werkzeugwege von **nicht benachbarten** Sätzen schneiden. Hierdurch lassen sich mögliche Kollisionen rechtzeitig erkennen und aktiv durch die Steuerung verhindern. |
| CDOF: | Kollisionsüberwachung ("Flaschenhalserkennung") ausschalten<br>Mit CDOF wird für den aktuellen Satz beim **vorhergehenden** Verfahrsatz (an Innenecken) nach einem gemeinsamen Schnittpunkt gesucht, gegebenenfalls auch in weiter zurückliegenden Sätzen. Wird ein Schnittpunkt gefunden, werden keine weiteren Sätze betrachtet. Bei Außenecken wird zwischen zwei aufeinander folgenden Sätzen immer ein Schnittpunkt gefunden.<br>**Hinweis:**<br>Mit CDOF lässt sich die fehlerhafte Erkennung von Engstellen vermeiden, die z. B. auf fehlende Informationen im NC-Programm zurückzuführen sind. |
| CDOF2: | Kollisionsüberwachung für das 3D-Umfangsfräsen ausschalten<br>Mit CDOF2 wird die Werkzeugkorrekturrichtung aus benachbarten Satzteilen ermittelt. CDOF2 wirkt nur beim 3D-Umfangsfräsen und hat bei allen anderen Bearbeitungsarten (z. B. 3D-Stirnfräsen) die gleiche Bedeutung wie CDOF. |

### Auswirkung der Kollisionsüberwachung an einem Beispiel

Das NC-Programm beschreibt die Mittelpunktsbahn eines Normwerkzeugs. Die Kontur für ein aktuell verwendetes Werkzeug ergibt ein Untermaß, welches im folgenden Bild zur Verdeutlichung der geometrischen Verhältnisse unrealistisch groß dargestellt ist.

Außerdem soll für das Beispiel gelten, dass die Steuerung nur drei Sätze überblickt:

MD20240 $MC_CUTCOM_MAXNUM_CHECK_BLOCKS = 3




Da ein Schnittpunkt nur zwischen den Offsetkurven der beiden Sätze N10 und N40 existiert, müssten die beiden Sätze N20 und N30 ausgelassen werden. Im Beispiel ist der Steuerung der Satz N40 noch nicht bekannt, wenn N10 abschließend bearbeitet werden muss. Damit kann nur ein einzelner Satz ausgelassen werden.

Bei aktiven CDOF2 wird die im Bild dargestellte Ausgleichsbewegung ausgeführt und nicht angehalten. In dieser Situation würde ein aktives CDOF oder CDON zu einem Alarm führen.

## 2.10.6 2 1/2 D-Werkzeugkorrektur (CUT2D, CUT2DD, CUT2DF, CUT2DFD)

Die 2½ D-Werkzeugradiuskorrektur ist zu verwenden, wenn für die Bearbeitung von schrägen Flächen nicht die Ausrichtung des Werkzeugs, sondern das **Werkstück** gedreht wird. Die Aktivierung erfolgt durch die Befehle CUT2D, CUT2DD, CUT2DF oder CUT2DFD.

### Werkzeuglängenkorrektur

Die Werkzeuglängenkorrektur wird immer bezogen auf die raumfeste, nicht gedrehte Arbeitsebene eingerechnet.

### 2½ D-Werkzeugradiuskorrektur für Konturwerkzeuge

Die 2½ D-Werkzeugradiuskorrektur für Konturwerkzeuge wird aktiviert, wenn zusammen mit CUT2D, CUT2DD, CUT2DF oder CUT2DFD einer der beiden Befehle G41 (Werkzeugradiuskorrektur links von der Kontur) oder G42 (Werkzeugradiuskorrektur rechts von der Kontur) programmiert wird. Sie dient zur automatischen Schneidenauswahl für nicht rotationssymmetrische Werkzeuge, mit denen stückweise einzelne Kontursegmente bearbeitet werden können.

**Hinweis**

Bei nicht aktiver 2½ D-Werkzeugradiuskorrektur verhält sich ein Konturwerkzeug wie ein normales Werkzeug, das nur aus der ersten Schneide besteht.

### 2½ D-Werkzeugradiuskorrektur bezogen auf ein Differenzwerkzeug

Die auf ein Differenzwerkzeug bezogene 2½ D-Werkzeugradiuskorrektur wird durch die Befehle CUT2DD oder CUT2DFD aktiviert. Sie ist anzuwenden, wenn sich die programmierte Kontur auf die Mittelpunktsbahn eines Differenzwerkzeugs bezieht und die Bearbeitung mit einem davon abweichenden Werkzeug erfolgt. Bei der Berechnung der 2½ D-Werkzeugradiuskorrektur werden dann nur der Verschleißwert des Radius des aktiven Werkzeugs ($TC_DP_15) und die gegebenenfalls programmierten Werkzeugkorrekturoffsets OFFN (Seite 254) und TOFFR (Seite 85) eingerechnet. Der Grundradius ($TC_DP6) des aktiven Werkzeugs wird **nicht** eingerechnet.

## Syntax

```
CUT2D
CUT2DD
CUT2DF
CUT2DFD
```

## Bedeutung

| | |
|---|---|
| CUT2D: | Aktivierung der 2½ D-Radiuskorrektur |
| CUT2DD: | Aktivierung der auf ein Differenzwerkzeug bezogenen 2½ D-Radiuskorrektur |
| CUT2DF: | Aktivierung der 2½ D-Radiuskorrektur, Werkzeugradiuskorrektur relativ zum aktuellen Frame bzw. schrägen Ebene |
| CUT2DFD: | Aktivierung der auf ein Differenzwerkzeug bezogenen 2½ D-Radiuskorrektur relativ zum aktuellen Frame bzw. schrägen Ebene |

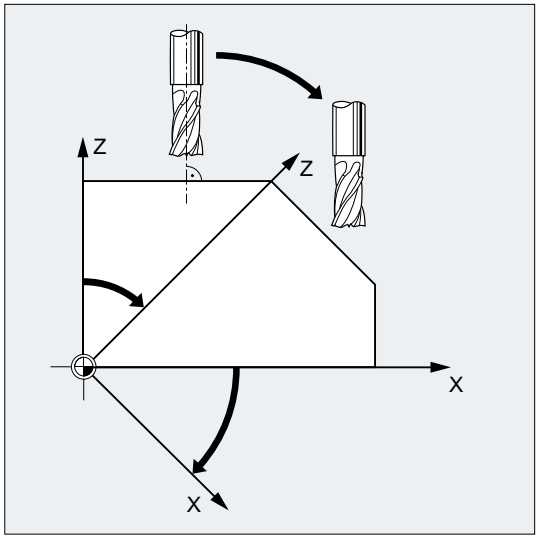
## Weitere Informationen

### Konturwerkzeuge

- Freigabe
  Die Freigabe der Werkzeugradiuskorrektur für Konturwerkzeuge erfolgt kanalspezifisch über:
  MD28290 $MC_MM_SHAPED_TOOLS_ENABLE
- Werkzeugtyp
  Die Werkzeugtypen von Konturwerkzeugen werden kanalspezifisch festgelegt über:
  MD20370 $MC_SHAPED_TOOL_TYPE_NO
- Schneiden
  Jedem Konturwerkzeug können in beliebiger Reihenfolge eine Anzahl Schneiden (D-Nummern) zugeordnet werden. Die maximale Anzahl von Schneiden pro Werkzeug wird parametriert über:
  MD18106 $MN_MM_MAX_CUTTING_EDGE_PERTOOL
  Die erste Schneide eines Konturwerkzeugs ist die Schneide, die bei der Aktivierung des Werkzeugs angewählt wird. Wird z. B. in einem Programm durch die Befehle `T3 D5` die fünfte Schneide (D5) des dritten Werkzeugs (T3) aktiviert, dann definiert D5 und die nachfolgenden Schneiden entweder mit einem Teil oder alle zusammen das Konturwerkzeug. Die vor D5 liegenden Schneiden werden ignoriert.

### 2½ D-Werkzeugradiuskorrektur ohne Drehung der Korrekturebene (CUT2D, CUT2DD)

Wird ein Frame programmiert, der eine Drehung enthält, wird bei `CUT2D` bzw. `CUT2DD` die Ebene, in der die Werkzeugradiuskorrektur (Korrekturebene) stattfindet, **nicht mitgedreht**. Die Werkzeugradiuskorrektur wird bezogen auf die **nicht gedrehte** Arbeitsebene (`G17`, `G18`, `G19`) eingerechnet. Die Werkzeuglängenkorrektur wirkt relativ zur Korrekturebene.

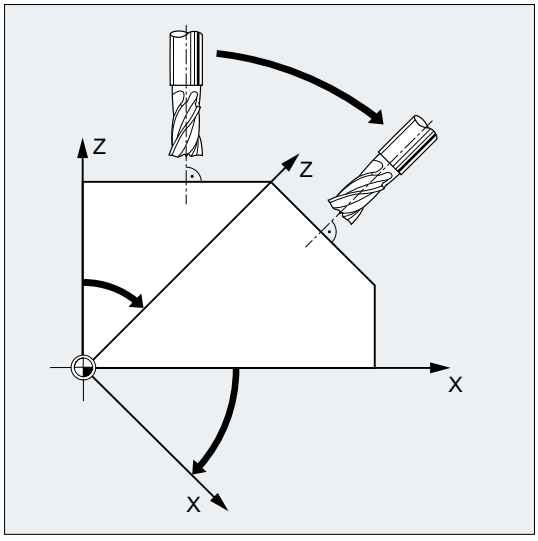



Für die Bearbeitung von schrägen Flächen müssen die Werkzeugkorrekturwerte entsprechend definiert oder unter Einsatz der Funktionalitäten zur "Werkzeuglängenkorrektur für orientierbare Werkzeuge" berechnet werden.

### 2½ D-Werkzeugradiuskorrektur mit Drehung der Korrekturebene (CUT2DF, CUT2DFD)

Wird ein Frame programmiert, der eine Drehung enthält, wird bei `CUT2DF` bzw. `CUT2DFD` die Ebene, in der die Werkzeugradiuskorrektur (Korrekturebene) stattfindet, **mitgedreht**. Die Werkzeugradiuskorrektur wird bezogen auf die **gedrehte** Arbeitsebene (`G17`, `G18`, `G19`) eingerechnet. Die Werkzeuglängenkorrektur wirkt aber weiterhin relativ zur **nicht gedrehten** Arbeitsebene.

**Voraussetzung**: An der Maschine muss die Werkzeugorientierung senkrecht zur gedrehten Arbeitsebene einstellbar sein und zur Bearbeitung eingestellt werden.




**Hinweis**

Die Werkzeuglängenkorrektur wirkt weiterhin relativ zur nicht gedrehten Arbeitsebene.

**Weitere Informationen siehe Funktionshandbuch "Werkzeuge".**

## 2.10.7 Werkzeugradiuskorrektur konstant halten (CUTCONON, CUTCONOF)

Die Funktion "Werkzeugradiuskorrektur konstant halten" dient dazu, die Werkzeugradiuskorrektur für eine Anzahl von Sätzen zu unterdrücken, wobei jedoch eine durch die Werkzeugradiuskorrektur in vorhergehenden Sätzen aufgebaute Differenz zwischen der programmierten und der tatsächlich abgefahrenen Bahn des Werkzeugmittelpunkts als Verschiebung beibehalten wird. Sie kann z. B. dann vorteilhaft eingesetzt werden, wenn beim Zeilenfräsen in den Umkehrpunkten mehrere Verfahrsätze notwendig sind, die von der Werkzeugradiuskorrektur erzeugten Konturen (Umfahrungsstrategien) jedoch nicht erwünscht sind. Sie ist unabhängig von der Art der Werkzeugradiuskorrektur ($2^1/_2$D, 3D-Stirnfräsen, 3D-Umfangsfäsen) einsetzbar.

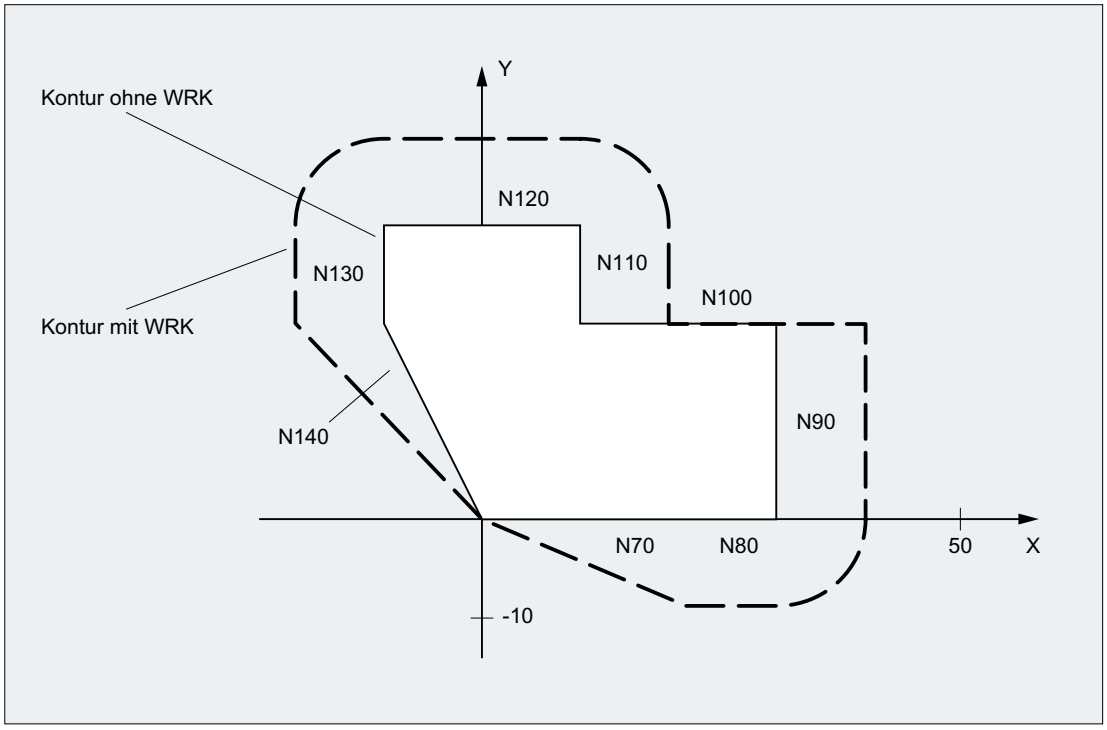
## Syntax

```
CUTCONON
CUTCONOF
```

## Bedeutung

| | |
|---|---|
| CUTCONON: | Befehl zum Einschalten der Funktion "Werkzeugradiuskorrektur konstant halten" |
| CUTCONOF: | Befehl zum Ausschalten der Funktion "Werkzeugradiuskorrektur konstant halten" |

## Beispiel




| **Programmcode** | **Kommentar** |
|---|---|
| N10 | ; Definition des Werkzeugs d1. |
| N20 $TC_DP1[1,1]= 110 | ; Typ |
| N30 $TC_DP6[1,1]= 10. | ; Radius |
| N40 | |
| N50 X0 Y0 Z0 G1 G17 T1 D1 F10000 | |
| N60 | |
| N70 X20 G42 NORM | |
| N80 X30 | |
| N90 Y20 | |
| N100 X10 CUTCONON | ; Einschalten der Korrekturunterdrückung. |

| Programmcode | Kommentar |
|---|---|
| N110 Y30 KONT | ; Beim Ausschalten der Konturunterdrückung ggf. Umfahrungskreis einfügen. |
| N120 X-10 CUTCONOF | |
| N130 Y20 NORM | ; Kein Umfahrungskreis beim Ausschalten der WRK. |
| N140 X0 Y0 G40 | |
| N150 M30 | |

### Weitere Informationen

Im Normalfall ist vor der Aktivierung der Korrekturunterdrückung die Werkzeugradiuskorrektur bereits aktiv, und sie ist noch aktiv, wenn die Korrekturunterdrückung wieder deaktiviert wird. Im letzen Verfahrsatz vor CUTCONON wird auf den Offsetpunkt im Satzendpunkt gefahren. Alle folgenden Sätze, in denen die Korrekturunterdrückung aktiv ist, werden ohne Korrektur verfahren. Sie werden dabei jedoch um den Vektor vom Endpunkt des letzten Korrektursatzes zu dessen Offsetpunkt verschoben. Der Interpolationstyp dieser Sätze (linear, zirkular, polynomisch) ist beliebig.

Der Deaktivierungssatz der Korrekturunterdrückung, d. h. der Satz, der CUTCONOF enthält, wird normal korrigiert. Er beginnt im Offsetpunkt des Startpunkts. Zwischen dem Endpunkt des Vorgängersatzes, d. h. des letzten programmierten Verfahrsatzes mit aktivem CUTCONON, und diesem Punkt wird ein linearer Satz eingefügt.

Kreissätze, bei denen die Kreisebene senkrecht auf der Korrekturebene steht (vertikale Kreise), werden so behandelt, als ob in ihnen CUTCONON programmiert wäre. Diese implizite Aktivierung der Korrekturunterdrückung wird im ersten Verfahrsatz, der eine Verfahrbewegung in der Korrekturebene enthält und der kein derartiger Kreis ist, automatisch rückgängig gemacht. Vertikale Kreise in diesem Sinne können nur beim Umfangsfräsen auftreten.

## 2.10.8 Werkzeuge mit relevanter Schneidenlage

Bei Werkzeugen mit relevanter Schneidenlage (Dreh- und Schleifwerkzeuge, Werkzeugtypen 400-599; siehe Kapitel "Vorzeichenbewertung Verschleiß" wird ein Wechsel von G40 nach G41/G42 bzw. umgekehrt wie ein Werkzeugwechsel betrachtet. Dies führt bei aktiver Transformation (z. B. TRANSMIT) zu einem Vorlaufstopp (Decodierungsstopp) und damit ggf. zu Abweichungen von der beabsichtigten Teilekontur.

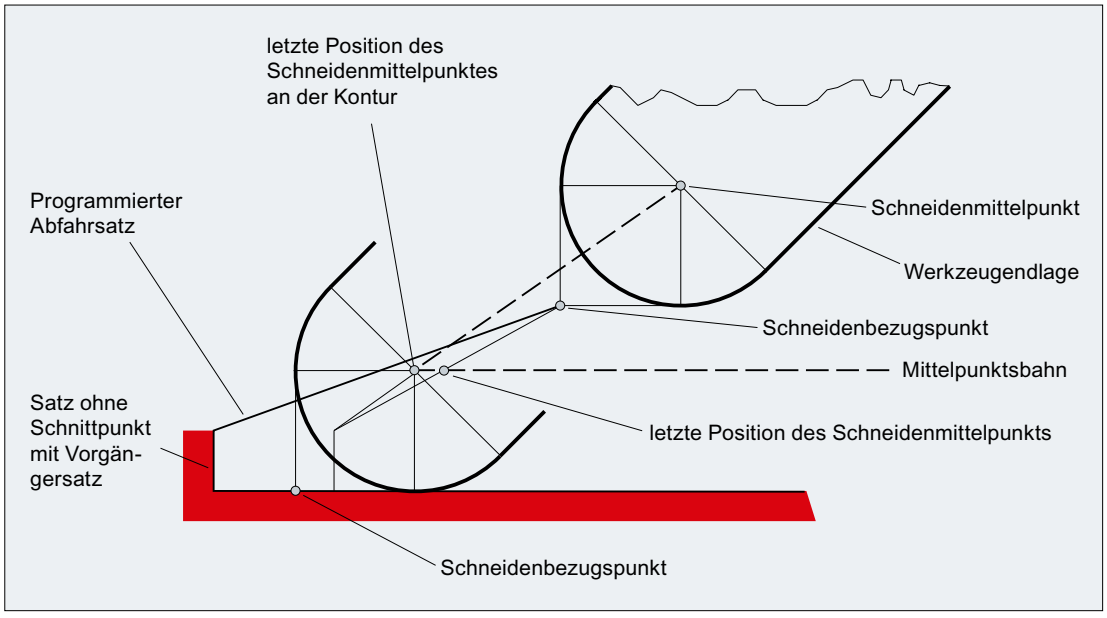
Diese ursprüngliche Funktionalität ändert sich hinsichtlich:

1. Vorlaufstopp bei TRANSMIT
2. Berechnung von Schnittpunkten beim An- bzw. Abfahren mit KONT
3. Wechsel eines Werkzeugs bei aktiver Werkzeugradiuskorrrektur
4. Werkzeugradiuskorrektur mit variabler Werkzeugorientierung bei Transformation

## Weitere Informationen

Die ursprüngliche Funktionalität wurde wie folgt geändert:

- Der Wechsel von G40 nach G41/G42 und umgekehrt wird nicht mehr als Werkzeugwechsel behandelt. Bei TRANSMIT kommt es deshalb nicht mehr zu einem Vorlaufstopp.
- Für die Berechnung von Schnittpunkten mit dem An- bzw. Abfahrsatz wird die Gerade zwischen den Schneidenmittelpunkten am Satzanfang und am Satzende verwendet. Die Differenz zwischen Schneidenbezugspunkt und Schneidenmittelpunkt wird dieser Bewegung überlagert.
  Beim An- bzw. Abfahren mit KONT (Werkzeug umfährt den Konturpunkt; siehe vorhergehenden Abschnitt "Kontur anfahren und verlassen") erfolgt die Überlagerung im linearen Teilsatz der An- bzw. Abfahrbewegung. Die geometrischen Verhältnisse sind deshalb bei Werkzeugen mit und ohne relevante Schneidenlage identisch. Unterschiede zum bisherigen Verhalten ergeben sich nur in den relativ seltenen Fällen, dass der An- bzw. Abfahrsatz mit einem nicht benachbarten Verfahrsatz einen Schnittpunkt bildet, siehe folgendes Bild:



- Der Wechsel eines Werkzeugs bei aktiver Werkzeugradiuskorrektur, bei dem sich der Abstand zwischen Schneidenmittelpunkt und Schneidenbezugspunkt ändert, ist in Kreissätzen und in Verfahrsätzen mit rationalen Polynomen mit einem Nennergrad > 4 verboten. Bei anderen Interpolationsarten ist ein Wechsel im Gegensatz zum bisherigen Zustand auch bei aktiver Transformation (z.B. TRANSMIT) zulässig.
- Bei der Werkzeugradiuskorrektur mit variabler Werkzeugorientierung ist die Transformation vom Schneidenbezugspunkt auf den Schneidenmittelpunkt nicht mehr durch eine einfache Nullpunktverschiebung realisierbar. Werkzeuge mit relevanter Schneidenlage werden deshalb beim 3D-Umfangsfräsen verboten (Alarm).

**Hinweis**

Für das Stirnfräsen ist das Thema nicht relevant, da hier ohnehin auch bisher nur definierte Werkzeugtypen ohne relevante Schneidenlage zugelassen sind. (Werkzeuge mit einem nicht ausdrücklich zugelassenen Werkzeugtyp werden als Kugelkopffräser mit dem angegebenen Radius behandelt. Die Angabe einer Schneidenlage wird ignoriert.)

# 2.11 Bahnfahrverhalten

## 2.11.1 Genauhalt (G60, G9, G601, G602, G603)

Genauhalt ist ein Verfahrmodus, bei dem am Ende eines jeden Verfahrsatzes alle an der Verfahrbewegung beteiligten Bahnachsen und Zusatzachsen, die nicht satzübergreifend verfahren, bis zum Stillstand abgebremst werden.

Genauhalt wird verwendet, wenn scharfe Außenecken hergestellt oder Innenecken auf Maß geschlichtet werden sollen.

Mit dem Genauhalt-Kriterium wird festgelegt, wie genau der Eckpunkt angefahren und wann zum nächsten Satz weitergeschaltet wird:

- "Genauhalt fein"
  Der Satzwechsel erfolgt, sobald für alle an der Verfahrbewegung beteiligten Achsen die achsspezifischen Toleranzgrenzen für "Genauhalt fein" erreicht sind.
  "Genauhalt fein" wird eingestellt über: MD36010 $MA_STOP_LIMIT_FINE[<Achse>]
- "Genauhalt grob"
  Der Satzwechsel erfolgt, sobald für alle an der Verfahrbewegung beteiligten Achsen die achsspezifischen Toleranzgrenzen für "Genauhalt grob" erreicht sind.
  "Genauhalt grob" wird eingestellt über: MD36000 $MA_STOP_LIMIT_COARSE[<Achse>]
- "Interpolator-Ende"
  Der Satzwechsel erfolgt, sobald die Steuerung für alle an der Verfahrbewegung beteiligten Achsen die Sollgeschwindigkeit Null errechnet hat. Die Istposition bzw. der Schleppabstand der beteiligten Achsen werden nicht betrachtet.

### Syntax

```
G60 ...
G9 ...
G601/G602/G603 ...
```

### Bedeutung

| | |
|---|---|
| G60: | Befehl zum Einschalten des **modal** wirksamen Genauhalts |
| G9: | Befehl zum Einschalten des **satzweise** wirksamen Genauhalts |
| G601: | Befehl zum Aktivieren von Genauhalt-Kriterium "**Genauhalt fein**" |
| G602: | Befehl zum Aktivieren von Genauhalt-Kriterium "**Genauhalt grob**" |
| G603: | Befehl zum Aktivieren von Genauhalt-Kriterium "**Interpolator-Ende**" |

**Hinweis**

Die Befehle zum Aktivieren der Genauhalt-Kriterien (G601 / G602 / G603) sind nur wirksam bei aktivem G60 oder G9.

## Beispiel

```
Programmcode            Kommentar
N5 G602                 ; Kriterium "Genauhalt grob" angewählt.
N10 G0 G60 Z...         ; Genauhalt modal aktiv.
N20 X... Z...           ; G60 wirkt weiterhin.
...
N50 G1 G601             ; Kriterium "Genauhalt fein" angewählt.
N80 G64 Z...            ; Umschalten auf Bahnsteuerbetrieb.
...
N100 G0 G9              ; Genauhalt wirkt nur in diesen Satz.
N110 ...                ; Bahnsteuerbetrieb wieder aktiv.
```

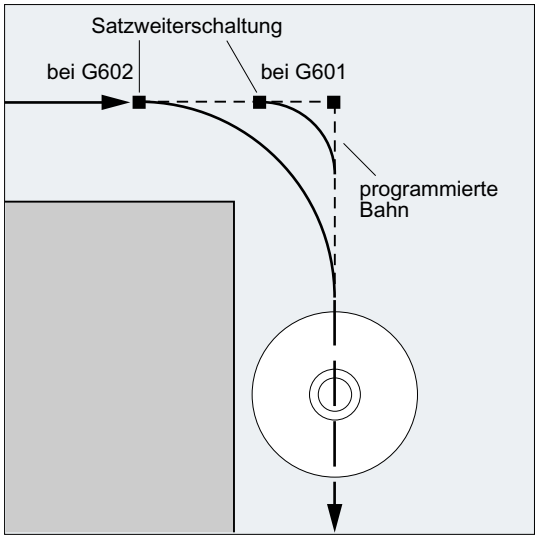
## Weitere Informationen

### G60, G9

G9 erzeugt im aktuellen Satz den Genauhalt, G60 im aktuellen Satz und in allen nachfolgenden Sätzen.

Mit den Bahnsteuerbetrieb-Befehlen G64 oder G641 ... G645 wird G60 ausgeschaltet.

### G601, G602



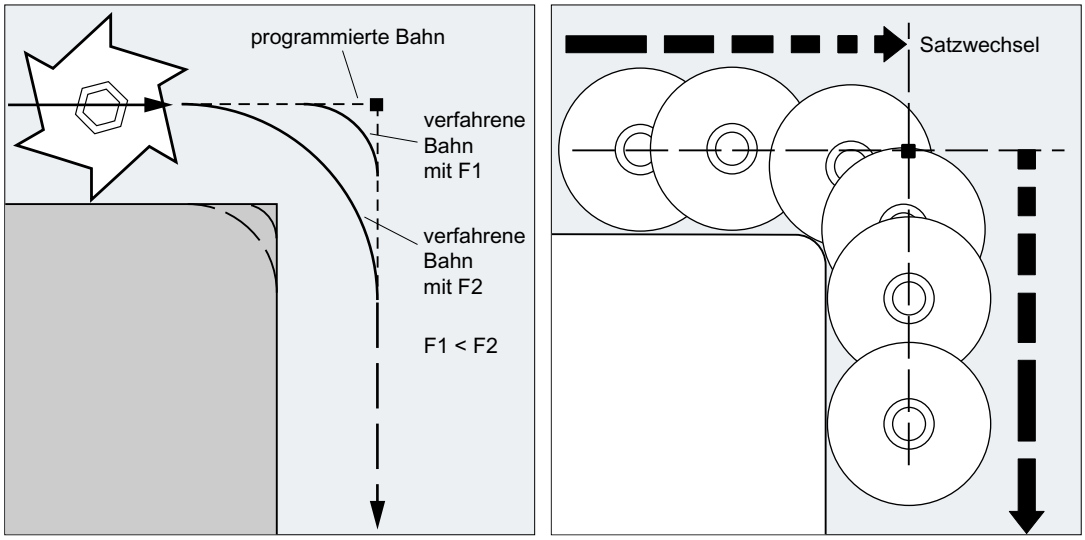

Die Bewegung wird abgebremst und am Eckpunkt kurz angehalten.

**Hinweis**

Die Grenzen für die Genauhalt-Kriterien sollten nur so eng wie nötig gesetzt sein. Je enger die Grenzen gefasst sind, desto länger dauern der Lageabgleich und das Anfahren der Zielposition.

### G603

Der Satzwechsel wird eingeleitet, wenn die Steuerung für die beteiligten Achsen die Sollgeschwindigkeit Null errechnet hat. Zu diesem Zeitpunkt liegt der Istwert – abhängig von der Dynamik der Achsen und der Bahngeschwindigkeit – um einen Nachlaufanteil zurück. Hierdurch lassen sich Werkstückecken verschleifen.




**Projektiertes Genauhalt-Kriterium**

Für G0 und die übrigen Befehle der 1. G-Gruppe kann kanalspezifisch hinterlegt sein, dass abweichend vom programmierten Genauhalt-Kriterium automatisch ein voreingestelltes Kriterium verwendet wird (siehe Angaben des Maschinenherstellers!).

## 2.11.2 Bahnsteuerbetrieb (G64, G641, G642, G643, G644, G645, ADIS, ADISPOS)

Im Bahnsteuerbetrieb wird die Bahngeschwindigkeit am Satzende zum Satzwechsel nicht auf eine Geschwindigkeit abgebremst, die ein Erreichen des Genauhaltkriteriums ermöglicht. Ziel ist dagegen, ein größeres Abbremsen der Bahnachsen am Satzwechselpunkt zu vermeiden, um mit möglichst gleicher Bahngeschwindigkeit in den nächsten Satz zu wechseln. Um dieses Ziel zu erreichen, wird mit Anwahl des Bahnsteuerbetriebs zusätzlich die Funktion "Vorausschauende Geschwindigkeitsführung (LookAhead)" aktiviert.

Bahnsteuerbetrieb mit Überschleifen bedeutet, dass knickförmige Satzübergänge durch lokale Änderungen des programmierten Verlaufs tangential gestaltet bzw. geglättet werden.

Bahnsteuerbetrieb bewirkt:

- eine Verrundung der Kontur
- kürzere Bearbeitungszeiten durch fehlende Brems- und Beschleunigungsvorgänge, die für das Erreichen des Genauhaltkriteriums benötigt werden.
- bessere Schnittbedingungen durch den gleichmäßigeren Geschwindigkeitsverlauf.

Bahnsteuerbetrieb ist sinnvoll, wenn:

- eine Kontur möglichst zügig abgefahren werden soll (z. B. mit Eilgang).
- der exakte Verlauf im Rahmen eines Fehlerkriteriums vom programmierten abweichen darf, um einen durchgehend stetigen Verlauf zu erzeugen.

Bahnsteuerbetrieb ist nicht sinnvoll, wenn:

- eine Kontur exakt abgefahren werden soll.
- absolute Geschwindigkeitskonstanz erforderlich ist.

**Hinweis**

Der Bahnsteuerbetrieb wird durch Sätze unterbrochen, die implizit Vorlaufstopp auslösen, z. B. durch:

- Zugriff auf bestimmte Zustandsdaten der Maschine ($A...)
- Hilfsfunktionsausgaben

## Syntax

```
G64 ...
G641 ADIS=…
G641 ADISPOS=…
G642 ...
G643 ...
G644 ...
G645 ...
```

## Bedeutung

| | |
|---|---|
| G64: | Bahnsteuerbetrieb mit Geschwindigkeitsabsenkung gemäß Überlastfaktor |
| G641: | Bahnsteuerbetrieb mit Überschleifen nach Wegkriterium |
| ADIS=... : | Wegkriterium bei G641 für die Bahnfunktionen G1, G2, G3, … |
| ADISPOS=... : | Wegkriterium bei G641 für Eilgang G0 |
| | Das Wegkriterium (= Überschleifabstand) ADIS bzw. ADISPOS beschreibt die Strecke, die der Überschleifsatz vor dem Satzende frühestens beginnen darf, bzw. die Strecke nach Satzende, in der der Überschleifsatz beendet sein muss.<br>**Hinweis:**<br>Wenn kein ADIS/ADISPOS programmiert wird, dann gilt der Wert "Null" und damit das Fahrverhalten wie bei G64. Bei kurzen Fahrwegen wird der Überschleifabstand automatisch reduziert (bis max. 36 %). |

| | |
|---|---|
| G642: | Bahnsteuerbetrieb mit Überschleifen unter Einhaltung definierter Toleranzen<br>In diesem Modus erfolgt das Überschleifen im Normalfall unter Einhaltung der maximal erlaubten Bahnabweichung. Anstelle dieser achsspezifischen Toleranz kann aber auch die Einhaltung der maximalen Konturabweichung (Konturtoleranz) oder der maximalen Winkelabweichung der Werkzeugorientierung (Orientierungstoleranz) konfiguriert sein.<br>**Hinweis:**<br>Die Erweiterung um Kontur- und Orientierungstoleranz existiert nur in Systemen mit vorhandener Option "Polynominterpolation". |
| G643: | Bahnsteuerbetrieb mit Überschleifen unter Einhaltung definierter Toleranzen (satzintern)<br>Mit G643 wird im Gegensatz zu G642 kein eigener Überschleifsatz gebildet, sondern es werden achsspezifisch satzinterne Überschleifbewegungen eingefügt. Der Überschleifweg kann für jede Achse unterschiedlich sein. |
| G644: | Bahnsteuerbetrieb mit Überschleifen mit maximal möglicher Dynamik<br>**Hinweis:**<br>G644 ist bei aktiver Kinematischer Transformation nicht möglich. Es wird intern auf G642 umgeschaltet. |
| G645: | Bahnsteuerbetrieb mit Überschleifen von Ecken und tangentialer Satzübergänge unter Einhaltung definierter Toleranzen<br>G645 arbeitet an Ecken gleich wie G642. Mit G645 werden lediglich auch bei tangentialen Satzübergängen Überschleifsätze gebildet, wenn der Krümmungsverlauf der Originalkontur in mindestens einer Achse einen Sprung aufweist. |

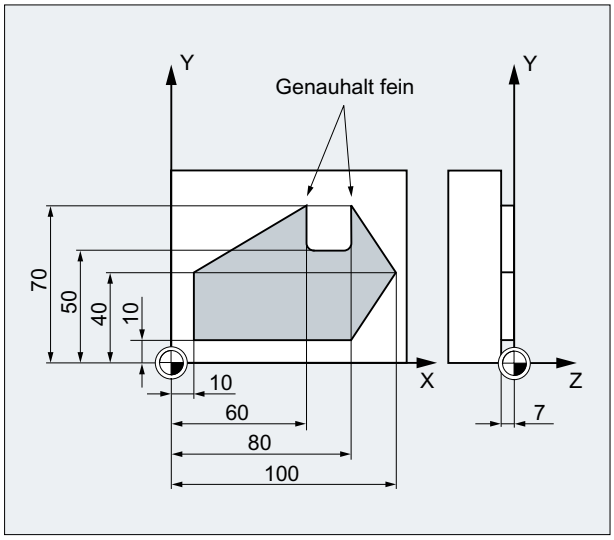
**Hinweis**

Überschleifen ist kein Ersatz für Eckenrunden (RND). Der Anwender sollte keine Annahme darüber treffen, wie die Kontur innerhalb des Überschleifbereichs aussieht. Insbesondere kann die Art des Überschleifens auch von dynamischen Gegebenheiten, z. B. der Bahngeschwindigkeit abhängen. Überschleifen an der Kontur macht daher nur mit kleinen ADIS-Werten einen Sinn. Wenn an der Ecke eine definierte Kontur gefahren werden soll, muss RND verwendet werden.

**Hinweis**

Wird eine durch G641, G642, G643, G644 oder G645 erzeugte Überschleifbewegung unterbrochen, wird beim nachfolgenden Repositionieren (REPOS) nicht der Unterbrechungspunkt angefahren, sondern der Anfangs- oder Endpunkt des originalen Verfahrsatzes (je nach REPOS-Modus).

## Beispiel



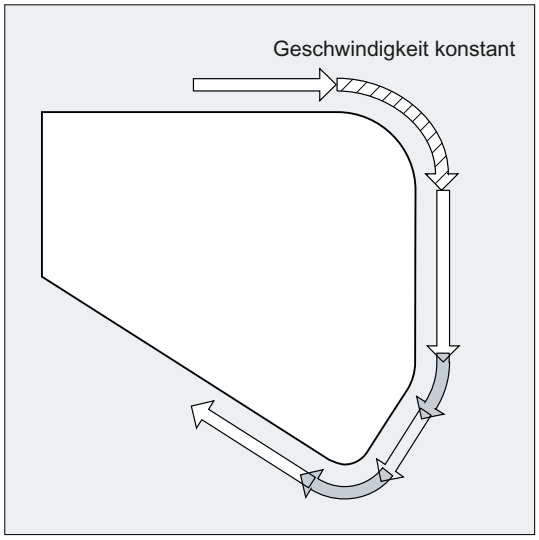

Die beiden Außenecken an der Nut sollen exakt angefahren werden. Ansonsten soll im Bahnsteuerbetrieb gefertigt werden.

| **Programmcode** | **Kommentar** |
|---|---|
| N05 DIAMOF | ; Radius als Maßangabe. |
| N10 G17 T1 G41 G0 X10 Y10 Z2 S300 M3 | ; Startposition anfahren, Spindel einschalten, Bahnkorrektur. |
| N20 G1 Z-7 F8000 | ; Werkzeug zustellen. |
| N30 G641 ADIS=0.5 | ; Konturübergänge werden verschliffen. |
| N40 Y40 | |
| N50 X60 Y70 G60 G601 | ; Position exakt mit Genauhalt fein anfahren. |
| N60 Y50 | |
| N70 X80 | |
| N80 Y70 | |
| N90 G641 ADIS=0.5 X100 Y40 | ; Konturübergänge werden verschliffen. |
| N100 X80 Y10 | |
| N110 X10 | |
| N120 G40 G0 X-20 | ; Bahnkorrektur ausschalten. |
| N130 Z10 M30 | ; Werkzeug wegfahren, Programmende. |

## Weitere Informationen

### Bahnsteuerbetrieb G64

Im Bahnsteuerbetrieb fährt das Werkzeug bei tangentialen Konturübergängen mit möglichst konstanter Bahngeschwindigkeit (kein Abbremsen an den Satzgrenzen). Vor Ecken und Sätzen mit Genauhalt wird vorausschauend gebremst (LookAhead).




Ecken werden ebenfalls stetig umfahren. Zur Verringerung des Konturfehlers wird die Geschwindigkeit unter Berücksichtigung einer Beschleunigungsgrenze und eines Überlastfaktors entsprechend reduziert.

**Hinweis**

Wie stark die Konturübergänge verschliffen werden, hängt von der Vorschubgeschwindigkeit und dem Überlastfaktor ab. Der Überlastfaktor ist im MD32310 $MA_MAX_ACCEL_OVL_FACTOR einstellbar.

Durch Setzen von MD20490 $MC_IGNORE_OVL_FACTOR_FOR_ADIS werden Satzübergänge immer unabhängig vom eingestellten Überlastfaktor überschliffen.

Um einen unerwünschten Stopp der Bahnbewegung zu vermeiden (Freischneiden!), sind folgende Punkte zu beachten:

- Hilfsfunktionen, die nach Bewegungsende oder vor der nächsten Bewegung geschaltet werden, unterbrechen den Bahnsteuerbetrieb (Ausnahme: Schnelle Hilfsfunktionen).
- Positionierachsen fahren immer nach dem Genauhaltprinzip, Positionierfenster fein (wie G601). Falls in einem NC-Satz auf Positionierachsen gewartet werden muss, wird der Bahnsteuerbetrieb der Bahnachsen unterbrochen.

Zwischenprogrammierte Sätze nur mit Kommentaren, Rechensätzen oder Unterprogrammaufrufen stören den Bahnsteuerbetrieb dagegen nicht.

**Hinweis**

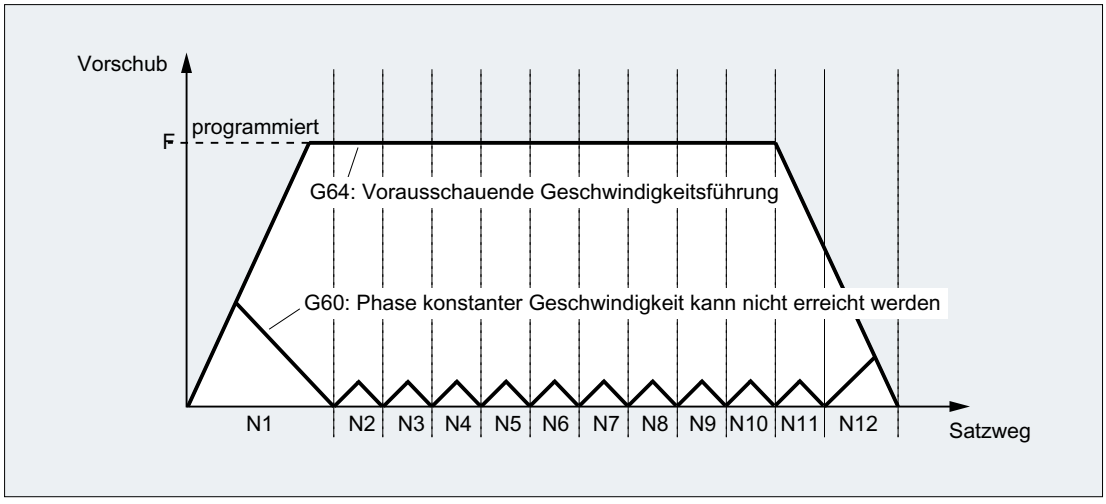
Sind nicht alle Bahnachsen in FGROUP enthalten, wird es an Satzübergängen für die nicht enthaltenen Achsen häufig einen Geschwindigkeitssprung geben, den die Steuerung durch ein Absenken der Geschwindigkeit am Satzwechsel auf den durch MD32300 $MA_MAX_AX_ACCEL und MD32310 $MA_MAX_ACCEL_OVL_FACTOR erlaubten Wert begrenzt. Dieses Abbremsen lässt sich vermeiden, wenn man den vorgegebenen Positionszusammenhang der Bahnachsen durch ein Überschleifen aufweicht.

### Vorausschauende Geschwindigkeitsführung LookAhead

Im Bahnsteuerbetrieb ermittelt die Steuerung automatisch für mehrere NC-Sätze im voraus die Geschwindigkeitsführung. Hierdurch kann bei annähernd tangentialen Übergängen über mehrere Sätze hinweg beschleunigt und gebremst werden.

Vor allem Bewegungsketten, die sich aus kurzen Fahrwegen zusammensetzen, lassen sich durch vorausschauende Geschwindigkeitsführung mit hohen Bahnvorschüben herstellen.

Die Anzahl der NC-Sätze, über die maximal vorausgeschaut wird, lässt sich über Maschinendatum einstellen.




### Bahnsteuerbetrieb mit Überschleifen nach Wegkriterium (G641)

Bei G641 fügt die Steuerung an Konturübergängen Übergangselemente ein. Mit dem Überschleifabstand ADIS (bzw. ADISPOS bei G0) wird angegeben, wie stark die Ecken maximal verschliffen werden dürfen. Innerhalb des Überschleifabstandes ist die Steuerung frei, den Bahnzusammenhang aufzulösen und durch einen dynamisch optimalen Weg zu ersetzen.

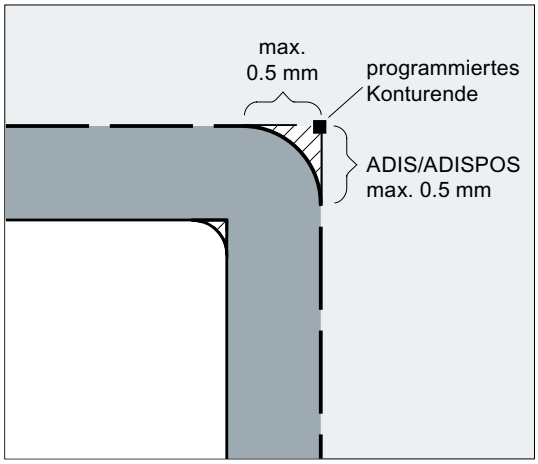
Nachteil: Für alle Achsen steht nur ein ADIS-Wert zur Verfügung.

G641 wirkt ähnlich wie RNDM, ist aber nicht auf die Achsen der Arbeitsebene beschränkt.

Wie G64 arbeitet G641 mit vorausschauender Geschwindigkeitsführung LookAhead. Überschleifsätze mit hoher Krümmung werden mit verringerter Geschwindigkeit angefahren.

Beispiel:

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 G641 ADIS=0.5 G1 X... Y...` | `; Der Überschleifsatz darf frühestens 0,5 mm vor dem programmierten Satzende beginnen und muss 0,5 mm nach Satzende beendet sein. Diese Einstellung bleibt modal wirksam.` |




**Hinweis**

Überschleifen kann und soll die Funktionen für definiertes Glätten (RND, RNDM, ASPLINE, BSPLINE, CSPLINE) nicht ersetzen.

### Überschleifen mit axialer Genauigkeit bei G642

Bei G642 findet das Überschleifen nicht innerhalb eines definierten ADIS-Bereichs statt, sondern es werden die mit MD33100 $MA_COMPRESS_POS_TOL definierten axialen Toleranzen eingehalten. Der Überschleifweg wird aus dem kürzesten Überschleifweg aller Achsen bestimmt. Dieser Wert wird bei der Erzeugung eines Überschleifsatzes berücksichtigt.

### Satzinternes Überschleifen bei G643

Die maximalen Abweichungen von der exakten Kontur werden beim Überschleifen mit G643 durch das Maschinendatum MD33100 $MA_COMPRESS_POS_TOL für jede Achse festgelegt.

Mit G643 wird kein eigener Überschleifsatz gebildet, sondern es werden achsspezifisch satzinterne Überschleifbewegungen eingefügt. Bei G643 kann der Überschleifweg jeder Achse unterschiedlich sein.

### Überschleifen mit Kontur- und Orientierungstoleranz bei G642/G643

Mit MD20480 $MC_SMOOTHING_MODE kann das Überschleifen mit G642 und G643 so konfiguriert werden, dass anstelle der achsspezifischen Toleranzen eine Konturtoleranz und eine Orientierungstoleranz wirksam sind.

Kontur- und Orientierungstoleranz werden eingestellt in den kanalspezifischen Settingdaten:

SD42465 $SC_SMOOTH_CONTUR_TOL (Maximale Konturabweichung)

SD42466 $SC_SMOOTH_ORI_TOL (Maximale Winkelabweichung der Werkzeugorientierung)

Die Settingdaten können im NC-Programm programmiert und dadurch für jeden Satzübergang anders vorgegeben werden. Sehr unterschiedliche Vorgaben für die Konturtoleranz und die Orientierungstoleranz können sich nur bei G643 auswirken.

**Hinweis**

Die Erweiterung um Konturtoleranz und Orientierungstoleranz existiert nur in Systemen mit vorhandener Option "Polynominterpolation".

**Hinweis**

Für das Überschleifen unter Einhaltung der Orientierungstoleranz muss eine Orientierungstransformation aktiv sein.

### Überschleifen mit maximal möglicher Dynamik bei G644

Das Überschleifen mit maximal möglicher Dynamik wird mit MD20480 $MC_SMOOTHING_MODE in der Tausenderstelle konfiguriert:

| Wert | Bedeutung |
|---|---|
| 0 | Vorgabe der maximalen axialen Abweichungen mit:<br>MD33100 $MA_COMPRESS_POS_TOL |
| 1 | Vorgabe des maximalen Überschleifwegs durch Programmierung von:<br>ADIS=... bzw. ADISPOS=... |
| 2 | Vorgabe der maximal auftretenden Frequenzen jeder Achse im Überschleifbereich mit:<br>MD32440 $MA_LOOKAH_FREQUENCY<br>Der Überschleifbereich wird so festgelegt, dass bei der Überschleifbewegung keine Frequenzen auftreten, die die vorgegebene maximale Frequenz überschreiten. |
| 3 | Beim Überschleifen mit G644 werden weder die Toleranz noch der Überschleifabstand überwacht. Jede Achse fährt mit maximal möglicher Dynamik um eine Ecke.<br>Bei SOFT wird hierbei sowohl die maximale Beschleunigung als auch der maximale Ruck jeder Achse eingehalten.<br>Bei BRISK wird der Ruck nicht begrenzt, sondern jede Achse fährt mit maximal möglicher Beschleunigung. |

### Überschleifen tangentialer Satzübergänge bei G645

Die Überschleifbewegung bei G645 wird so festgelegt, dass alle beteiligten Achsen keinen Sprung in der Beschleunigung erfahren und die parametrierten maximalen Abweichungen zur Originalkontur (MD33120 $MA_PATH_TRANS_POS_TOL) nicht überschritten werden.

Bei knickförmigen, nicht-tangentialen Satzübergängen ist das Überschleifverhalten wie bei G642.

### Keine Überschleifzwischensätze

In den folgenden Fällen wird kein Überschleifzwischensatz eingefügt:

- Zwischen beiden Sätzen wird angehalten.
  Dies tritt auf, wenn:
  - eine Hilfsfunktionsausgabe vor Bewegung im Folgesatz steht.
  - der Folgesatz keine Bahnbewegung enthält.
  - für den Folgesatz zum ersten Mal eine Achse als Bahnachse verfährt, die zuvor Positionierachse war.
  - für den Folgesatz zum ersten Mal eine Achse als Positionierachse verfährt, die zuvor Bahnachse war.
  - der Vorgängersatz Geometrieachsen verfährt und der Folgesatz nicht.
  - der Folgesatz Geometrieachsen verfährt und der Vorgängersatz nicht.
  - vor Gewindeschneiden der Folgesatz G33 als Wegbedingung hat und der Vorgängersatz nicht.
  - zwischen BRISK und SOFT gewechselt wird.
  - transformationsbedeutsame Achsen nicht vollständig der Bahnbewegung zugeordnet sind (z. B. bei Pendeln, Positionierachsen).
- Der Überschleifsatz würde die Teileprogrammbearbeitung verlangsamen.
  Dies tritt auf:
  - zwischen sehr kurzen Sätzen.
    Da jeder Satz mindestens einen Interpolatortakt benötigt, würde der eingefügte Zwischensatz die Bearbeitungszeit verdoppeln.
  - wenn ein Satzübergang mit G64 (Bahnsteuerbetrieb ohne Überschleifen) ohne Geschwindigkeitsreduzierung überfahren werden darf.
    Überschleifen würde die Bearbeitungszeit erhöhen. D. h. der Wert des erlaubten Überlastfaktors (MD32310 $MA_MAX_ACCEL_OVL_FACTOR) hat Einfluss darauf, ob ein Satzübergang überschliffen wird oder nicht. Der Überlastfaktor wird nur beim Überschleifen mit G641 / G642 berücksichtigt. Beim Überschleifen mit G643 hat der Überlastfaktor keinen Einfluss (dieses Verhalten kann auch für G641 und G642 eingestellt werden, indem
    MD20490 $MC_IGNORE_OVL_FACTOR_FOR_ADIS = TRUE gesetzt wird).

- Das Überschleifen ist nicht parametriert.
  Dies tritt auf, wenn:
  - bei G641 in G0-Sätzen ADISPOS=0 ist (Vorbelegung!).
  - bei G641 in Nicht-G0-Sätzen ADIS=0 ist (Vorbelegung!).
  - bei G641 beim Übergang zwischen G0 und Nicht-G0 bzw. Nicht-G0 und G0 der kleinere Wert aus ADISPOS und ADIS gilt.
  - bei G642/G643 alle achsspezifischen Toleranzen gleich Null sind.
- Der Satz enthält keine Verfahrbewegung (Nullsatz).
  Dies tritt auf, wenn:
  - Synchronaktionen aktiv sind.
    Normalerweise werden Nullsätze vom Interpreter eliminiert. Wenn aber Synchronaktionen aktiv sind, wird dieser Nullsatz eingekettet und ausgeführt. Hierbei wird ein Genauhalt entsprechend aktiver Programmierung ausgelöst. Damit soll die Synchronaktion die Möglichkeit bekommen, gegebenenfalls zu schalten.
  - durch Programmsprünge Nullsätze erzeugt werden.

### Bahnsteuerbetrieb im Eilgang G0

Auch für das Fahren im Eilgang muss eine der genannten Funktionen G60/G9 oder G64 bzw. G641 - G645 angegeben werden. Ansonsten wirkt die über Maschinendatum eingegebene Voreinstellung.

# 2.12 Koordinatentransformationen (Frames)

## 2.12.1 Frames

### Frame

Der Frame ist eine in sich geschlossene Rechenvorschrift, die ein kartesisches Koordinatensystem in ein anderes kartesisches Koordinatensystem überführt.

### Basisframe (Basisverschiebung)

Der Basisframe beschreibt die Koordinatentransformation vom Basiskoordinatensystem (BKS) in das Basis-Nullpunktsystem (BNS) und wirkt wie die einstellbaren Frames.

Siehe Basis-Koordinatensystem (BKS) (Seite 38) .

### Einstellbare Frames

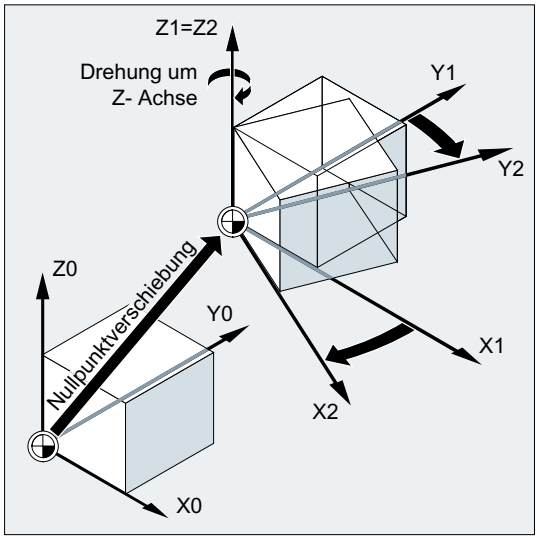
Einstellbare Frames sind die mit den Befehlen `G54` bis `G57` und `G505` bis `G599` aus jedem beliebigen NC-Programm abrufbaren einstellbaren Nullpunktverschiebungen. Die Verschiebewerte werden vom Bediener voreingestellt und im Nullpunktspeicher der Steuerung abgespeichert. Mit ihnen wird das einstellbare Nullpunktsystem (ENS) festgelegt.

Siehe:

- Einstellbares Nullpunktsystem (ENS) (Seite 40)
- Einstellbare Nullpunktverschiebung (G54 ... G57, G505 ... G599, G53, G500, SUPA, G153) (Seite 147)

## Programmierbare Frames

Manchmal erweist es sich als sinnvoll bzw. notwendig, innerhalb eines NC-Programms das ursprünglich gewählte Werkstück-Koordinatensystem (bzw. das "Einstellbare Nullpunktsystem") an eine andere Stelle zu verschieben und ggf. zu drehen, zu spiegeln und / oder zu skalieren. Dies erfolgt über programmierbare Frames.



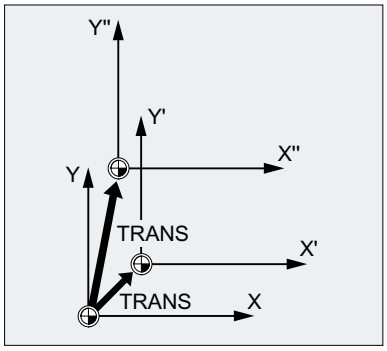

Siehe Frame-Anweisungen (Seite 310) .

## 2.12.2 Frame-Anweisungen

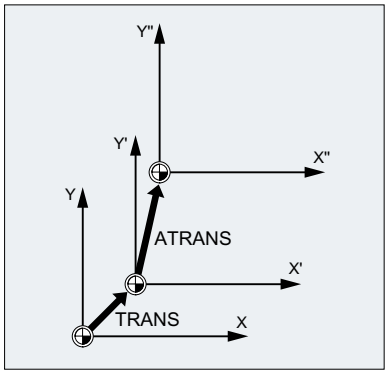
### Funktion

Die Anweisungen für die programmierbaren Frames gelten im aktuellen NC-Programm. Sie wirken entweder additiv oder ersetzend:

- Ersetzende Anweisung
  Löscht alle zuvor programmierten Frame-Anweisungen. Als Bezug gilt die zuletzt aufgerufene einstellbare Nullpunktverschiebung (G54 ... G57, G505 ... G599).

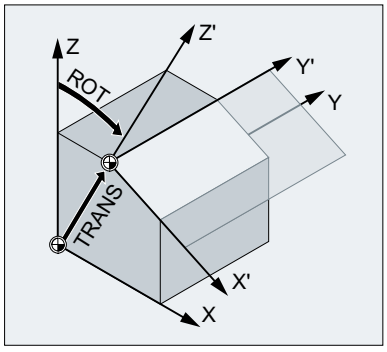



- Additive Anweisung
  Setzt auf bereits bestehenden Frames auf. Als Bezug dient der aktuell eingestellte oder über eine Frame-Anweisung zuletzt programmierte Werkstück-Nullpunkt.




#### Anwendungsbeispiel

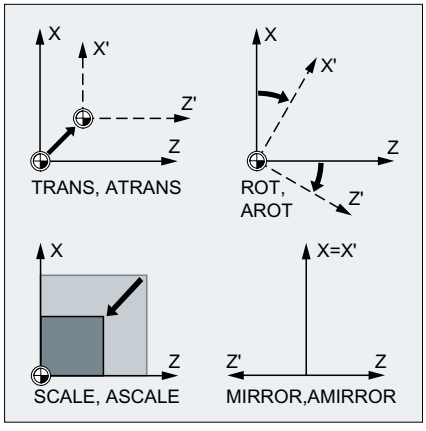
1. Verschieben des Nullpunkts des Werkstückkoordinatensystems (WKS).
2. Drehen des Werkstückkoordinatensystems (WKS) zum Ausrichten einer Ebene parallel zur gewünschten Arbeitsebene.

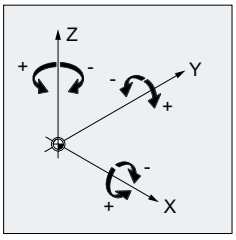



## Syntax

| Ersetzende Anweisungen | Additive Anweisungen |
|---|---|
| TRANS X… Y… Z… | ATRANS X… Y… Z… |
| ROT X… Y… Z… | AROT X… Y… Z… |
| ROT RPL=… | AROT RPL=… |
| ROTS/CROTS X... Y... | AROTS X... Y... |
| SCALE X… Y… Z… | ASCALE X… Y… Z… |
| MIRROR X0/Y0/Z0 | AMIRROR X0/Y0/Z0 |

## Bedeutung



| | | | |
|---|---|---|---|
| TRANS/ATRANS: | WKS-Verschiebung in Richtung der angegebenen Geometrieachse(n) | | |
| ROT/AROT: | WKS-Drehung:<br>• durch die Verkettung von Einzeldrehungen um die angegebenen Geometrieachse(n)<br>oder<br>• um den Winkel RPL=... in der aktuellen Arbeitsebene (G17/G18/G19) | | |
| | Drehrichtung: |  | |
| | Drehreihenfolge: | mit RPY-Notation: | Z, Y', X'' |
| | | mit Eulerwinkel: | Z, X', Z'' |
| | Wertebereich: | Die Drehwinkel sind nur eindeutig in den folgenden Bereichen definiert: | |
| | | mit RPY-Notation: | -180 ≤ x ≤ 180 |
| | | | -90 < y < 90 |
| | | | -180 ≤ z ≤ 180 |
| | | mit Eulerwinkel: | 0 ≤ x < 180 |
| | | | -180 ≤ y ≤ 180 |
| | | | -180 ≤ z ≤ 180 |
| ROTS/AROTS: | WKS-Drehung durch die Angabe von Raumwinkeln<br>Die Orientierung einer Ebene im Raum ist durch die Angabe zweier Raumwinkel eindeutig bestimmt. Es dürfen deshalb maximal 2 Raumwinkel programmiert werden:<br>ROTS/AROTS X... Y... / Z... X... / Y... Z... | | |
| CROTS: | CROTS wirkt wie ROTS, bezieht sich aber auf das gültige Frame in der Datenhaltung. | | |
| SCALE/ASCALE: | Skalierung in Richtung der angegebenen Geometrieachse(n) zum Vergrößern/Verkleinern einer Kontur | | |
| MIRROR/AMIRROR: | WKS-Spiegelung durch Spiegelung (Richtungswechsel) der angegebenen Geometrieachse | | |
| | Wert: | frei wählbar (hier: "0") | |

## Randbedingungen

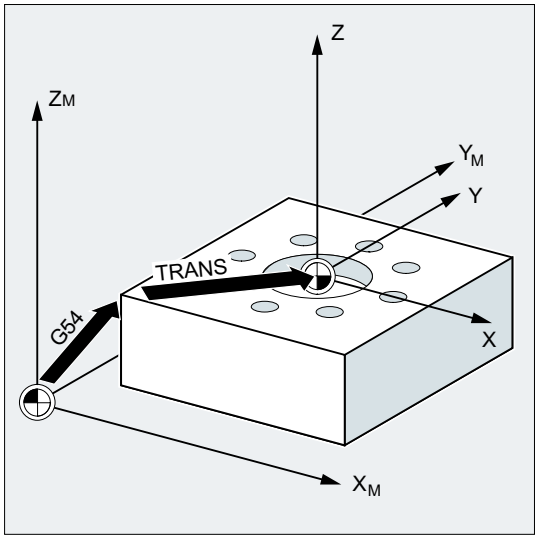
- Frame-Anweisungen müssen in einem eigenen NC-Satz programmiert werden.
- Frame-Anweisungen können einzeln angewendet oder beliebig kombiniert werden.
- Frame-Anweisungen werden in der programmierten Reihenfolge ausgeführt.
- Additive Anweisungen werden häufig in Unterprogrammen eingesetzt. Die im Hauptprogramm definierten Basisanweisungen bleiben nach Unterprogrammende erhalten, wenn das Unterprogramm mit dem SAVE-Attribut programmiert wurde.

## 2.12.3 Programmierbare Nullpunktverschiebung (TRANS, ATRANS)

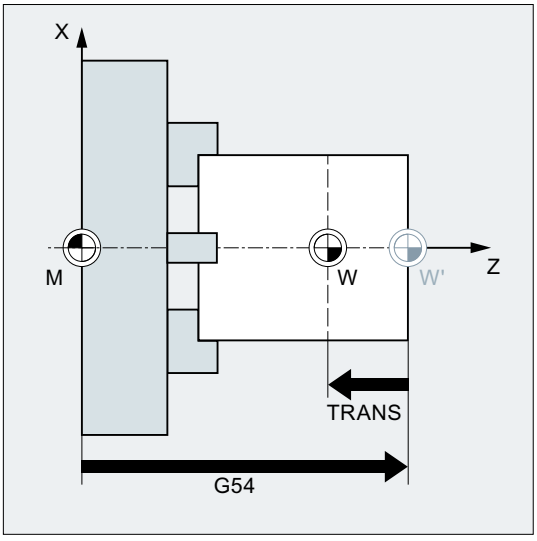
Mit dem Befehl `TRANS` wird das WKS absolut, bezogen auf das mit einer einstellbaren Nullpunktverschiebung (G54 ... G57, G505 ... G599) erzeugte ENS, verschoben.

Mit dem Befehl `ATRANS` wird das mit `TRANS` erzeugte WKS additiv verschoben.

Fräsen:




Drehen:




### Syntax

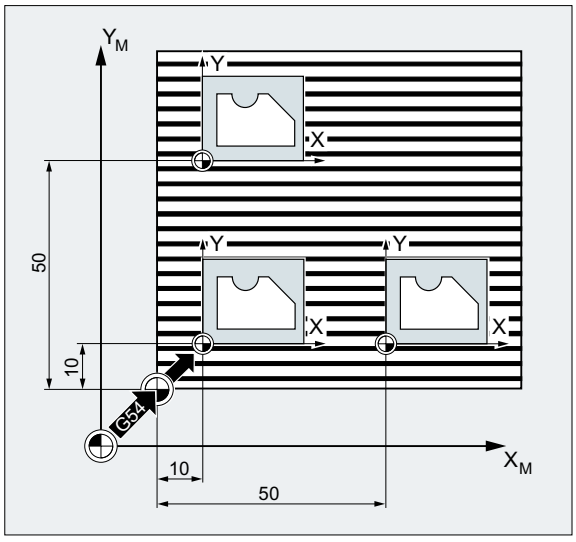
```
TRANS X… Y… Z…
ATRANS X… Y… Z…
```

### Bedeutung

| | | |
|---|---|---|
| `TRANS`: | Absolute Verschiebung des WKS, bezogen auf den mit einer einstellbaren Nullpunktverschiebung (G54 ... G57, G505 ... G599) eingestellten Werkstücknullpunkt (ENS) | |
| | Alleine im Satz: | ja |
| `ATRANS`: | Additive Nullpunktverschiebung des WKS, bezogen auf den mit `TRANS` eingestellten Werkstücknullpunkt | |
| | Alleine im Satz: | ja |
| `X... Y... Z...`: | Verschiebungswerte in Richtung der angegebenen Geometrieachsen | |

## Beispiele

### Beispiel 1: Fräsen



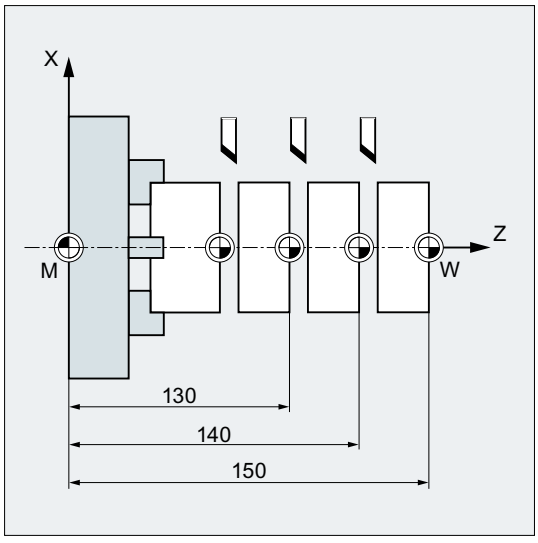

Bei diesem Werkstück kommen die gezeigten Formen in einem Programm mehrfach vor.

Die Bearbeitungsfolge für diese Form ist im Unterprogramm abgelegt.

Durch Nullpunktverschiebung werden die jeweils benötigten Werkstücknullpunkte gesetzt und dann das Unterprogramm aufgerufen.

| **Programmcode** | **Kommentar** |
|---|---|
| N10 G1 G54 | ; Arbeitsebene X/Y, Werkstücknullpunkt |
| N20 G0 X0 Y0 Z2 | ; Startpunkt anfahren |
| N30 TRANS X10 Y10 | ; Absolute Verschiebung |
| N40 L10 | ; Unterprogramm-Aufruf |
| N50 TRANS X50 Y10 | ; Absolute Verschiebung |
| N60 L10 | ; Unterprogramm-Aufruf |
| N70 M30 | ; Programmende |

### Beispiel 2: Drehen




| Programmcode | Kommentar |
|---|---|
| ... | |
| N10 TRANS X0 Z150 | ; Absolute Verschiebung |
| N15 L20 | ; Unterprogramm-Aufruf |
| N20 TRANS X0 Z140 (oder ATRANS Z-10) | ; Absolute Verschiebung |
| N25 L20 | ; Unterprogramm-Aufruf |
| N30 TRANS X0 Z130 (oder ATRANS Z-10) | ; Absolute Verschiebung |
| N35 L20 | ; Unterprogramm-Aufruf |
| ... | |

## Weitere Informationen

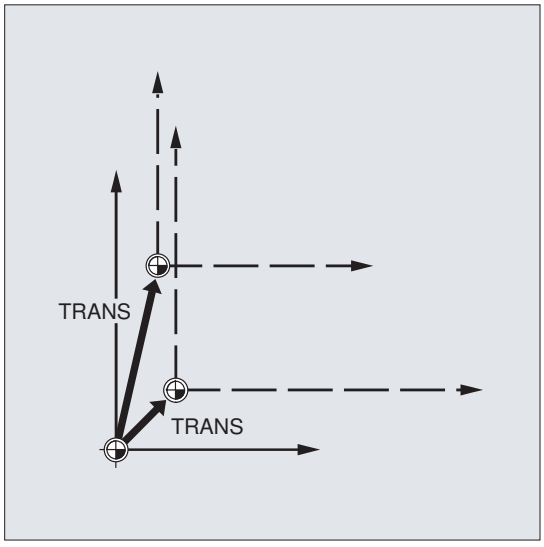
### TRANS X... Y... Z...

Nullpunktverschiebung um die in den jeweils angegebenen Achsrichtungen (Bahn-, Synchron- und Positionierachsen) programmierten Verschiebewerte. Als Bezug gilt die zuletzt angegebene einstellbare Nullpunktverschiebung (G54 ... G57, G505 ... G599).

**ACHTUNG**

**Kein Ursprungs-Frame**

Der Befehl `TRANS` setzt alle Frame-Komponenten des vorher gesetzten programmierbaren Frames zurück.

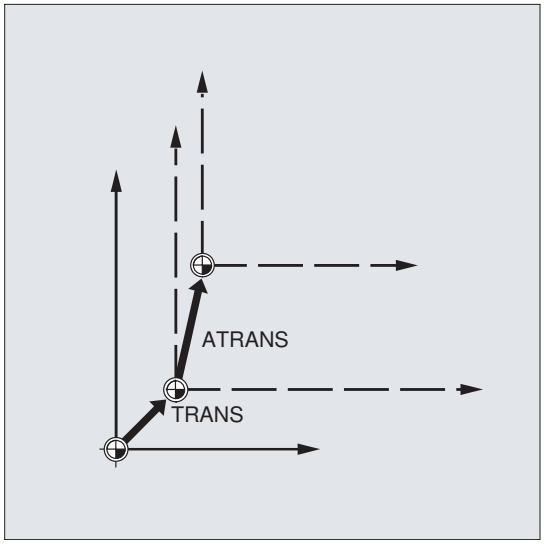



**Hinweis**

Eine Verschiebung, die auf bereits bestehenden Frames aufbauen soll, muss mit `ATRANS` programmiert werden.

### ATRANS X... Y... Z...

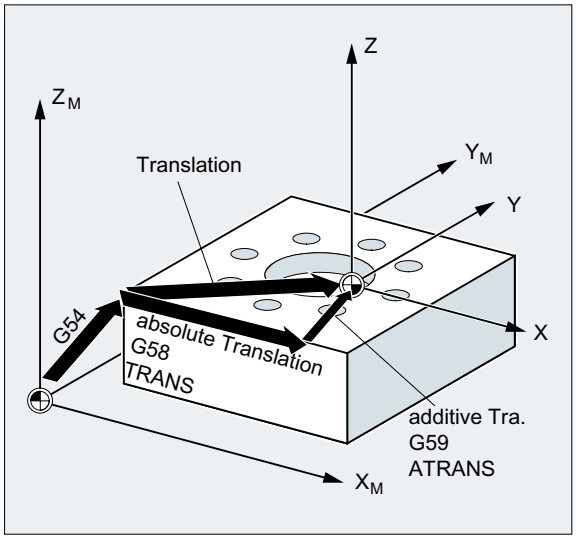
Nullpunktverschiebung um die in den jeweils angegebenen Achsrichtungen programmierten Verschiebewerte. Als Bezug gilt der aktuell eingestellte oder zuletzt programmierte Nullpunkt.



## 2.12.4 Programmierbare Nullpunktverschiebung (G58, G59)

Mit den Funktionen G58 und G59 können Translationsanteile der programmierbaren Nullpunktverschiebung (TRANS/ATRANS) (Seite 313) achsspezifisch ersetzt werden:

- G58: absoluter Translationsanteil (Grobverschiebung)
- G59: additiver Translationsanteil (Feinverschiebung)




### Voraussetzungen

Die Funktionen G58 und G59 sind nur einsetzbar, wenn die Feinverschiebung projektiert ist (MD24000 $MC_FRAME_ADD_COMPONENTS = 1).

### Syntax

```
G58 <Achse_1><Wert_1> ... <Achse_3><Wert_3>
G59 <Achse_1><Wert_1> ... <Achse_3><Wert_3>
```

### Bedeutung

| | |
|---|---|
| G58: | G58 ersetzt den absoluten Translationsanteil der programmierbaren Nullpunktverschiebung für die angegebene Achse, die additiv programmierte Verschiebung bleibt erhalten. Als Bezug gilt die zuletzt aufgerufene einstellbare Nullpunktverschiebung (G54 ... G57, G505 ... G599). |
| | Alleine im Satz: ja |
| G59: | G59 ersetzt den additiven Translationsanteil der programmierbaren Nullpunktverschiebung für die angegebene Achse, die absolut programmierte Verschiebung bleibt erhalten. |
| | Alleine im Satz: ja |
| <Achse_n>: | Geometrieachse im Kanal |
| <Wert_n>: | Verschiebewerte in Richtung der angegebenen Geometrieachse |

## Beispiel

```
Programmcode                Kommentar
...
N50 TRANS X10 Y10 Z10       ; Absoluter Translationsanteil X10 Y10 Z10
N60 ATRANS X5 Y5            ; Additiver Translationsanteil X5 Y5
                            → Gesamtverschiebung: X15 Y15 Z10
N70 G58 X20                 ; Absoluter Translationsanteil X20
                            → Gesamtverschiebung X25 Y15 Z10
N80 G59 X10 Y10             ; Additiver Translationsanteil X10 Y10
                            → Gesamtverschiebung X30 Y20 Z10
...
```

## Weitere Informationen

Der absolute Translationsanteil (**Grob-Verschiebung**) wird durch folgende Befehle modifiziert:

- `TRANS`
- `G58`
- `CTRANS`
- `CFINE`
- `$P_PFRAME[X,TR]`

Der additive Translationsanteil (**Fein-Verschiebung**) wird durch folgende Befehle modifiziert:

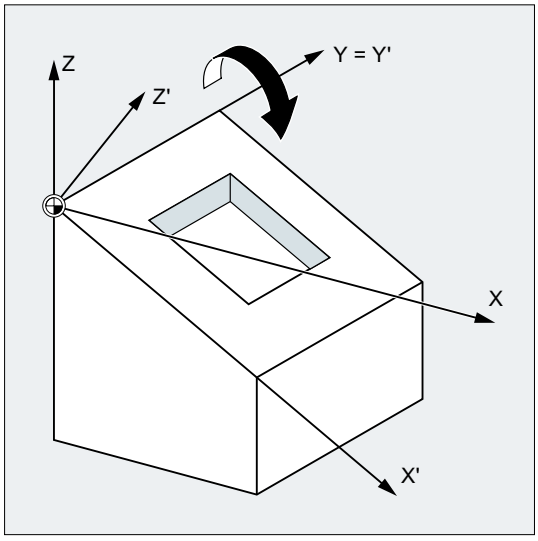
- `ATRANS`
- `G59`
- `CTRANS`
- `CFINE`
- `$P_PFRAME[X,FI]`

### Beispiele

| Befehl | Grob-Verschiebung $V_C$ | Fein-Verschiebung $V_F$ |
|---|---|---|
| `TRANS X10` | $V_C = 10$ | unverändert |
| `G58 X10` | $V_C = 10$ | unverändert |
| `$P_PFRAME[X,TR]=10` | $V_C = 10$ | unverändert |
| `ATRANS X10` | unverändert | $V_F = V_F + 10$ |
| `G59 X10` | unverändert | $V_F = 10$ |
| `$P_PFRAME[X,FI]=10` | unverändert | $V_F = 10$ |
| `CTRANS(X,10)` | $V_C = 10$ | $V_F = 0$ |
| `CTRANS()` | $V_C = 0$ | $V_F = 0$ |
| `CFINE(X,10)` | $V_C = 0$ | $V_F = 10$ |

## 2.12.5 Programmierbare Drehung (ROT, AROT, RPL)

Mit den Befehlen `ROT` / `AROT` kann das Werkstückkoordinatensystem im Raum gedreht werden. Die Befehle beziehen sich ausschließlich auf den Programmierbaren Frame `$P_PFRAME`.




### Syntax

```
ROT <1. GeoAx><Winkel> <2. GeoAx><Winkel> <3. GeoAx><Winkel>
ROT RPL=<Winkel>
AROT <1. GeoAx><Winkel> <2. GeoAx><Winkel> <3. GeoAx><Winkel>
AROT RPL=<Winkel>
```

**Hinweis**

Die Drehungen des Werkstückkoordinatensystems erfolgen über Euler-Winkel.

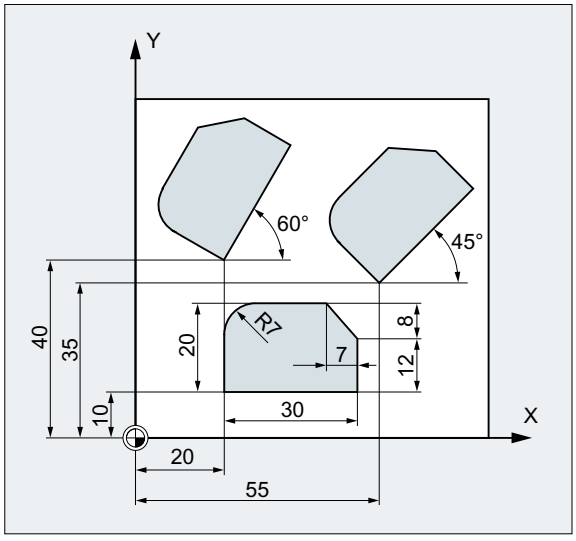
### Bedeutung

| | | |
|---|---|---|
| `ROT`: | Drehung absolut | |
| | Bezugsframe: | Programmierbarer Frame `$P_PFRAME` |
| | Bezugspunkt: | Nullpunkt des aktuellen mit `G54` ... `G57`, `G505` ... `G599` eingestellten Werkstückkoordinatensystems |
| `AROT`: | Drehung additiv | |
| | Bezugsframe: | Programmierbarer Frame `$P_PFRAME` |
| | Bezugspunkt: | Nullpunkt des aktuellen mit `G54` ... `G57`, `G505` ... `G599` eingestellten Werkstückkoordinatensystems |
| `<n-te GeoAx>`: | Bezeichner der n-ten Geometrieachse um die um den angegebenen Winkel gedreht werden soll.<br>Für nicht programmierte Geometrieachse wird als Drehwinkel implizit der Wert 0° gesetzt. | |

| | | |
|---|---|---|
| RPL: | Drehung um die zur aktiven Ebene (G17, G18, G19) senkrechte Geometrieachse um den angegebenen Winkel | |
| | Bezugsframe: | Programmierbarer Frame $P_PFRAME |
| | Bezugspunkt: | Nullpunkt des aktuellen mit G54 ... G57, G505 ... G599 eingestellten Werkstückkoordinatensystems |
| <Winkel> | Winkelangabe in Grad | |
| | Wertebereich: | -360° ≤ Winkel ≤ 360° |

## Beispiele

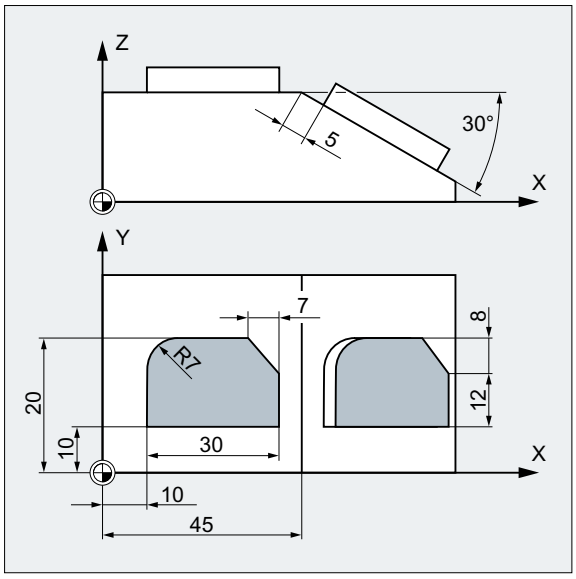
### Beispiel 1: Drehung in der G17-Ebene




Bei diesem Werkstück kommen die gezeigten Formen in einem Programm mehrfach vor. Zusätzlich zur Nullpunktverschiebung müssen Drehungen durchgeführt werden, da die Formen nicht achsparallel angeordnet sind.

| **Programmcode** | **Kommentar** |
|---|---|
| N10 G17 G54 | ; Arbeitsebene X/Y, Werkstücknullpunkt |
| N20 TRANS X20 Y10 | ; Absolute Verschiebung |
| N30 L10 | ; Unterprogramm-Aufruf |
| N40 TRANS X55 Y35 | ; Absolute Verschiebung |
| N50 AROT RPL=45 | ; Additive Drehung um die zur G17-Ebene |
| | ; senkrechte Z-Achse um 45° |
| N60 L10 | ; Unterprogramm-Aufruf |
| N70 TRANS X20 Y40 | ; Absolute Verschiebung |
| | ; (setzt alle bisherigen Verschiebungen zurück) |
| N80 AROT RPL=60 | ; Additive Drehung um die zur G17-Ebene |
| | ; senkrechte Z-Achse um 60° |
| N90 L10 | ; Unterprogramm-Aufruf |
| N100 G0 X100 Y100 | ; Wegfahren |
| N110 M30 | ; Programmende |

### Beispiel 2: Räumliche Drehung um die Y-Achse

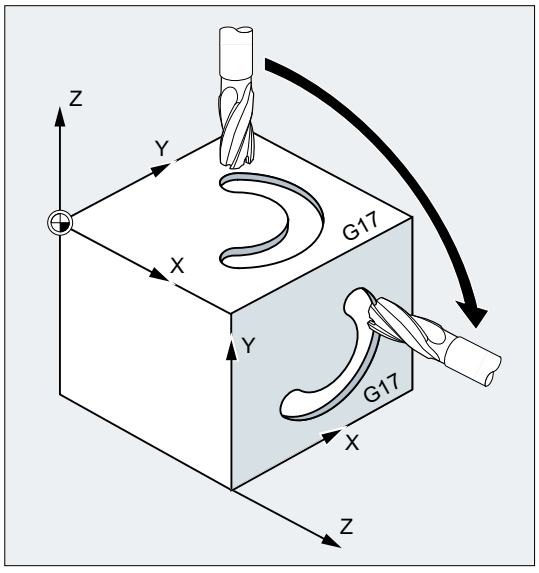



In diesem Beispiel sollen achsparallele und schräg liegende Werkstückflächen in einer Aufspannung bearbeitet werden.

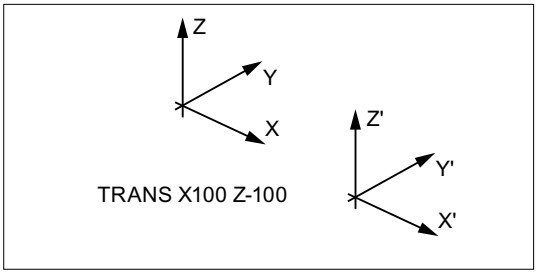
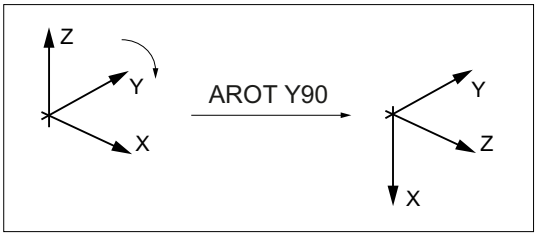
Voraussetzung:
Das Werkzeug muss zur schrägen Fläche senkrecht in der gedrehten Z-Richtung ausgerichtet werden.

| **Programmcode** | **Kommentar** |
|---|---|
| N10 G17 G54 | ; Arbeitsebene X/Y, Werkstücknullpunkt |
| N20 TRANS X10 Y10 | ; Absolute Verschiebung |
| N30 L10 | ; Unterprogramm-Aufruf |
| N40 ATRANS X35 | ; Additive Verschiebung |
| N50 AROT Y30 | ; Additive Drehung um die Y-Achse |
| N60 ATRANS X5 | ; Additive Verschiebung |
| N70 L10 | ; Unterprogramm-Aufruf |
| N80 G0 X300 Y100 M30 | ; Wegfahren, Programmende |

### Beispiel 3: Mehrseitenbearbeitung




In diesem Beispiel werden in zwei senkrecht zueinander stehenden Werkstückflächen identische Formen über Unterprogramme hergestellt. Im neuen Koordinatensystem auf der rechten Werkstückfläche sind Zustellrichtung, Arbeitsebene und der Nullpunkt so eingerichtet wie in der oberen Fläche. Damit gelten weiterhin die für den Unterprogrammablauf notwendigen Bedingungen: Arbeitsebene `G17`, Koordinatenebene X/Y, Zustellrichtung Z.

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 G17 G54` | `; Arbeitsebene X/Y, Werkstücknullpunkt` |
| `N20 L10` | `; Unterprogramm-Aufruf` |
| `N30 TRANS X100 Z-100` | `; Absolute Verschiebung des WKS` |
| |  |
| `N40 AROT Y90` | `; Additive Drehung des WKS um Y um 90°` |
| |  |

| Programmcode | Kommentar |
|---|---|
| N50 AROT Z90 | ; Additive Drehung des WKS um Z um 90° |
| | |
| N60 L10 | ; Unterprogramm-Aufruf |
| N70 G0 X300 Y100 M30 | ; Wegfahren, Programmende |



## Weitere Informationen

### Drehung in der aktiven Ebene

Bei Programmierung mittels `RPL=…` wird das WKS wird um die zur aktiven Ebene senkrechte Achse gedreht.

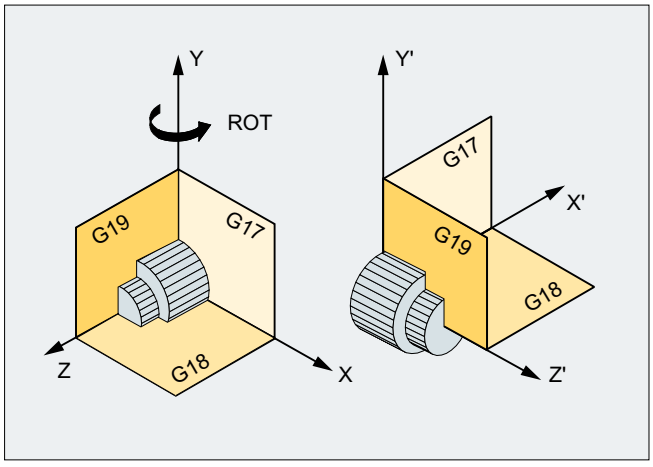



Bild 2-19 Drehung um die Y-Achse bzw. in der G18-Ebene

**⚠ WARNUNG**

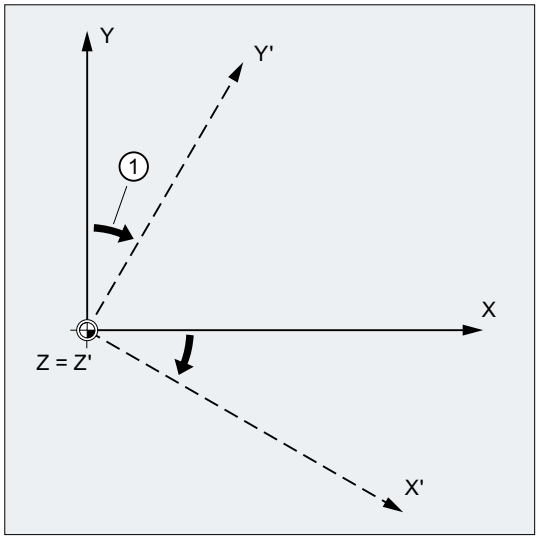
**Ebenenwechsel**

Wird nach einer Drehung ein Ebenenwechsel (`G17`, `G18`, `G19`) programmiert, bleiben die aktuellen Drehwinkel der jeweiligen Achsen erhalten und wirken auch in der neuen Ebene. Es wird daher dringend empfohlen, vor einem Ebenenwechsel die aktuellen Drehwinkel auf 0 zurückzusetzen:

- `N100 ROT X0 Y0 Z0 ; explizite Winkelprogrammierung`
- `N100 ROT ; implizite Winkelprogrammierung`

### Absolute Drehung mit ROT X... Y... Z...

Das WKS wird um die angegebenen Achsen auf die programmierten Drehwinkel gedreht.

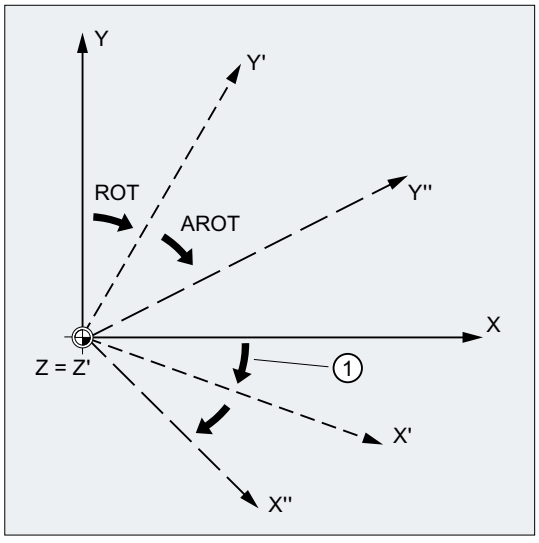



① Drehwinkel

Bild 2-20 Absolute Drehung um die Z-Achse

### Additive Drehung mit AROT X... Y... Z...

Das WKS wird um die angegebenen Achsen um die programmierten Drehwinkel weiter gedreht.




① Drehwinkel

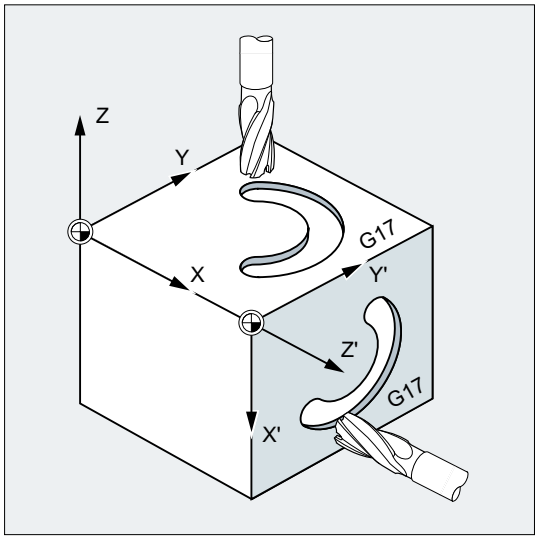
Bild 2-21 Absolute und additive Drehung um die Z-Achse

### Drehung der Arbeitsebene

Bei einer Drehung mittels `ROT` / `AROT` dreht sich die Arbeitsebene (`G17`, `G18`, `G19`) mit.

Beispiel: Arbeitsebene G17
Das WKS liegt auf der Deckfläche des Werkstücks. Durch Verschiebung und Drehung wird das Koordinatensystem in eine der Seitenflächen verschoben. Die Arbeitsebene G17 dreht sich mit. Hierdurch können Verfahrbewegungen in der G17-Ebene weiterhin über X und Y und Zustellungen über Z programmiert werden.

Voraussetzung:
Das Werkzeug muss senkrecht zur Arbeitsebene stehen und die positive Richtung der Zustellachse zeigt in Richtung Werkzeugaufnahme.
Durch Angabe von `CUT2DF` wirkt die Werkzeugradiuskorrektur in der gedrehten Ebene.




## 2.12.6 Programmierbare Framedrehungen mit Raumwinkeln (ROTS, AROTS, CROTS)

Mit den Befehlen `ROTS`, `AROTS` und `CROTS` können Drehungen des Werkstückkoordinatensystems in Raumwinkeln angegeben werden. Raumwinkel sind die Winkel, welche die Schnittgeraden der gewünschten, im Raum gedrehten Ebene, mit den Hauptebenen des noch nicht gedrehten WKS bilden.

**Hinweis**

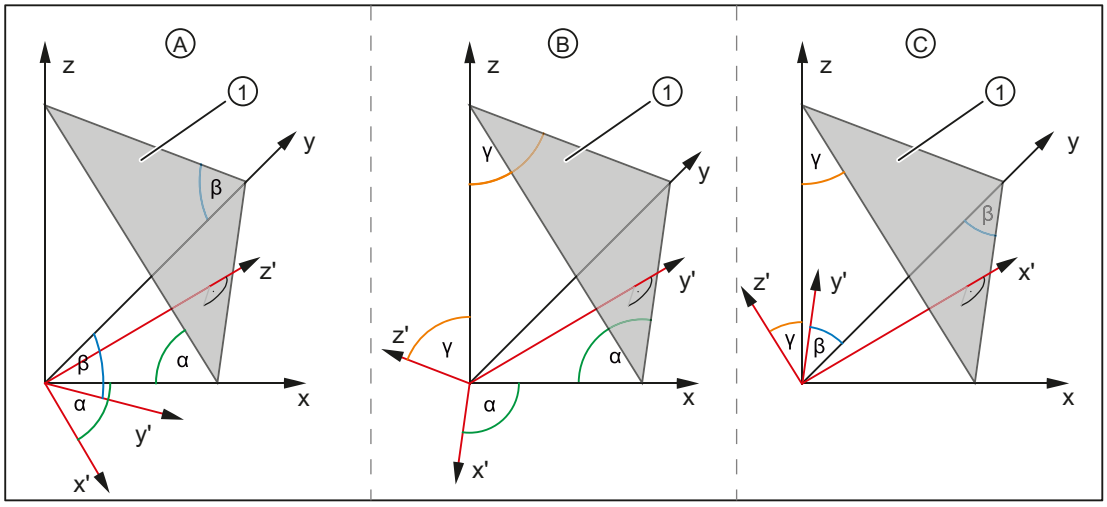
**Geometrieachsbezeichner**

Beispielhaft wird für die weitere Beschreibung folgende Festlegung getroffen:

- 1. Geometrieachse: X
- 2. Geometrieachse: Y
- 3. Geometrieachse: Z

Beispielhaft bewirkt, wie im folgenden Bild dargestellt, die Programmierung `ROTS X`α `Y`β ein Ausrichten der G17-Ebene des WKS parallel zur dargestellten schrägen Ebene. Die Lage des Nullpunkts des WKS bleibt dabei unverändert.

Die Orientierung des gedrehten WKS wird so festgelegt, dass die erste gedrehte Achse in der Ebene liegt, die durch diese und die 3. Achse des ursprünglichen Koordinatensystems aufgespannt wird. Im Beispiel: X' liegt in der ursprünglichen X/Z-Ebene.




① Schräge Ebene

α, β, γ Raumwinkel

A Ausrichten der G17-Ebene parallel zur schrägen Ebene:

- 1. Drehung
  Drehung von x um y um Winkel α ⇒
  x'-Achse parallel zur schrägen Ebene
- 2. Drehung
  Drehung von y' um x' um β ⇒
  y'-Achse parallel zur schrägen Ebene
  ⇒ z'-Achse senkrecht zur schrägen Ebene
  ⇒ G17 parallel zur schrägen Ebene

B Ausrichten der G18-Ebene parallel zur schrägen Ebene:

- 1. Drehung
  Drehung von z um x um den Winkel γ ⇒
  z'-Achse parallel zur schrägen Ebene
- 2. Drehung
  Drehung von x' um z' um Winkel α ⇒
  x'-Achse parallel zur schrägen Ebene
  ⇒ y'-Achse senkrecht zur schrägen Ebene
  ⇒ G18 parallel zur schrägen Ebene

C Ausrichten der G19-Ebene parallel zur schrägen Ebene:

- 1. Drehung
  Drehung von y um z um den Winkel β ⇒
  y'-Achse parallel zur schrägen Ebene
- 2. Drehung
  Drehung von z' um y' um Winkel γ ⇒
  z'-Achse parallel zur schrägen Ebene
  ⇒ x'-Achse senkrecht zur schrägen Ebene
  ⇒ G19 parallel zur schrägen Ebene

## Syntax

### Festlegungen

Die Lage einer Ebene im Raum ist durch zwei Raumwinkel eindeutig bestimmt. Durch Angabe eines dritten Raumwinkels wäre die Ebene überbestimmt. Sie ist daher nicht zulässig.

Bei Programmierung nur eines Raumwinkels erfolgt die Drehung des WKS identisch zu `ROT`, `AROT` (siehe Kapitel "Programmierbare Drehung (ROT, AROT, RPL) (Seite 319)").

Durch die beiden programmierten Achsen wird eine Ebene gemäß den Ebenendefinitionen für `G17`, `G18`, `G19` festgelegt. Dadurch ist die Reihenfolge der Koordinatenachsen (1. Achse / 2. Achse der Ebene) bzw. die Reihenfolge der Drehungen um die Raumwinkel festgelegt:

| Ebene | 1. Achse | 2. Achse |
|---|---|---|
| G17 | X | Y |
| G18 | Z | X |
| G19 | Y | Z |

### Ausrichtung der G17-Ebene ⇒ Raumwinkel für X und Y

- 1. Drehung: X um Y um den Winkel α
- 2. Drehung: Y um X' um den Winkel β
- Orientierung: X' liegt in der ursprünglichen Z/X-Ebene.

```
ROTS  X<α> Y<β>
AROTS X<α> Y<β>
CROTS X<α> Y<β>
```

### Ausrichtung der G18-Ebene ⇒ Raumwinkel für Z und X

- 1. Drehung: Z um X um den Winkel γ
- 2. Drehung: X um Z' um den Winkel α
- Orientierung: Z' liegt in der ursprünglichen Y/Z-Ebene

```
ROTS  Z<γ> X<α>
AROTS Z<γ> X<α>
CROTS Z<γ> X<α>
```

### Ausrichtung der G19-Ebene ⇒ Raumwinkel für Y und Z

- 1. Drehung: Y um Z um den Winkel β
- 2. Drehung: Z um Y' um den Winkel γ
- Orientierung: Y' liegt in der ursprünglichen X/Y-Ebene.

```
ROTS  Y<β> Z<γ>
AROTS Y<β> Z<γ>
CROTS Y<β> Z<γ>
```

### Bedeutung

| | |
|---|---|
| ROTS: | Frame-Drehungen mit Raumwinkeln absolut,<br>Bezugsframe: Programmierbarer Frame $P_PFRAME |
| AROTS: | Frame-Drehungen mit Raumwinkeln additiv;<br>Bezugsframe: Programmierbarer Frame $P_PFRAME |
| CROTS: | Frame-Drehungen mit Raumwinkeln absolut,<br>Bezugsframe: programmierter Frame $P_... |
| X, Y, Z: | Geometrieachsbezeichner (siehe oben Hinweis: Geometrieachsbezeichner) |
| α, β, γ: | Raumwinkel bezogen auf die entsprechende Geometrieachse:<br>• α → X<br>• β → Y<br>• γ → Z |

## 2.12.7 Programmierbarer Maßstabsfaktor (SCALE, ASCALE)

Mit SCALE/ASCALE können für alle Bahn-, Synchron- und Positionierachsen Maßstabsfaktoren zum Vergrößern oder Verkleinern in Richtung der jeweils angegebenen Achsen programmiert werden. Dadurch ist es möglich, geometrisch ähnliche Formen oder unterschiedliche Schwundmaße bei der Programmierung zu berücksichtigen.

### Syntax

```
SCALE X… Y… Z…
ASCALE X… Y… Z…
```

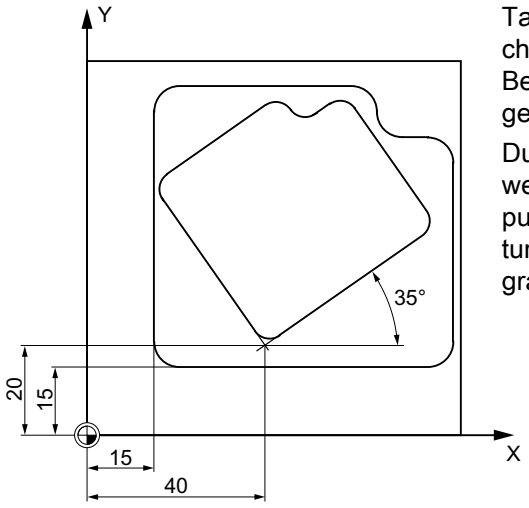
**Hinweis**

Frame-Anweisungen werden jeweils in einem eigenen NC-Satz programmiert.

### Bedeutung

| | |
|---|---|
| SCALE: | Vergrößern / Verkleinern absolut, bezogen auf das aktuell gültige, mit G54 ... G57, G505 ... G599 eingestellte Koordinatensystem |
| ASCALE: | Vergrößern/Verkleinern additiv, bezogen auf das aktuell gültige eingestellte oder programmierte Koordinatensystem |
| X… Y… Z… : | Maßstabsfaktoren in Richtung der angegebenen Geometrieachsen |

## Beispiel




Bei diesem Werkstück kommen die beiden Taschen zweimal vor, jedoch in unterschiedlichen Größen und zueinander verdreht. Die Bearbeitungsfolge ist im Unterprogramm abgelegt.

Durch Nullpunktverschiebung und Rotation werden die jeweils benötigten Werkstücknullpunkte gesetzt, durch Skalierung wird die Kontur verkleinert und dann wieder das Unterprogramm aufgerufen.

| **Programmcode** | **Kommentar** |
|---|---|
| N10 G17 G54 | ; Arbeitsebene X/Y, Werkstücknullpunkt |
| N20 TRANS X15 Y15 | ; Absolute Verschiebung |
| N30 L10 | ; Große Tasche fertigen |
| N40 TRANS X40 Y20 | ; Absolute Verschiebung |
| N50 AROT RPL=35 | ; Drehung in der Ebene um 35° |
| N60 ASCALE X0.7 Y0.7 | ; Maßstabsfaktor für die kleine Tasche |
| N70 L10 | ; Kleine Tasche fertigen |
| N80G0 X300 Y100 M30 | ; Wegfahren, Programmende |

## Weitere Informationen

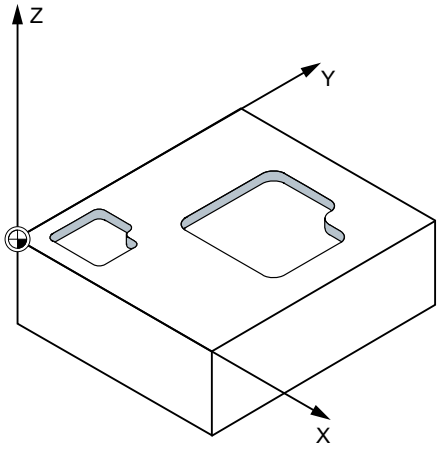
### SCALE X... Y... Z...

Für jede Achse kann ein eigener Maßstabsfaktor angegeben werden, um den vergrößert oder verkleinert werden soll. Die Skalierung bezieht sich auf das mit G54 ... G57, G505 ... G599 eingestellte Werkstückkoordinatensystem.

**ACHTUNG**

**Kein Ursprungs-Frame**

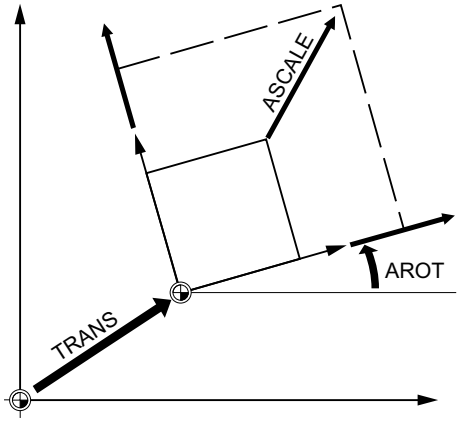
Der Befehl `SCALE` setzt alle Frame-Komponenten des vorher gesetzten programmierbaren Frames zurück.




### ASCALE X... Y... Z...

Eine Maßstabsveränderung, die auf bereits bestehenden Frames aufbauen soll, wird mit `ASCALE` programmiert. In diesem Fall wird der zuletzt gültige mit dem neuen Maßstabsfaktor multipliziert.

Als Bezug für die Maßstabsveränderung gilt das aktuell eingestellte oder zuletzt programmierte Koordinatensystem.



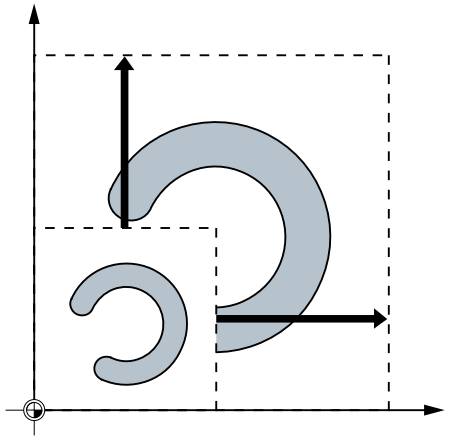

### Skalierung und Verschiebung

**Hinweis**

Wenn nach `SCALE` eine Verschiebung mit `ATRANS` programmiert wird, dann werden die Verschiebewerte ebenfalls skaliert.

### Unterschiedliche Maßstabsfaktoren

| ACHTUNG |
| --- |
| **Kollisionsgefahr** |
| Vorsicht mit unterschiedlichen Maßstabsfaktoren! Kreisinterpolationen können z. B. nur mit den gleichen Faktoren skaliert werden. |



**Hinweis**

Für die Programmierung verzerrter Kreise können unterschiedliche Maßstabsfaktoren jedoch gezielt eingesetzt werden.

## 2.12.8 Programmierbare Spiegelung (MIRROR, AMIRROR)

Mit `MIRROR`/`AMIRROR` können Werkstückformen an Koordinatenachsen gespiegelt werden. Alle Fahrbewegungen, die danach z. B. im Unterprogramm programmiert sind, werden gespiegelt ausgeführt.

### Syntax

```
MIRROR X... Y... Z...
AMIRROR X... Y... Z...
```

**Hinweis**

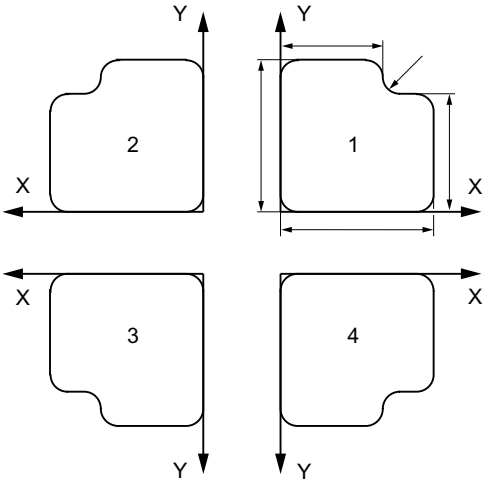
Frame-Anweisungen werden jeweils in einem eigenen NC-Satz programmiert.

## Bedeutung

| | |
|---|---|
| MIRROR: | Spiegeln absolut, bezogen auf das aktuell gültige, mit G54 ... G57, G505 ... G599 eingestellte Koordinatensystem |
| AMIRROR: | Spiegeln additiv, bezogen auf das aktuell gültige eingestellte oder programmierte Koordinatensystem |
| X... Y... Z...: | Geometrieachse, deren Richtung getauscht werden soll. Der hier angegebene Wert ist frei wählbar, z. B. X0 Y0 Z0. |

## Beispiele

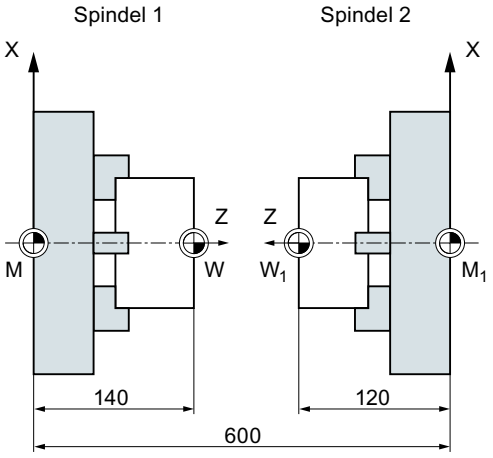
### Beispiel 1: Fräsen




Die hier gezeigte Kontur wird einmal als Unterprogramm programmiert. Die drei weiteren Konturen werden durch Spiegelung erzeugt. Der Werkstücknullpunkt wird zentral zu den Konturen angeordnet.

| **Programmcode** | **Kommentar** |
|---|---|
| N10 G17 G54 | ; Arbeitsebene X/Y, Werkstücknullpunkt |
| N20 L10 | ; Erste Kontur rechts oben fertigen |
| N30 MIRROR X0 | ; Spiegeln der X-Achse (in X wird die Richtung getauscht) |
| N40 L10 | ; Zweite Kontur links oben fertigen |
| N50 AMIRROR Y0 | ; Spiegeln der Y-Achse (in Y wird die Richtung getauscht) |
| N60 L10 | ; Dritte Kontur links unten fertigen |
| N70 MIRROR Y0 | ; MIRROR setzt vorherige Frames zurück. Spiegeln der Y-Achse (in Y wird die Richtung getauscht) |
| N80 L10 | ; Vierte Kontur rechts unten fertigen |
| N90 MIRROR | ; Spiegeln ausschalten |
| N100 G0 X300 Y100 M30 | ; Wegfahren, Programmende |

### Beispiel 2: Drehen




Die eigentliche Bearbeitung wird als Unterprogramm abgelegt, die Abarbeitung an der jeweiligen Spindel durch Spiegelungen und Verschiebungen realisiert.

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 TRANS X0 Z140` | `; Nullpunktverschiebung auf W` |
| `...` | `; Bearbeitung der 1. Seite mit Spindel 1` |
| `N30 TRANS X0 Z600` | `; Nullpunktverschiebung auf Spindel 2` |
| `N40 AMIRROR Z0` | `; Spiegeln der Z-Achse` |
| `N50 ATRANS Z120` | `; Nullpunktverschiebung auf W1` |
| `...` | `; Bearbeitung der 2. Seite mit Spindel 2` |

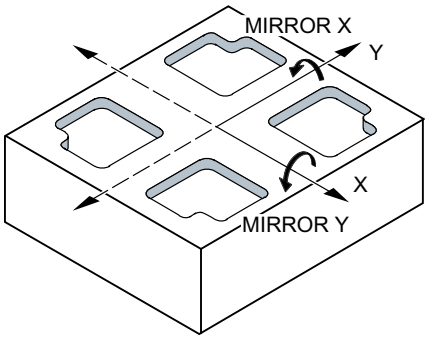
## Weitere Informationen

### MIRROR X... Y... Z...

Die Spiegelung wird über axiale Richtungswechsel in der gewählten Arbeitsebene programmiert.

Beispiel: Arbeitsebene G17 X/Y

Die Spiegelung (an der Y-Achse) erfordert einen Richtungswechsel in X und wird demnach programmiert mit `MIRROR X0`. Die Kontur wird dann spiegelverkehrt auf der gegenüberliegenden Seite der Spiegelachse Y bearbeitet.

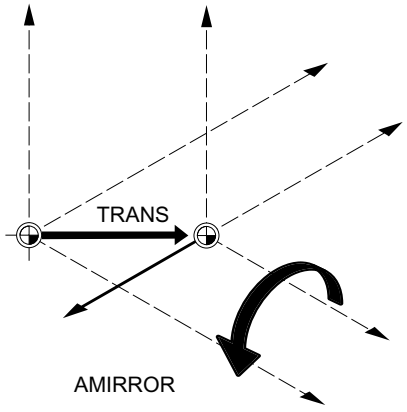



Die Spiegelung bezieht sich auf das aktuell gültige, mit G54 ... G57, G505 ... G599 eingestellte Koordinatensystem.

| ACHTUNG |
| --- |
| **Kein Ursprungs-Frame**<br>Der Befehl `MIRROR` setzt alle Frame-Komponenten des vorher gesetzten programmierbaren Frames zurück. |

### AMIRROR X... Y... Z...

Eine Spiegelung, die auf bereits bestehenden Transformationen aufbauen soll, wird mit `AMIRROR` programmiert. Als Bezug gilt das aktuell eingestellte oder zuletzt programmierte Koordinatensystem.




### Spiegelung ausschalten

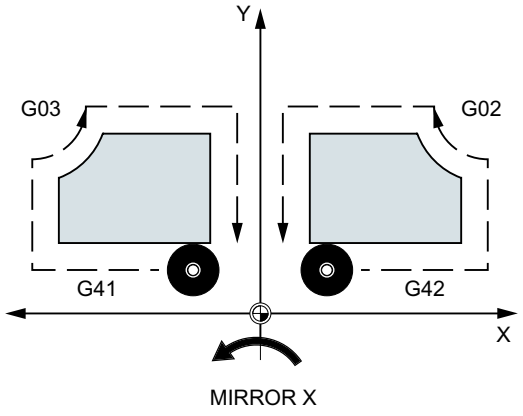
Für alle Achsen: `MIRROR` (ohne Achsangabe)

Hierbei werden alle Frame-Komponenten des vorher programmierten Frames zurückgesetzt.

### Werkzeugradiuskorrektur

**Hinweis**

Die Steuerung stellt mit dem Spiegelbefehl automatisch die Bahnkorrekturbefehle (`G41`/`G42` bzw. `G42`/`G41`) entsprechend der veränderten Bearbeitungsrichtung um.




Gleiches gilt für den Kreisdrehsinn (G2/G3 bzw. G3/G2).

**Hinweis**

Wenn nach `MIRROR` eine additive Drehung mit `AROT` programmiert wird, muss fallweise mit umgekehrten Drehrichtungen (positiv/negativ bzw. negativ/positiv) gearbeitet werden. Spiegelungen in den Geometrieachsen werden von der Steuerung selbsttätig in Rotationen und ggf. Spiegelungen der durch Maschinendatum einstellbaren Spiegelachse umgerechnet. Dies gilt auch für einstellbare Nullpunktverschiebungen.

### Spiegelachse

Über Maschinendatum kann eingestellt werden, um welche Achse gespiegelt wird:

MD10610 $MN_MIRROR_REF_AX = <Wert>

| Wert | Bedeutung |
|---|---|
| 0 | Es wird um die programmierte Achse gespiegelt (Negieren der Werte). |
| 1 | X-Achse ist Bezugsachse. |
| 2 | Y-Achse ist Bezugsachse. |
| 3 | Z-Achse ist Bezugsachse. |

### Interpretation der programmierten Werte

Über Maschinendatum kann eingestellt werden, wie die programmierten Werte zu interpretieren sind:

MD10612 $MN_MIRROR_TOGGLE = <Wert>

| Wert | Bedeutung |
|---|---|
| 0 | Programmierte Achswerte werden nicht ausgewertet. |
| 1 | Programmierte Achswerte werden ausgewertet:<br>• Bei programmierten Achswerten ≠ 0 wird die Achse gespiegelt, wenn sie noch nicht gespiegelt ist.<br>• Bei einem programmierten Achswert = 0 wird eine Spiegelung ausgeschaltet. |

## 2.12.9 Frame-Erzeugung nach Werkzeugausrichtung (TOFRAME, TOROT, PAROT)

TOFRAME erzeugt ein rechtwinkliges Koordinatensystem, dessen Z-Achse mit der aktuellen Werkzeugausrichtung übereinstimmt. Dadurch hat der Anwender die Möglichkeit, das Werkzeug in Z-Richtung kollisionsfrei freizufahren (z. B. nach einem Werkzeugbruch bei einem 5-Achs-Programm).

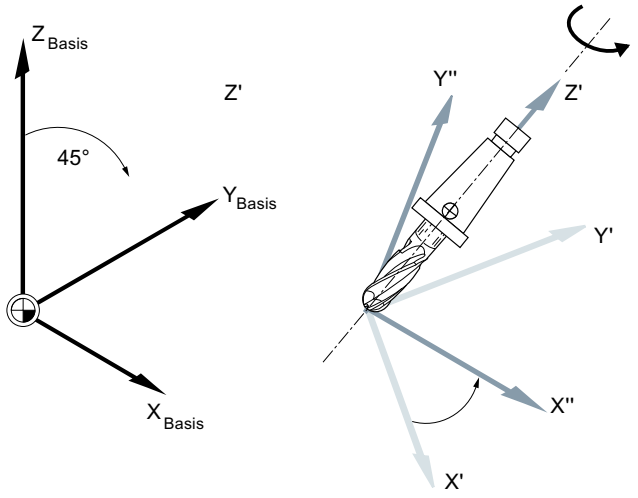
Die Lage der beiden Achsen X und Y ist dabei abhängig von der Einstellung im Maschinendatum MD21110 $MC_X_AXES_IN_OLD_X_Z_PLANE (Koordinatensystem bei automatischer Frame-Definition). Das neue Koordinatensystem wird entweder so belassen, wie es sich aus der Kinematik der Maschine ergibt, oder es wird zusätzlich so um die neue Z-Achse gedreht, dass die neue X-Achse in der alten Z-X-Ebene liegt (siehe Angaben des Maschinenherstellers).

Der resultierende Frame, der die Orientierung beschreibt, steht in der Systemvariablen für den programmierbaren Frame ($P_PFRAME).

Mit TOROT wird im programmierten Frame nur der Rotationsanteil überschrieben. Alle übrigen Komponenten bleiben unverändert.

TOFRAME und TOROT sind auf Fräsbearbeitungen zugeschnitten, bei denen typischerweise G17 (Arbeitsebene X/Y) aktiv ist. Bei Drehbearbeitungen oder allgemein bei aktiven G18 oder G19 werden dagegen Frames benötigt, bei denen die X- oder Y-Achse mit der Ausrichtung des Werkzeugs übereinstimmt. Diese Frames werden mit den Befehlen TOFRAMEX/TOROTX oder TOFRAMEY/TOROTY programmiert.

Mit PAROT wird das Werkstückkoordinatensystem (WKS) am Werkstück ausgerichtet.




## Syntax

```
TOFRAME/TOFRAMEZ/TOFRAMEY/TOFRAMEX
...
TOROTOF

TOROT/TOROTZ/TOROTY/TOROTX
...
TOROTOF

PAROT
...
PAROTOF
```

## Bedeutung

| | |
|---|---|
| TOFRAME: | Z-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten |
| TOFRAMEZ: | wie TOFRAME |
| TOFRAMEY: | Y-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten |
| TOFRAMEX: | X-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten |
| TOROT: | Z-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten<br>Die durch TOROT definierte Drehung ist die gleiche wie bei TOFRAME. |
| TOROTZ: | wie TOROT |

| | |
|---|---|
| TOROTY: | Y-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten |
| TOROTX: | X-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten |
| TOROTOF: | Ausrichtung parallel zur Werkzeugorientierung ausschalten |
| PAROT: | WKS durch Frame-Drehung am Werkstück ausrichten<br>Translationen, Skalierungen und Spiegelungen im aktiven Frame bleiben erhalten. |
| PAROTOF: | Die mit PAROT aktivierte werkstückbezogene Frame-Drehung wird mit PAROTOF ausgeschaltet. |

### Hinweis

Mit dem Befehl TOROT wird eine konsistente Programmierung bei aktiven orientierbaren Werkzeugträgern für jeden Kinematiktyp erreicht.

Analog zur Situation bei drehbarem Werkzeugträger kann mit PAROT eine Drehung des Werkzeugtisches aktiviert werden. Damit wird ein Frame definiert, welches die Lage des Werkstückkoordinatensystems so verändert, dass es zu keiner Ausgleichsbewegung der Maschine kommt. Der Sprachbefehl PAROT wird nicht abgelehnt, wenn kein orientierbarer Werkzeugträger aktiv ist.

## Beispiel

| Programmcode | Kommentar |
|---|---|
| N100 G0 G53 X100 Z100 D0 | |
| N120 TOFRAME | |
| N140 G91 Z20 | ; TOFRAME wird eingerechnet, alle programmierten Geometrieachsbewegungen beziehen sich auf das neue Koordinatensystem. |
| N160 X50 | |
| ... | |

## Weitere Informationen

### Zuordnung Achsrichtung

Wird an Stelle von TOFRAME / TOFRAMEZ oder TOROT / TOROTZ einer der Befehle TOFRAMEX, TOFRAMEY, TOROTX, TOROTY programmiert, dann gelten die Zuordnungen der Achsrichtungen entsprechend dieser Tabelle:

| Befehl | Werkzeugrichtung (Applikate) | Nebenachse (Abszisse) | Nebenachse (Ordinate) |
|---|---|---|---|
| TOFRAME / TOFRAMEZ / TOROT / TOROTZ | Z | X | Y |
| TOFRAMEY / TOROTY | Y | Z | X |
| TOFRAMEX / TOROTX | X | Y | Z |

#### Eigener Systemframe für TOFRAME oder TOROT

Die durch TOFRAME oder TOROT entstehenden Frames können in einen eigenen Systemframe $P_TOOLFRAME geschrieben werden. Dazu muss das Bit 3 im Maschinendatum MD28082 $MC_MM_SYSTEM_FRAME_MASK gesetzt werden. Der programmierbare Frame bleibt hierbei unverändert erhalten. Unterschiede ergeben sich, wenn der programmierbare Frame weiter bearbeitet wird.

## 2.12.10 Frame abwählen (G53, G153, SUPA, G500)

Beim Abarbeiten bestimmter Vorgänge, wie z. B. dem Anfahren des Werkzeugwechselpunkts, müssen verschiedene Frame-Komponenten definiert und zeitlich bestimmt unterdrückt werden.

Einstellbare Frames können entweder modal ausgeschaltet oder satzweise unterdrückt werden.

Programmierbare Frames können satzweise unterdrückt oder gelöscht werden.

### Syntax

```
G53
G153
SUPA
G500
TRANS
ROT
SCALE
MIRROR
```

### Bedeutung

| | |
|---|---|
| G53: | Satzweise wirksames Unterdrücken aller programmierbaren und einstellbaren Frames |
| G153: | G153 wirkt wie G53 und unterdrückt darüber hinaus den Gesamt-Basisframe ($P_ACTBFRAME) |
| SUPA: | SUPA wirkt wie G153 und unterdrückt darüber hinaus:<br>• Handradverschiebungen (DRF)<br>• überlagerte Bewegungen<br>• externe Nullpunktverschiebung<br>• PRESET-Verschiebung |
| G500: | Modal wirksames Ausschalten aller einstellbaren Frames (G54 ... G57, G505 ... G599), wenn in G500 kein Wert steht. |
| TRANS ROT SCALE MIRROR: | ohne Achsangabe bewirkt ein Löschen der programmierbaren Frames. |

## 2.12.11 Programmierung: Überlagerungen achsspezifisch abwählen (CORROF)

Mit der Prozedur CORROF werden folgende achsspezifischen Überlagerungen gelöscht:

- Über Handradverfahren eingestellte additive Nullpunktverschiebunge (DRF-Verschiebungen)
- Über die Systemvariable $AA_OFF programmierte Positionsoffsets

Durch das Löschen eines Überlagerungswertes wird ein Vorlaufstopp ausgelöst und der Positionsanteil der abgewählten überlagerten Bewegung in die Position im Basiskoordinatensystem übernommen. Dabei wird keine Achse verfahren.

Der über die Systemvariable $AA_IM (Aktueller **MKS**-Sollwert der Achse) lesbare Positionswert im **Maschinen**koordinatensysstem **ändert sich nicht**.

Der über die Systemvariable $AA_IW (Aktueller **WKS**-Sollwert der Achse) lesbare Positionswert im **Werkstück**koordinatensysstem **ändert sich**, da er nun den abgewählten Anteil der überlagerten Bewegung enthält.

**Hinweis**

In einem **NC-Programm** darf `CORROF` programmiert werden.

In einer **Synchronaktion** darf `CORROF` **nicht** programmiert werden.

### Syntax

```
CORROF(<Axis>,"<String>"[,<Axis>,"<String>"])
```

### Bedeutung

| | | |
|---|---|---|
| `CORROF`: | Prozedur zur Abwahl folgender Verschiebungen bzw. Überlagerungen einer Achse:<br>• DRF-Verschiebung<br>• Positionsoffsets ($AA_OFF) | |
| | Wirksamkeit: | modal |
| `<Axis>`: | Achsbezeichner (Kanal-, Geometrie- oder Maschinenachsbezeichner) | |
| | Datentyp: | AXIS |
| `<String>`: | Zeichenkette zur Definition der Überlagerungsart | |
| | Datentyp: | BOOL |
| | **Wert** | **Bedeutung** |
| | `DRF` | DRF-Verschiebung |
| | `AA_OFF` | Positionsoffset ($AA_OFF) |

## Beispiele

### Beispiel 1: Achsspezifische Abwahl einer DRF-Verschiebung (1)

Über DRF-Handradverfahren wird eine DRF-Verschiebung in der X-Achse erzeugt. Für alle anderen Achsen des Kanals sind keine DRF-Verschiebungen wirksam.

```
Programmcode                Kommentar
N10 CORROF(X,"DRF")         ; CORROF wirkt hier wie DRFOF.
...
```

### Beispiel 2: Achsspezifische Abwahl einer DRF-Verschiebung (2)

Über DRF-Handradverfahren wird eine DRF-Verschiebung in der X-und in der Y-Achse erzeugt. Für alle anderen Achsen des Kanals sind keine DRF-Verschiebungen wirksam.

```
Programmcode                Kommentar
; Nur die DRF-Verschiebung der X-Achse wird abgewählt, die DRF-Verschiebung der Y-
Achse bleibt erhalten.
; Bei DRFOF wären beide Verschiebungen abgewählt worden.
N10 CORROF(X,"DRF")
...
```

### Beispiel 3: Achsspezifische Abwahl eines $AA_OFF-Positionsoffsets

```
Programmcode                                          Kommentar
; Für die X-Achse wird ein Positionsoffset == 10 interpoliert.
N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5


...
; Der Positionsoffset der X-Achse wird abgewählt: $AA_OFF[X]=0
; Die X-Achse wird nicht verfahren.
; Zur aktuellen Position der X-Achse wird der Positionsoffset hinzugerechnet.
N80 CORROF(X,"AA_OFF")
...
```

### Beispiel 4: Achsspezifische Abwahl einer DRF-Verschiebung und eines $AA_OFF-Positionsoffsets (1)

Über DRF-Handradverfahren wird eine DRF-Verschiebung in der X-Achse erzeugt. Für alle anderen Achsen des Kanals sind keine DRF-Verschiebungen wirksam.

```
Programmcode                                          Kommentar
; Für die X-Achse wird ein Positionsoffset von 10 interpoliert.
N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5
...
; Nur die DRF-Verschiebung und der Positionsoffset der X-Achse wird abgewählt.
; Die DRF-Verschiebung der Y-Achse bleibt erhalten.
N70 CORROF(X,"DRF",X,"AA_OFF")
...
```

#### Beispiel 5: Achsspezifische Abwahl einer DRF-Verschiebung und eines $AA_OFF-Positionsoffsets (2)

Über DRF-Handradverfahren wird eine DRF-Verschiebung in der X-Achse und in der Y-Achse erzeugt. Für alle anderen Achsen des Kanals sind keine DRF-Verschiebungen wirksam.

| Programmcode | Kommentar |
|---|---|
| `; Für die X-Achse wird ein Positionsoffset == 10 interpoliert.` | |
| `N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5` | |
| `...` | |
| `; Die DRF-Verschiebung der Y-Achse und der Positionsoffset der X-Achse werden abgewählt.` | |
| `; Die DRF-Verschiebung der X-Achse bleibt erhalten.` | |
| `N70 CORROF(Y,"DRF",X,"AA_OFF")` | |
| `...` | |

## Weitere Informationen

#### $AA_OFF_VAL

Nach der Abwahl des Positionsoffsets aufgrund von $AA_OFF ist die Systemvariable $AA_OFF_VAL (Integrierter Weg der Achsüberlagerung) der entsprechenden Achse gleich Null.

#### $AA_OFF in der Betriebsart JOG

Auch in der Betriebsart JOG findet bei einer Änderung von $AA_OFF eine Interpolation des Positionsoffsets als überlagerte Bewegung statt, wenn die Freischaltung dieser Funktion über das Maschinendatum MD36750 $MA_AA_OFF_MODE erfolgt ist.

#### $AA_OFF in Synchronaktion

Ist bei der Abwahl des Positionsoffsets über `CORROF(<Achse>,"AA_OFF")` eine Synchronaktion aktiv, die $AA_OFF sofort wieder setzt (`DO $AA_OFF[<Achse>]=<Wert>`), dann wird $AA_OFF abgewählt und nicht wieder gesetzt und der Alarm 21660 angezeigt. Wird die Synchronaktion jedoch später aktiv, z. B. im Satz nach `CORROF`, dann wird $AA_OFF gesetzt und ein Positionsoffset interpoliert.

#### Automatischer Kanalachstausch

Falls eine Achse, für die ein `CORROF` programmiert wurde, in einem anderen Kanal aktiv ist, dann wird sie mit Achstausch in den Kanal geholt (Voraussetzung:
MD30552 $MA_AUTO_GET_TYPE > 0) und dann der Positionsoffset und/oder die DRF-Verschiebung abgewählt.

## 2.12.12 Additive Nullpunktverschiebungen abwählen (DRFOF)

Die über Handradverfahren eingestellten additiven Nullpunktverschiebungen (DRF-Verschiebungen) werden über die Prozedur `DRFOF` abgewählt.

Durch die Abwahl wird ein Vorlaufstopp ausgelöst und der Positionsanteil der abgewählten DRF-Verschiebung in die Position im Basiskoordinatensystem übernommen, dabei wird keine

Achse verfahren. Der Wert der Systemvariablen $AA_IM (Aktueller MKS-Sollwert einer Achse) ändert sich nicht, der Wert der Systemvariablen $AA_IW (Aktueller WKS-Sollwert einer Achse) verändert sich, da er nun den abgewählten Anteil aus der überlagerten Bewegung enthält.

## Syntax

```
DRFOF
```

## Bedeutung

| | | |
|---|---|---|
| DRFOF: | Prozedur zur Abwahl der DRF-Verschiebungen für alle aktiven Achsen des Kanals | |
| | Wirksamkeit: | modal |

## Beispiele

### Beispiel 1: Achsspezifische Abwahl einer DRF-Verschiebung (1)

Über DRF-Handradverfahren wird eine DRF-Verschiebung in der X-Achse erzeugt. Für alle anderen Achsen des Kanals sind keine DRF-Verschiebungen wirksam.

| Programmcode | Kommentar |
|---|---|
| N10 CORROF(X,"DRF") | ; CORROF wirkt hier wie DRFOF. |
| ... | |

### Beispiel 2: Achsspezifische Abwahl einer DRF-Verschiebung (2)

Über DRF-Handradverfahren wird eine DRF-Verschiebung in der X-und in der Y-Achse erzeugt. Für alle anderen Achsen des Kanals sind keine DRF-Verschiebungen wirksam.

| Programmcode | Kommentar |
|---|---|
| N10 CORROF(X,"DRF") | ; Nur die DRF-Verschiebung der X-Achse wird abgewählt, die DRF-Verschiebung der Y-Achse bleibt erhalten (bei DRFOF wären beide Verschiebungen abgewählt worden). |
| ... | |

### Beispiel 3: Achsspezifische Abwahl einer DRF-Verschiebung und eines $AA_OFF-Positionsoffsets (1)

Über DRF-Handradverfahren wird eine DRF-Verschiebung in der X-Achse erzeugt. Für alle anderen Achsen des Kanals sind keine DRF-Verschiebungen wirksam.

| Programmcode | Kommentar |
|---|---|
| N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5 | ; Für die X-Achse wird ein Positionsoffset == 10 interpoliert. |
| ... | |
| N70 CORROF(X,"DRF",X,"AA_OFF") | ; Nur die DRF-Verschiebung und der Positionsoffset der X-Achse wird abgewählt, die DRF-Verschiebung der Y-Achse bleibt erhalten. |
| ... | |

**Beispiel 4: Achsspezifische Abwahl einer DRF-Verschiebung und eines $AA_OFF-Positionsoffsets (2)**

Über DRF-Handradverfahren wird eine DRF-Verschiebung in der X-Achse und in der Y-Achse erzeugt. Für alle anderen Achsen des Kanals sind keine DRF-Verschiebungen wirksam.

| Programmcode | Kommentar |
|---|---|
| N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5 | ; Für die X-Achse wird ein Positionsoffset == 10 interpoliert. |
| ... | |
| N70 CORROF(Y,"DRF",X,"AA_OFF") | ; Die DRF-Verschiebung der Y-Achse und der Positionsoffset der X-Achse werden abgewählt, die DRF-Verschiebung der X-Achse bleibt erhalten. |
| ... | |

## 2.12.13 Schleifspezifische Nullpunktverschiebungen (GFRAME0, GFRAME1 ... GFRAME100)

### Befehl zur Aktivierung eines Schleifframes im Kanal

Durch Programmierung des Befehles GFRAME<n> wird der entsprechende Schleifframe der Datenhaltung $P_GFR[<n>] im Kanal aktiv. Dazu wird der aktive Schleifframe $P_GFRAME gleich dem Schleifframe der Datenhaltung $P_GFR[<n>] gesetzt:

GFRAME<n> ⇒ $P_GFRAME = $P_GFR[<n>]

| Befehl | Im Kanal aktivierter Schleifframe |
|---|---|
| GFRAME0 | $P_GFR[ 0 ] (**Nullframe**) |
| GFRAME1 | $P_GFR[ 1 ] |
| ... | ... |
| GFRAME100 | $P_GFR[ 100 ] |

### Syntax

GFRAME<n>

### Bedeutung

| | | |
|---|---|---|
| GFRAME<n>: | Aktivierung des Schleifframes <n> der Datenhaltung | |
| | G-Gruppe: | 64 |
| | Grundstellung: | MD20150 $MC_GCODE_RESET_VALUES[ **63** ] |
| | Wirksamkeit: | Modal |
| <n>: | Nummer des Schleifframes | |
| | Wertebereich: | 0, 1, 2, ... 100 |

# 2.13 Hilfsfunktionsausgaben

## Funktion

Mit der Hilfsfunktionsausgabe wird der PLC zeitgerecht mitgeteilt, wann das Teileprogramm bestimmte Schalthandlungen der Werkzeugmaschine durch die PLC vornehmen lassen will. Dies geschieht durch Übergabe der entsprechenden Hilfsfunktionen mit ihren Parametern an die PLC-Schnittstelle. Die Verarbeitung der übergebenen Werte und Signale muss durch das PLC-Anwenderprogramm erfolgen.

## Hilfsfunktionen

Folgende Hilfsfunktionen können an die PLC übertragen werden:

| Hilfsfunktion | Adresse |
|---|---|
| Werkzeugwahl | T |
| Werkzeugkorrektur | D, DL |
| Vorschub | F / FA |
| Spindeldrehzahl | S |
| M-Funktionen | M |
| H-Funktionen | H |

Für jede Funktionsgruppe oder Einzelfunktion wird mit Maschinendaten festgelegt, ob die Ausgabe **vor**, **mit** oder **nach** der Verfahrbewegung ausgelöst wird.

Die PLC kann zu verschiedenem Quittungsverhalten für Hilfsfunktionsausgaben veranlasst werden.

## Eigenschaften

Wichtige Eigenschaften der Hilfsfunktionen sind in folgender Übersichtstabelle zusammengefasst:

| Funktion | Adresserweiterung | | Wert | | | Erläuterungen | Maximale Anzahl pro Satz |
|---|---|---|---|---|---|---|---|
| | Bedeutung | Bereich | Bereich | Typ | Bedeutung | | |
| M | - | 0 (implizit) | 0 ... 99 | INT | Funktion | Für den Wertebereich zwischen 0 und 99 ist die Adresserweiterung 0.<br>Zwingend ohne Adresserweiterung:<br>M0, M1, M2, M17, M30 | 5 |
| | Spindel-Nr. | 1 - 12 | 1 ... 99 | INT | Funktion | M3, M4, M5, M19, M70 mit Adresserweiterung Spindel-Nr. (z. B. `M2=5` ; Spindel-Halt für Spindel 2). <br>Ohne Spindel-Nr. gilt die Funktion für die Masterspindel. | |
| | Beliebig | 0 - 99 | 100 ... 2147483647 | INT | Funktion | Anwender-M-Funktion* | |
| S | Spindel-Nr. | 1 - 12 | 0 ... ± 1,8*10$^{308}$ | REAL | Drehzahl | Ohne Spindel-Nr. gilt die Funktion für die Masterspindel. | 3 |
| H | Beliebig | 0 - 99 | 0 ...<br>± 2147483647<br>± 1,8*10$^{308}$ | <br>INT<br>REAL | Beliebig | Funktionen haben im NC keine Wirkungen, ausschließlich durch PLC zu realisieren.* | 3 |
| T | Spindel-Nr (bei aktiver WZV) | 1 - 12 | 0 - 32000 (auch Werkzeugnamen bei aktiver WZV) | INT | Werkzeugwahl | Werkzeugnamen gehen nicht an die PLC-Schnittstelle. | 1 |
| D | - | - | 0 - 12 | INT | Werkzeugkorrekturwahl | D0: Abwahl<br>Vorbesetzung: D1 | 1 |
| DL | Ortsabhängige Korrektur | 1 - 6 | 0 ... ± 1,8*10$^{308}$ | REAL | Werkzeugfeinkorrekturwahl | Bezieht sich auf zuvor gewählte D-Nummer. | 1 |
| F | - | - | 0.001 - 999 999,999 | REAL | Bahnvorschub | | 6 |
| FA | Achs-Nr. | 1 - 31 | 0.001 - 999 999,999 | REAL | Achsvorschub | | |
| * Die Bedeutung der Funktionen wird vom Maschinenhersteller festgelegt (**siehe Angaben des Maschinenherstellers!**). | | | | | | | |

## Weitere Informationen

### Anzahl an Funktionsausgaben pro NC-Satz

In einem NC-Satz können maximal 10 Funktionsausgaben programmiert werden. Hilfsfunktionen können auch aus dem Aktionsteil von **Synchronaktionen** ausgegeben werden.

### Gruppierung

Die genannten Funktionen können zu Gruppen zusammengefasst werden. Für einige M-Befehle ist die Gruppeneinteilung bereits vorgegeben. Mit der Gruppierung kann das Quittungsverhalten festgelegt werden.

### Schnelle Funktionsausgaben (QU)

Funktionen, die nicht als schnelle Ausgaben projektiert wurden, können für einzelne Ausgaben mit dem Schlüsselwort `QU` als schnelle Ausgabe definiert werden. Der Programmablauf wird fortgesetzt, ohne auf die Quittung für die Ausführung der Zusatzfunktion zu warten (Transportquittung wird abgewartet). Hierdurch lassen sich unnötige Haltepunkte und Unterbrechungen der Fahrbewegungen vermeiden.

**Hinweis**

Für die Funktion "Schnelle Funktionsausgaben" müssen entsprechende Maschinendaten gesetzt sein (→ **Maschinenhersteller!**).

### Funktionsausgaben bei Fahrbewegungen

Die Übertragung von Informationen sowie das Warten auf entsprechende Reaktionen kosten Zeit und beeinflussen daher auch die Fahrbewegungen.

### Schnelle Quittung ohne Satzwechselverzögerung

Das Satzwechselverhalten kann durch Maschinendatum beeinflusst werden. Mit der Einstellung "ohne Satzwechselverzögerung" ergibt sich für schnelle Hilfsfunktionen folgendes Verhalten:

| Hilfsfunktionsausgabe | Verhalten |
|---|---|
| **vor** Bewegung | Der Satzübergang zwischen Sätzen mit schnellen Hilfsfunktionen erfolgt **ohne** Unterbrechung und **ohne** Geschwindigkeitsreduzierung. Die Ausgabe der Hilfsfunktionen erfolgt im ersten Interpolatortakt des Satzes. Der Folgesatz wird ohne Quittungsverzögerung abgefahren. |
| **während** Bewegung | Der Satzübergang zwischen Sätzen mit schnellen Hilfsfunktionen erfolgt **ohne** Unterbrechung und **ohne** Geschwindigkeitsreduzierung. Die Ausgabe der Hilfsfunktionen erfolgt während des Satzes. Der Folgesatz wird ohne Quittungsverzögerung abgefahren. |
| **nach** Bewegung | Die Bewegung kommt am Ende des Satzes zum Stillstand. Die Ausgabe der Hilfsfunktionen erfolgt am Satzende. Der Folgesatz wird ohne Quittungsverzögerung abgefahren. |

| ⚠ VORSICHT |
|---|
| **Funktionsausgaben im Bahnsteuerbetrieb**<br>Funktionsausgaben **vor** den Verfahrbewegungen unterbrechen den Bahnsteuerbetrieb (G64 / G641) und erzeugen für den vorherigen Satz einen Genauhalt.<br>Funktionsausgaben **nach** den Verfahrbewegungen unterbrechen den Bahnsteuerbetrieb (G64 / G641) und erzeugen für den aktuellen Satz einen Genauhalt.<br>**Wichtig:** Das Warten auf ein ausstehendes Quittungssignal von der PLC kann ebenfalls zur Unterbrechung des Bahnsteuerbetriebs führen, z. B. bei M-Befehlsfolgen in Sätzen mit extrem kurzen Bahnweglängen. |

## 2.13.1 M-Funktionen

Mit den M-Funktionen können z. B. Schalthandlungen wie "Kühlmittel EIN/AUS" und sonstige Funktionalitäten an der Maschine ausgelöst werden.

### Syntax

```
M<Wert>
M[<Adresserweiterung>]=<Wert>
```

### Bedeutung

| | | |
|---|---|---|
| `M`: | Adresse zur Programmierung der M-Funktionen | |
| `<Adresserweiterung>`: | Für einige M-Funktionen gilt die erweiterte Adressschreibweise (z. B. Angabe der Spindelnummer bei Spindelfunktionen). | |
| `<Wert>`: | Durch die Wertzuweisung (M-Funktionsnummer) erfolgt die Zuordnung zu einer bestimmten Maschinenfunktion. | |
| | Typ: | INT |
| | Wertebereich: | 0 ... 2147483647 (max. INT-Wert) |

### Vordefinierte M-Funktionen

Einige für den Programmablauf wichtige M-Funktionen sind im Standardumfang der Steuerung bereits vorbelegt:

| M-Funktion | Bedeutung |
|---|---|
| M0* | Programmierter Halt |
| M1* | Wahlweiser Halt |
| M2* | Programmende Hauptprogramm (wie M30) |
| M3 | Spindel Rechtslauf |
| M4 | Spindel Linkslauf |
| M5 | Spindel Halt |

| M-Funktion | Bedeutung |
|---|---|
| M6 | Werkzeugwechsel (Standardeinstellung) |
| M17* | Programmende Unterprogramm |
| M19 | Spindel positionieren |
| M30* | Programmende Hauptprogramm (wie M2) |
| M40 | Automatische Getriebeschaltung |
| M41 | Getriebestufe 1 |
| M42 | Getriebestufe 2 |
| M43 | Getriebestufe 3 |
| M44 | Getriebestufe 4 |
| M45 | Getriebestufe 5 |
| M70 | Spindel wird in den Achsbetrieb geschaltet |

**Hinweis**

Für die mit * gekennzeichneten Funktionen ist die erweiterte Adressschreibweise nicht zulässig.

Die Funktionen M0, M1, M2, M17 und M30 werden immer **nach** der Verfahrbewegung ausgelöst.

## Vom Maschinenhersteller definierte M-Funktionen

Alle freien M-Funktionsnummern können vom Maschinenhersteller belegt werden, z. B. mit Schaltfunktionen zur Steuerung von Spannvorrichtungen oder zum Ein- / Ausschalten weiterer Maschinenfunktionen.

**Hinweis**

Die den freien M-Funktionsnummern zugeordneten Funktionalitäten sind Maschinen-spezifisch. Eine bestimmte M-Funktion kann daher an unterschiedlichen Maschinen eine unterschiedliche Funktionalität besitzen.

Die an einer Maschine zur Verfügung stehenden M-Funktionen und deren Funktionalitäten sind den Angaben des Maschinenherstellers zu entnehmen.

## Beispiele

### Beispiel 1: Maximale Anzahl an M-Funktionen im Satz

```
Programmcode                Kommentar
N10 S...
N20 X... M3                 ; M-Funktion im Satz mit Achsbewegung,
                            ; Spindel läuft vor der X-Achsbewegung hoch.
N180 M789 M1767 M100 M102   ; Maximal 5 M-Funktionen im Satz.
M376
```

### Beispiel 2: M-Funktion als schnelle Ausgabe

| Programmcode | Kommentar |
|---|---|
| N10 H=QU(735) | ; Schnelle Ausgabe für H735. |
| N10 G1 F300 X10 Y20 G64 | |
| N20 X8 Y90 M=QU(7) | ; Schnelle Ausgabe für M7. |

M7 wurde als schnelle Ausgabe programmiert, so dass der Bahnsteuerbetrieb (G64) nicht unterbrochen wird.

**Hinweis**

Setzen Sie diese Funktion nur in Einzelfällen ein, da z. B. im Zusammenwirken mit anderen Funktionsausgaben die zeitliche Abstimmung verändert wird.

## Weitere Informationen zu den vordefinierten M-Befehlen

### Programmierter Halt: M0

Im NC-Satz mit `M0` wird die Bearbeitung angehalten. Jetzt können Sie z. B. Späne entfernen, nachmessen usw.

### Programmierter Halt 1 - Wahlweiser Halt: M1

M1 ist einstellbar über:

- HMI/Dialog "Programmbeeinflussung"
  oder
- NC/PLC-Schnittstelle

Die Programmbearbeitung der NC wird jeweils bei den programmierten Sätzen angehalten.

### Programmierter Halt 2 - Eine zu M1 assoziierte Hilfsfunktion mit Halt im Programmablauf

Der Programmierte Halt 2 ist über HMI/Dialog "Programmbeeinflussung" einstellbar und erlaubt jederzeit eine Unterbrechung von technologischen Abläufen am Ende des zu bearbeitenden Teils. Damit kann der Bediener in die laufende Produktion eingreifen um z. B. Fließspäne zu beseitigen.

### Programmende: M2, M17, M30

Ein Programm wird mit `M2`, `M17` oder `M30` beendet. Wird das Hauptprogramm aus einem anderen Programm (als Unterprogramm) aufgerufen, wirkt `M2` / `M30` wie `M17` und umgekehrt, d. h. `M17` wirkt im Hauptprogramm wie `M2` / `M30`.

### Spindelfunktionen: M3, M4, M5, M19, M70

Für alle Spindelfunktionen gilt die erweiterte Adressschreibweise mit Angabe der Spindelnummer.

### Beispiel:

| Programmcode | Kommentar |
|---|---|
| M2=3 | ; Spindeldrehung rechts für die zweite Spindel |

Ist keine Adresserweiterung programmiert, gilt die Funktion für die Masterspindel.

# 2.14 Ergänzende Befehle

## 2.14.1 Meldung ausgeben (MSG)

Mit der Anweisung `MSG()` kann eine beliebige Zeichenkette vom Teileprogramm aus als Meldung an den Bediener ausgegeben werden.

### Syntax

```
MSG("<Meldungstext>"[,<Ausführung>])
...
MSG()
```

### Bedeutung

| | | | |
|---|---|---|---|
| `MSG`: | Vordefinierter Unterprogrammaufruf zur Ausgabe einer Meldung | | |
| `<Meldungstext>`: | Beliebige Zeichenkette zur Anzeige als Meldung | | |
| | Typ: | STRING | |
| | Maximale Länge: | 124 Zeichen; die Anzeige erfolgt in zwei Zeilen (2*62 Zeichen) | |
| | Im Meldungstext können durch Verwendung des Verkettungsoperators "<<" auch Variablen ausgegeben werden. | | |
| `<Ausführung>`: | Parameter zum Festlegen des Zeitpunkts, zu dem das Schreiben der Meldung ausgeführt wird (optional) | | |
| | Typ: | INT | |
| | Wert: | 0 (Grund-einstellung) | Für das Schreiben der Meldung wird kein eigener Hauptlaufsatz erzeugt. Es erfolgt im nächsten ausführbaren NC-Satz. Keine Unterbrechung eines aktiven Bahnsteuerbetriebs. |
| | | 1 | Für das Schreiben der Meldung wird ein eigener Hauptlaufsatz erzeugt. Ein aktiver Bahnsteuerbetrieb wird unterbrochen. |
| `MSG()`: | Durch Programmierung von `MSG()` ohne Meldungstext wird die aktuelle Meldung wieder gelöscht. Ohne Löschen bleibt die Anzeige solange bestehen, bis die nächste Meldung ansteht. | | |

**Hinweis**

Soll eine Meldung in der auf der Bedienoberfläche aktiven Sprache erfolgen, benötigt der Anwender Informationen über die aktuell auf dem HMI eingestellten Sprache. Diese Information kann im Teileprogramm und in Synchronaktionen über die Systemvariable $AN_LANGUAGE_ON_HMI (Seite 1326)abgefragt werden.

## Beispiele

### Beispiel 1: Meldung ausgeben / löschen

| Programmcode | Kommentar |
|---|---|
| N10 G91 G64 F100 | ; Bahnsteuerbetrieb |
| N20 X1 Y1 | |
| N... X... Y... | |
| N20 MSG ("Bearbeitung Teil 1") | ; Die Meldung wird erst mit N30 ausgegeben. |
| | ; Der Bahnsteuerbetrieb bleibt erhalten. |
| N30 X... Y... | |
| N... X... Y... | |
| N400 X1 Y1 | |
| N410 MSG ("Bearbeitung Teil 2",1) | ; Die Meldung wird mit N410 ausgegeben. |
| | ; Der Bahnsteuerbetrieb wird unterbrochen. |
| N420 X1 Y1 | |
| N... X... Y... | |
| N900 MSG () | ; Meldung löschen. |

### Beispiel 2: Meldungstext mit Variable

| Programmcode | Kommentar |
|---|---|
| N10 R12=$AA_IW[X] | ; Aktuelle Position der X-Achse in R12. |
| N20 MSG("Position der X-Achse"<<R12<<"prüfen") | ; Meldung mit Variable R12 ausgeben. |
| ... | |
| N90 MSG () | ; Meldung aus N20 löschen. |

## 2.14.2 Zeichenkette in BTSS-Variable schreiben (WRTPR)

Mit der Funktion `WRTPR()` ist es möglich, eine beliebige Zeichenkette vom Teileprogramm aus in die BTSS-Variable progProtText zu schreiben.

## Syntax

```
WRTPR(<String>[,<ExecTime>])
```

## Bedeutung

| | | | |
|---|---|---|---|
| WRTPR: | Funktionsaufruf zur Ausgabe einer Zeichenkette. | | |
| <String>: | Beliebige Zeichenkette, die in die BTSS-Variable progProtText geschrieben wird. | | |
| | Typ: | STRING | |
| | Maximale Länge: | 128 Zeichen | |
| <ExecTime>: | Optionaler Parameter zum Festlegen des Zeitpunkts, zu dem das Schreiben der Zeichenkette ausgeführt wird. | | |
| | Typ: | INT | |
| | Wertebereich: | 0, 1 | |
| | | 0 (Default) | Für das Schreiben der Zeichenkette wird kein eigener Hauptlaufsatz erzeugt. Es erfolgt im nächsten ausführbaren NC-Satz. Keine Unterbrechung eines aktiven Bahnsteuerbetriebs. |
| | | 1 | Für das Schreiben der Zeichenkette wird ein eigener Hauptlaufsatz erzeugt. Ein aktiver Bahnsteuerbetrieb wird unterbrochen. |

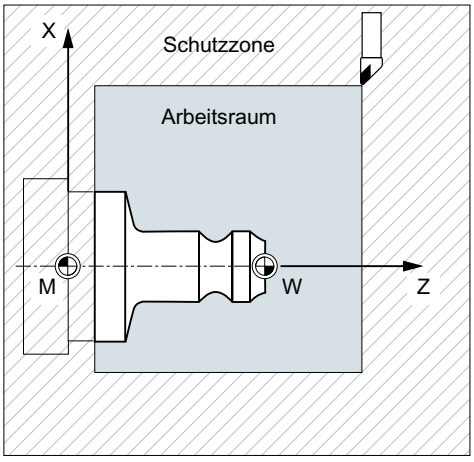
## Beispiele

```
Programmcode                    Kommentar
N10 G91 G64 F100                ; Bahnsteuerbetrieb
N20 X1 Y1
N30 WRTPR("N30")                ; Die Zeichenkette "N30" wird erst in N40 geschrieben.
                                ; Der Bahnsteuerbetrieb bleibt erhalten.
N40 X1 Y1
N50 WRTPR("N50",1)              ; Die Zeichenkette "N50" wird in N50 geschrieben.
                                ; Der Bahnsteuerbetrieb wird unterbrochen.
N60 X1 Y1
```

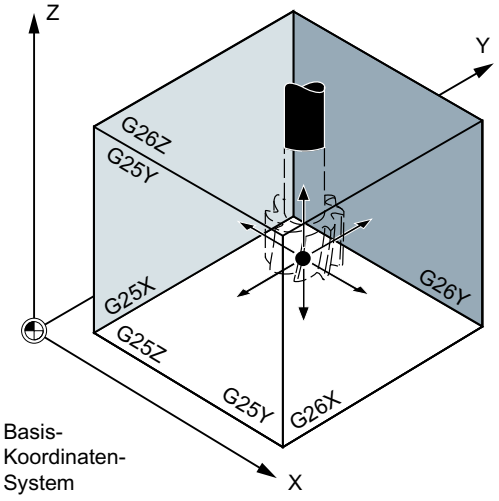
## 2.14.3 Arbeitsfeldbegrenzung

### 2.14.3.1 Arbeitsfeldbegrenzung im BKS (G25/G26, WALIMON, WALIMOF)

Mit G25/G26 lässt sich der Arbeitsbereich (Arbeitsfeld, Arbeitsraum), in dem das Werkzeug verfahren soll, in allen Kanalachsen begrenzen. Die Bereiche außerhalb der mit G25/G26 definierten Arbeitsfeldgrenzen sind für Werkzeugbewegungen gesperrt.




Die Koordinatenangaben für die einzelnen Achsen gelten im Basiskoordinatensystem:




Die Arbeitsfeldbegrenzung für alle gültig gesetzten Achsen muss mit dem Befehl WALIMON programmiert sein. Mit WALIMOF ist die Arbeitsfeldbegrenzung unwirksam. WALIMON ist Standardeinstellung und muss nur programmiert werden, wenn zuvor die Arbeitsfeldbegrenzung ausgeschaltet wurde.

## Syntax

```
G25 X…Y…Z…
G26 X…Y…Z…
WALIMON
...
WALIMOF
```

## Bedeutung

| | |
|---|---|
| G25: | **Untere** Arbeitsfeldbegrenzung<br>Wertzuweisung in Kanalachsen im Basiskoordinatensystem |
| G26: | **Obere** Arbeitsfeldbegrenzung<br>Wertzuweisung in Kanalachsen im Basiskoordinatensystem |
| X…Y…Z… : | Untere bzw. obere Arbeitsfeldgrenzen für die einzelnen Kanalachsen<br>Die Angaben beziehen sich auf das Basiskoordinatensystem. |
| WALIMON: | Arbeitsfeldbegrenzung für alle Achsen **ein**schalten |
| WALIMOF: | Arbeitsfeldbegrenzung für alle Achsen **aus**schalten |

Neben der programmierbaren Eingabe der Werte über G25/G26 ist auch eine Eingabe über achsspezifische Settingdaten möglich:

SD43420 $SA_WORKAREA_LIMIT_PLUS (Arbeitsfeldbegrenzung plus)

SD43430 $SA_WORKAREA_LIMIT_MINUS (Arbeitsfeldbegrenzung minus)

Aktivierung und Deaktivierung der über SD43420 und SD43430 parametrierten Arbeitsfeldbegrenzung erfolgen richtungsspezifisch über die sofort wirksamen achsspezifischen Settingdaten:

SD43400 $SA_WORKAREA_PLUS_ENABLE (Arbeitsfeldbegrenzung in positiver Richtung aktiv)

SD43410 $SA_WORKAREA_MINUS_ENABLE (Arbeitsfeldbegrenzung in negativer Richtung aktiv)

Durch die richtungsspezifische Aktivierung/Deaktivierung ist es möglich, den Arbeitsbereich für eine Achse nur in einer Richtung zu begrenzen.

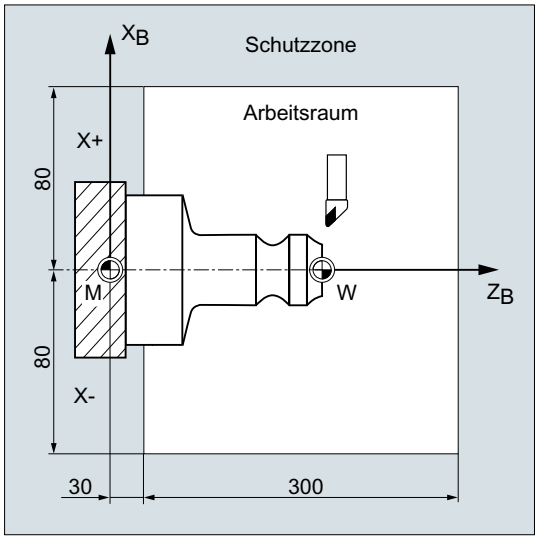
**Hinweis**

Die mit G25/G26 programmierte Arbeitsfeldbegrenzung hat Vorrang und überschreibt die in SD43420 und SD43430 eingetragenen Werte.

**Hinweis**

Mit G25/G26 können unter der Adresse S auch Grenzwerte für Spindeldrehzahlen programmiert werden. Mehr Informationen hierzu siehe " Programmierbare Spindeldrehzahlbegrenzung (G25, G26) (Seite 106) ".

## Beispiel




Durch die Arbeitsfeldbegrenzung mit G25/26 wird der Arbeitsraum einer Drehmaschine so begrenzt, dass die umliegenden Einrichtungen wie Revolver, Messstation usw. vor Beschädigung geschützt sind.

Grundeinstellung: WALIMON

| Programmcode | Kommentar |
|---|---|
| N10 G0 G90 F0.5 T1 | |
| N20 G25 X-80 Z30 | ; Festlegung der unteren Begrenzung für die einzelnen Koordinatenachsen |
| N30 G26 X80 Z330 | ; Festlegung der oberen Begrenzung |
| N40 L22 | ; Abspanprogramm |
| N50 G0 G90 Z102 T2 | ; zum Werkzeugwechselpunkt |
| N60 X0 | |
| N70 WALIMOF | ; Arbeitsfeldbegrenzung ausschalten |
| N80 G1 Z-2 F0.5 | ; Bohren |
| N90 G0 Z200 | ; zurück |
| N100 WALIMON | ; Arbeitsfeldbegrenzung einschalten |
| N110 X70 M30 | ; Programmende |

## Weitere Informationen

### Bezugspunkt am Werkzeug

Bei aktiver Werkzeuglängenkorrektur wird als Bezugspunkt die Werkzeugspitze überwacht, ansonsten der Werkzeugträgerbezugspunkt.

Die Berücksichtigung des Werkzeugradius muss separat aktiviert werden. Dies erfolgt über das kanalspezifische Maschinendatum:

MD21020 $MC_WORKAREA_WITH_TOOL_RADIUS

Falls der Werkzeug-Bezugspunkt außerhalb des durch die Arbeitsfeldbegrenzung definierten Arbeitsraums steht oder diesen Bereich verlässt, wird der Programmablauf gestoppt.

**Hinweis**

Wenn Transformationen aktiv sind, kann die Berücksichtigung der Werkzeugdaten (Werkzeuglänge und Werkzeugradius) vom beschriebenen Verhalten abweichen.

**Programmierbare Arbeitsfeldbegrenzung, G25/G26**

Für jede Achse lassen sich eine obere (G26) und eine untere (G25) Arbeitsfeldbegrenzung festlegen. Diese Werte gelten sofort und bleiben bei entsprechender MD-Einstellung (→ MD10710 $MN_PROG_SD_RESET_SAVE_TAB) nach RESET und Wiedereinschalten erhalten.

**Hinweis**

Mit der vordefinierten Funktion CALCPOSI (Seite 591) lässt sich vor Verfahrbewegungen prüfen, ob der vorgesehene Weg unter Berücksichtigung von Arbeitsfeldbegrenzungen und/oder Schutzbereichen abgefahren wird.

### 2.14.3.2 Arbeitsfeldbegrenzung im WKS/ENS (WALCS0 ... WALCS10)

Die "Arbeitsfeldbegrenzung im WKS/ENS" ermöglicht eine flexible werkstückspezifische Begrenzung des Verfahrbereichs der Kanalachsen im Werkstückkoordinatensystem (WKS) oder Einstellbaren Nullpunktsystem (ENS). Sie ist hauptsächlich für den Einsatz im Bereich konventioneller Drehmaschinen gedacht.

#### Voraussetzung

Die Kanalachsen müssen referenziert sein.

#### Arbeitsfeldbegrenzungsgruppe

Damit beim Umschalten von Achszuordnungen, z. B. beim Ein/Ausschalten von Transformationen oder des aktiven Frames, die achsspezifischen Arbeitsfeldgrenzen nicht für alle Kanalachsen neu geschrieben werden müssen, stehen Arbeitsfeldbegrenzungsgruppen zur Verfügung.

Eine Arbeitsfeldbegrenzungsgruppe umfasst folgende Daten:

- Arbeitsfeldgrenzen für alle Kanalachsen
- Bezugssystem der Arbeitsfeldbegrenzung

#### Syntax

```
...
$P_WORKAREA_CS_COORD_SYSTEM[<WALimNo>]=<Value>
$P_WORKAREA_CS_PLUS_ENABLE[<WALimNo>,<Ax>]=<Value>
```

```
$P_WORKAREA_CS_LIMIT_PLUS[<WALimNo>,<Ax>]=<Value>
$P_WORKAREA_CS_MINUS_ENABLE[<WALimNo>,<Ax>]=<Value>
$P_WORKAREA_CS_LIMIT_MINUS[<WALimNo>,<Ax>]=<Value>
...
WALCS<n>
...
WALCS0
```

## Bedeutung

| $P_WORKAREA_CS_COORD_SYSTEM[<WALimNo>]=<Value> | | | |
|---|---|---|---|
| Koordinatensystem, auf das sich die Arbeitsfeldbegrenzung einer Gruppe bezieht | | | |
| <WALimNo>: | Arbeitsfeldbegrenzungsgruppe | | |
| | Typ: | INT | |
| | Wertebereich: | 0 (Gruppe 1) ... 9 (Gruppe 10) | |
| <Value>: | Wert vom Typ INT | | |
| | 1 | Werkstück-Koordinatensystem (WKS) | |
| | 3 | Einstellbares Nullpunktsystem (ENS) | |

| $$P_WORKAREA_CS_PLUS_ENABLE[<WALimNo>,<Ax>]=<Value> | | |
|---|---|---|
| Arbeitsfeldbegrenzung in **positiver** Achsrichtung für die angegebene Kanalachse freigeben | | |
| <WALimNo>: | Arbeitsfeldbegrenzungsgruppe | |
| | Typ: | INT |
| | Wertebereich: | 0 (Gruppe 1) ... 9 (Gruppe 10) |
| <Ax>: | Kanalachsname | |
| <Value>: | Wert vom Typ BOOL | |
| | 0 (FALSE) | keine Freigabe |
| | 1 (TRUE) | Freigabe |

| $P_WORKAREA_CS_MINUS_ENABLE[<WALimNo>,<Ax>]=<Value> | | |
|---|---|---|
| Arbeitsfeldbegrenzung in **negativer** Achsrichtung für die angegebene Kanalachse freigeben | | |
| <WALimNo>: | Arbeitsfeldbegrenzungsgruppe | |
| | Typ: | INT |
| | Wertebereich: | 0 (Gruppe 1) ... 9 (Gruppe 10) |
| <Ax>: | Kanalachsname | |
| <Value>: | Wert vom Typ BOOL | |
| | 0 (FALSE) | keine Freigabe |
| | 1 (TRUE) | Freigabe |

| $P_WORKAREA_CS_LIMIT_PLUS[<WALimNo>,<Ax>]=<Value> |
|---|
| Arbeitsfeldbegrenzung in **positiver** Richtung der angegebenen Kanalachse |

| | | |
|---|---|---|
| <WALimNo>: | Arbeitsfeldbegrenzungsgruppe | |
| | Typ: | INT |
| | Wertebereich: | 0 (Gruppe 1) ... 9 (Gruppe 10) |
| <Ax>: | Kanalachsname | |
| <Value>: | Wert vom Typ REAL | |

| $P_WORKAREA_CS_LIMIT_MINUS[<WALimNo>,<Ax>]=<Value> | | |
|---|---|---|
| Arbeitsfeldbegrenzung in **negativer** Richtung der angegebenen Kanalachse | | |
| <WALimNo>: | Arbeitsfeldbegrenzungsgruppe | |
| | Typ: | INT |
| | Wertebereich: | 0 (Gruppe 1) ... 9 (Gruppe 10) |
| <Ax>: | Kanalachsname | |
| <Value>: | Wert vom Typ REAL | |

| | | |
|---|---|---|
| WALCS<n>: | Einschalten der Arbeitsfeldbegrenzungen einer Arbeitsfelbegrenzungsgruppe | |
| <n>: | Nummer der Arbeitsfeldbegrenzungsgruppe | |
| | Wertebereich: | 1 ... 10 |

| | |
|---|---|
| WALCS0: | Ausschalten der im Kanal aktiven Arbeitsfeldbegrenzungen |

**Hinweis**

Die tatsächlich verfügbare Anzahl an Arbeitsfeldbegrenzungsgruppen ist abhängig von der Projektierung (→ siehe Angaben des Maschinenherstellers).

## Beispiel

Im Kanal sind 3 Achsen definiert: X, Y und Z

Es soll eine Arbeitsfeldbegrenzungsgruppe Nr. 2 definiert und anschließend aktiviert werden, in der die Achsen im WKS nach folgenden Vorgaben begrenzt werden:

- X-Achse in Plus-Richtung: 10 mm
- X-Achse in Minus-Richtung: keine Begrenzung
- Y-Achse in Plus-Richtung: 34 mm
- Y-Achse in Minus-Richtung: -25 mm
- Z-Achse in Plus-Richtung: keine Begrenzung
- Z-Achse in Minus-Richtung: -600 mm

```
Programmcode                                    Kommentar

...
```

| Programmcode | Kommentar |
|---|---|
| N51 $P_WORKAREA_CS_COORD_SYSTEM[1]=1 | ; Die Arbeitsfeldbegrenzung der Arbeitsfeldbegrenzungsgruppe 2 gilt im WKS. |
| N60 $P_WORKAREA_CS_PLUS_ENABLE[1,X]=TRUE | |
| N61 $P_WORKAREA_CS_LIMIT_PLUS[1,X]=10 | |
| N62 $P_WORKAREA_CS_MINUS_ENABLE[1,X]=FALSE | |
| N70 $P_WORKAREA_CS_PLUS_ENABLE[1,Y]=TRUE | |
| N73 $P_WORKAREA_CS_LIMIT_PLUS[1,Y]=34 | |
| N72 $P_WORKAREA_CS_MINUS_ENABLE[1,Y]=TRUE | |
| N73 $P_WORKAREA_CS_LIMIT_MINUS[1,Y]=-25 | |
| N80 $P_WORKAREA_CS_PLUS_ENABLE[1,Z]=FALSE | |
| N82 $P_WORKAREA_CS_MINUS_ENABLE[1,Z]=TRUE | |
| N83 $P_WORKAREA_CS_LIMIT_PLUS[1,Z]=-600 | |
| ... | |
| N90 WALCS2 | ; Arbeitsfeldbegrenzungsgruppe 2 einschalten. |
| ... | |

### Weitere Informationen

#### Wirksamkeit

Die Arbeitsfeldbegrenzung mit `WALCS1` - `WALCS10` wirkt unabhängig von der Arbeitsfeldbegrenzung mit `WALIMON`. Wenn beide Funktionen aktiv sind, wirkt diejenige Begrenzung, auf die die Achsbewegung als erstes trifft.

#### Bezugspunkt am Werkzeug

Die Berücksichtigung der Werkzeugdaten (Werkzeuglänge und Werkzeugradius) und damit der Bezugspunkt am Werkzeug bei der Überwachung der Arbeitsfeldbegrenzung entspricht dem Verhalten bei der Arbeitsfeldbegrenzung mit `WALIMON`.

## 2.14.4 Referenzpunktfahren (G74)

Nach dem Einschalten der Maschine müssen (bei Verwendung von inkrementellen Wegmesssystemen) alle Achsschlitten auf ihre Referenzmarke gefahren werden. Erst dann können Fahrbewegungen programmiert werden.

Mit `G74` kann das Referenzpunktfahren im NC-Programm durchgeführt werden.

### Syntax

`G74 X1=0 Y1=0 Z1=0 A1=0 …` ; Programmierung im eigenen NC-Satz

### Bedeutung

| | |
|---|---|
| `G74`: | G-Befehlsaufruf Referenzpunktfahren |
| `X1=0 Y1=0 Z1=0 …`: | Die angegebene Maschinenachsadresse `X1`, `Y1`, `Z1` … für **Linearachsen** wird in den Referenzpunkt gefahren |
| `A1=0 B1=0 C1=0 …`: | Die angegebene Maschinenachsadresse `A1`, `B1`, `C1` … für **Rundachsen** wird in den Referenzpunkt gefahren |

**Hinweis**

Vor der Referenzpunktfahrt darf keine Transformation für eine Achse programmiert sein, die mit `G74` auf die Referenzmarke gefahren werden soll.

Die Transformation wird mit dem Befehl `TRAFOOF` ausgeschaltet.

### Beispiel

Beim Wechsel des Messsystems wird der Referenzpunkt angefahren und der Werkstücknullpunkt eingerichtet.

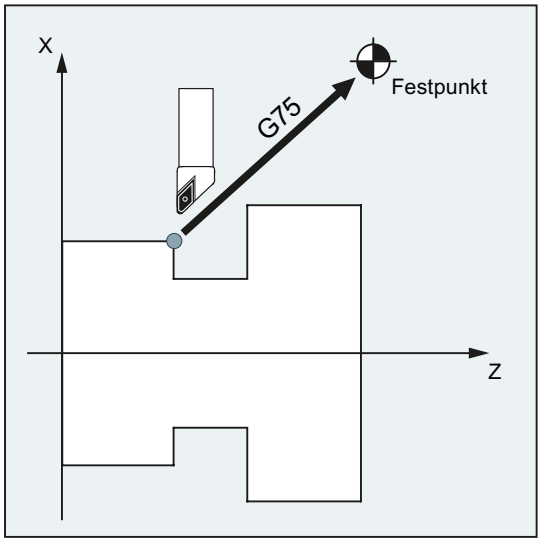
| **Programmcode** | **Kommentar** |
|---|---|
| `N10 SPOS=0` | `; Spindel in Lageregelung` |
| `N20 G74 X1=0 Y1=0 Z1=0 C1=0` | `; Referenzpunktfahrt für Linearachsen und Rundachsen` |
| `N30 G54` | `; Nullpunktverschiebung` |
| `N40 L47` | `; Abspanprogramm` |
| `N50 M30` | `; Programmende` |

## 2.14.5 Festpunkt anfahren (G75)

Mit dem satzweise wirksamen Befehl `G75` können Achsen einzeln und unabhängig voneinander auf feste Punkte im Maschinenraum gefahren werden, z. B. auf Werkzeugwechselpunkte, Beladepunkte, Palettenwechselpunkte etc.

Die Festpunkte sind Positionen im Maschinenkoordinatensystem, die in Maschinendaten (MD30600 $MA_FIX_POINT_POS[n]) abgelegt sind. Pro Achse können maximal 4 Festpunkte definiert sein.

Die Festpunkte können aus jedem NC-Programm unabhängig von aktuellen Werkzeug- oder Werkstückpositionen angefahren werden. Vor der Bewegung der Achsen wird ein interner Vorlaufstopp durchgeführt.




## Voraussetzungen

Für das Anfahren von Festpunkten mit `G75` müssen folgende Voraussetzungen erfüllt sein:

- Die Festpunktkoordinaten müssen exakt ermittelt und in Maschinendaten hinterlegt sein.
- Die Festpunkte müssen innerhalb des gültigen Verfahrbereichs liegen (→ Software-Endschalter-Grenzen beachten!)
- Die zu verfahrenden Achsen müssen referenziert sein.
- Es darf keine Werkzeugradiuskorrektur aktiv sein.
- Es darf keine kinematische Transformation aktiv sein.
- Die zu verfahrenden Achsen dürfen an keiner aktiven Transformation beteiligt sein.
- Keine der zu verfahrenden Achsen darf Folgeachse einer aktiven Kopplung sein.
- Keine der zu verfahrenden Achsen darf Achse eines Gantry-Verbundes sein.
- Compile–Zyklen dürfen keinen Bewegungsanteil aufschalten.

## Syntax

```
G75 <Achsname><Achsposition> ... FP=<n>
```

## Bedeutung

| | |
|---|---|
| `G75`: | Festpunkt anfahren |
| `<Achsname>`: | Name der Maschinenachse, die zum Festpunkt verfahren werden soll<br>Es sind alle Achsbezeichner zulässig. |
| `<Achsposition>`: | Der Positionswert ist ohne Bedeutung. Es wird daher in der Regel der Wert "0" angegeben. |

| | | | |
|---|---|---|---|
| FP=: | Festpunkt, der angefahren werden soll | | |
| | <n>: | Festpunktnummer | |
| | | Wertebereich: | 1, 2, 3, 4 |
| | **Hinweis:**<br>Wenn kein FP=<n> oder keine Festpunktnummer oder wenn FP=0 programmiert ist, wird dies wie FP=1 interpretiert und es wird Festpunkt 1 angefahren. | | |

**Hinweis**

In einem G75-Satz können auch mehrere Achsen programmiert werden. Die Achsen werden dann gleichzeitig auf den angegebenen Festpunkt verfahren.

**Hinweis**

Der Wert der Adresse FP darf nicht größer sein als die Anzahl der festgelegten Festpunkte für jede programmierte Achse (MD30610 $MA_NUM_FIX_POINT_POS).

## Beispiel

Für einen Werkzeugwechsel sollen die Achsen X (= AX1) und Z (= AX3) auf die feste Maschinenachsposition 1 mit X = 151,6 und Z = -17,3 fahren.

Maschinendaten:

- MD30600 $MA_FIX_POINT_POS[AX1,0] = 151.6
- MD30600 $MA_FIX_POINT[AX3,0] = 17.3

NC–Programm:

```
Programmcode                Kommentar
…
N100 G55                    ; Einstellbare Nullpunktverschiebung aktivieren.
N110 X10 Y30 Z40            ; Positionen im WKS anfahren.
N120 G75 X0 Z0 FP=1 M0      ; Die X-Achse fährt auf 151,6
                            ; und die Z-Achse fährt auf 17,3 (im MKS).
                            ; Jede Achse fährt für sich mit maximaler Geschwindig-
                            keit.
                            ; In diesem Satz dürfen keine zusätzlichen Bewegungen ak-
                            tiv sein.
                            ; Damit nach dem Erreichen der Endpositionen
                            ; weiterhin keine Zusatzbewegungen durchgeführt werden,
                            ; ist hier ein Stopp eingefügt.
N130 X10 Y30 Z40            ; Es wird wieder die Position von N110 angefahren.
                            ; Die Nullpunktverschiebung ist wieder aktiv.
…
```

**Hinweis**

Ist die Funktion "Werkzeugverwaltung mit Magazinen" aktiv, reicht die Hilfsfunktion `T…` bzw. `M...` (typischerweise `M6`) zum Auslösen der Satzwechselsperre am Ende der `G75`–Bewegung nicht aus.

Grund: Bei der Einstellung "Werkzeugverwaltung mit Magazinen ist aktiv" werden die Hilfsfunktionen für den Werkzeugwechsel nicht an die PLC ausgegeben.

## Weitere Informationen

### G75

Die Achsen werden als Maschinenachsen im Eilgang verfahren. Die Bewegung wird intern durch die Funktionen "SUPA" (Unterdrückung aller Frames) und "G0 RTLIOF" (Eilgangbewegung mit Einzelachsinterpolation) abgebildet.

Wenn die Bedingungen für "RTLIOF" (Einzelachsinterpolation) nicht erfüllt sind, wird der Festpunkt als Bahn angefahren.

Bei Erreichen des Festpunkts kommen die Achsen innerhalb des Toleranzfensters "Genauhalt fein" zum Stehen.

#### Parametrierbare Dynamik für G75

Für Positionierbewegungen auf Festpunktpositionen (G75) kann über nachfolgendes Maschinendatum der gewünschte Dynamikmodus eingestellt werden:

MD18960 $MN_POS_DYN_MODE (Art der Positionierachsdynamik)

#### Axiale Zusatzbewegungen

Die folgenden axialen Zusatzbewegungen werden zum Zeitpunkt der Interpretation des G75-Satzes berücksichtigt:

- externe Nullpunktverschiebung
- DRF
- Synchronisationsoffset ($AA_OFF)

Danach dürfen sich die Zusatzbewegungen der Achsen nicht ändern, bis das Ende der Verfahrbewegung durch den G75-Satz erreicht ist.

Zusatzbewegungen nach der Interpretation des G75-Satzes führen zu einer entsprechenden Verschiebung des angefahrenen Festpunkts.

Folgende Zusatzbewegungen werden unabhängig vom Interpretationszeitpunkt nicht berücksichtigt und führen zu einer entsprechenden Verschiebung der Zielposition:

- Online-Werkzeugkorrektur
- Zusatzbewegungen aus Compile–Zyklen im BKS wie MKS

#### Aktive Frames

Alle aktiven Frames werden ignoriert. Es wird im Maschinenkoordinatensystem verfahren.

#### Arbeitsfeldbegrenzung im WKS/ENS

Die Koordinatensystem-spezifische Arbeitsfeldbegrenzung (WALCS0 ... WALCS10) wirkt in dem Satz mit G75 nicht. Der Zielpunkt wird als Startpunkt des nachfolgenden Satzes überwacht.

#### Achs-/Spindelbewegungen mit POSA/SPOSA

Wenn programmierte Achsen/Spindeln vorher mit POSA bzw. SPOSA verfahren wurden, werden diese Bewegungen vor dem Anfahren des Festpunkts erst zu Ende gefahren.

#### Spindelfunktionen im G75-Satz

Wenn die Spindel vom "Festpunkt anfahren" ausgenommen ist, dann können im G75-Satz zusätzlich Spindelfunktionen programmiert werden (z. B. Positionierung mit SPOS/SPOSA).

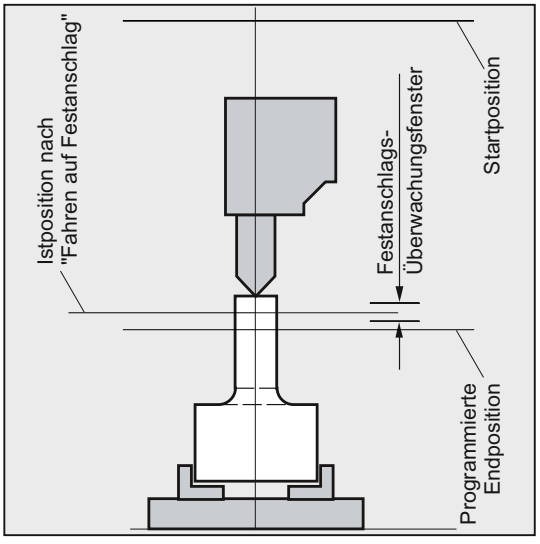
#### Modulo-Achsen

Bei Modulo-Achsen wird der Festpunkt auf kürzestem Weg angefahren.

## 2.14.6 Fahren auf Festanschlag (FXS, FXST, FXSW)

### Funktion

Mit Hilfe der Funktion "Fahren auf Festanschlag" ist es möglich, definierte Kräfte für das Klemmen von Werkstücken aufzubauen, wie sie z. B. bei Reitstöcken, Pinolen und Greifern notwendig sind. Außerdem können mit der Funktion mechanische Referenzpunkte angefahren werden.




Bei hinreichend reduziertem Moment sind auch einfache Messvorgänge möglich, ohne dass ein Taster angeschlossen werden muss. Die Funktion "Fahren auf Festanschlag" kann für Achsen und als Achsen fahrbare Spindeln eingesetzt werden.

## Syntax

```
FXS[<Achse>]=…
FXST[<Achse>]=…
FXSW[<Achse>]=…
FXS[<Achse>]=… FXST[<Achse>]=…
FXS[<Achse>]=… FXST[<Achse>]=… FXSW[<Achse>]=…
```

## Bedeutung

| | |
|---|---|
| FXS: | Befehl zum Ein- und Ausschalten der Funktion "Fahren auf Festanschlag" |
| | FXS[<Achse>]=1: Funktion einschalten |
| | FXS=[<Achse>]=0: Funktion ausschalten |
| FXST: | Optionaler Befehl zum Einstellen des Klemmmoments |
| | Angabe in % vom maximalen Moment des Antriebs. |
| FXSW: | Optionaler Befehl zum Einstellen der Fensterbreite für die Festanschlag-Überwachung |
| | Angabe in mm, inch oder Grad. |
| <Achse>: | Maschinenachsnamen |
| | Programmiert werden Maschinenachsen (X1, Y1, Z1 usw.) |

**Hinweis**

Die Befehle FXS, FXST und FXSW sind modal wirksam.

Die Programmierung von FXST und FXSW ist optional: Erfolgt keine Angabe, gilt jeweils der zuletzt programmierte Wert bzw. der im entsprechenden Maschinendatum eingestellte Wert.

## Fahren auf Festanschlag aktivieren: FXS[<Achse>] = 1

Die Bewegung zum Zielpunkt kann als Bahn- oder Positionierachsbewegung beschrieben werden. Bei Positionierachsen ist die Funktion auch über Satzgrenzen hinaus möglich.

Fahren auf Festanschlag kann auch für mehrere Achsen gleichzeitig und parallel zur Bewegung anderer Achsen stattfinden. Der Festanschlag muss zwischen Start- und Zielposition liegen.

**ACHTUNG**

**Kollisionsgefahr**

Sobald die Funktion "Fahren auf Festanschlag" für eine Achse / Spindel aktiviert wurde, darf für diese Achse keine neue Position programmiert werden.

Spindeln müssen vor Anwahl der Funktion in den lagegeregelten Betrieb geschaltet werden.

Beispiel:

| Programmcode | Kommentar |
|---|---|
| X250 Y100 F100 FXS[X1]=1 FXST[X1]=12.3 FXSW[X1]=2 | ; Achse X1 wird mit Vorschub F100 (Angabe optional) auf Zielposition X=250 mm gefahren.<br>Das Klemmmoment beträgt 12.3% vom maximalen Antriebsmoment, die Überwachung erfolgt in einem Fenster der Breite 2 mm. |
| ... | |

## Fahren auf Festanschlag deaktivieren: FXS[<Achse>] = 0

Die Abwahl der Funktion löst einen Vorlaufstopp aus.

Im Satz mit `FXS[<Achse>]=0` dürfen und sollen Verfahrbewegungen stehen.

**ACHTUNG**

**Kollisionsgefahr**

Die Verfahrbewegung auf Rückzugsposition muss vom Festanschlag wegführen, sonst sind Anschlag- oder Maschinenbeschädigung möglich.

Der Satzwechsel erfolgt nach Erreichen der Rückzugsposition. Wird keine Rückzugsposition angegeben, findet der Satzwechsel sofort nach dem Abschalten der Momentenbegrenzung statt.

Beispiel:

| Programmcode | Kommentar |
|---|---|
| X200 Y400 G01 G94 F2000 FXS[X1]=0 | ; Achse X1 wird von Festanschlag auf Position X=200mm zurückgezogen. Alle weiteren Angaben sind optional. |
| ... | |

## Klemmmoment (FXST) und Überwachungsfenster (FXSW)

Eine programmierte Momentenbegrenzung FXST wirkt ab Satzbeginn, d. h. auch das Anfahren des Anschlags erfolgt mit reduziertem Moment. FXST und FXSW können zu einem beliebigen Zeitpunkt im Teileprogramm programmiert bzw. geändert werden. Die Änderungen werden vor Verfahrbewegungen, die im gleichen Satz stehen, wirksam.

| ACHTUNG |
| --- |
| **Kollisionsgefahr**<br>Wenn ein neues Festanschlags-Überwachungsfenster programmiert wird, ändert sich nicht nur die Fensterbreite, sondern auch der Bezugspunkt für die Fenstermitte, wenn sich die Achse vorher bewegt hat. Die Istposition der Maschinenachse bei Änderung des Fensters ist die neue Fenstermitte.<br>Das Fenster muss so gewählt werden, dass nur ein Wegbrechen des Anschlags zum Ansprechen der Festanschlagsüberwachung führt. |

## Weitere Informationen

### Anstiegsrampe

Über Maschinendatum kann eine Anstiegsrampe für die neue Momentgrenze definiert werden, um ein sprunghaftes Einstellen der Momentgrenze zu vermeiden (z. B. beim Eindrücken einer Pinole).

### Alarmunterdrückung

Bei Anwendungen kann der Anschlagsalarm vom Teileprogramm her unterdrückt werden, indem in einem Maschinendatum der Alarm maskiert und die neue MD-Einstellung mit NEW_CONF wirksam gesetzt wird.

### Aktivierung

Die Befehle zum Fahren auf Festanschlag können aus Synchronaktionen / Technologiezyklen heraus aufgerufen werden. Die Aktivierung kann auch ohne Bewegung erfolgen, das Moment wird sofort begrenzt. Sobald die Achse sollwertseitig bewegt wird, wird auf Anschlag überwacht.

### Aktivieren aus Synchronaktionen

Beispiel:

Wenn das erwartete Ereignis ($R1) eintrifft und Fahren auf Festanschlag nicht schon läuft, soll FXS für Achse Y aktiviert werden. Das Moment soll 10 % des Nennmomentes betragen. Für die Breite des Überwachungsfensters gilt der Vorbesetzungswert.

**Programmcode**

```
N10 IDS=1 WHENEVER (($R1=1) AND ($AA_FXS[Y]==0)) DO $R1=0 FXS[Y]=1
FXST[Y]=10
```

Das normale Teileprogramm muss dafür sorgen, dass $R1 zum gewünschten Zeitpunkt gesetzt wird.

### Deaktivieren aus Synchronaktionen

Beispiel:

Wenn ein erwartetes Ereignis vorliegt ($R3) und der Zustand "Anschlag angefahren" (Systemvariable $AA_FXS) besteht, soll FXS abgewählt werden.

**Programmcode**

```
IDS=4 WHENEVER (($R3==1) AND ($AA_FXS[Y]==1)) DO FXS[Y]=0
FA[Y]=1000 POS[Y]=0
```

### Festanschlag wurde erreicht

Nachdem der Festanschlag erreicht ist:

- Wird der Restweg gelöscht und der Lagesollwert nachgeführt.
- Steigt das Antriebsmoment bis zum programmierten Grenzwert `FXSW` an und bleibt dann konstant.
- Wird die Überwachung des Festanschlags innerhalb der gegebenen Fensterbreite aktiv.

### Randbedingungen

- Messen mit Restweglöschen
  "Messen mit Restweglöschen" (Befehl `MEAS`) und "Fahren auf Festanschlag" können nicht gleichzeitig in einem Satz programmiert werden.
  **Ausnahme**: Eine Funktion wirkt auf eine Bahnachse und die andere auf eine Positionierachse, oder beide wirken auf Positionierachsen.
- Konturüberwachung
  Während "Fahren auf Festanschlag" aktiv ist, erfolgt keine Konturüberwachung.
- Positionierachsen
  Bei "Fahren auf Festanschlag" mit Positionierachsen wird der Satzwechsel unabhängig von der Festanschlagsbewegung durchgeführt.
- Link- und Containerachsen
  Fahren auf Festanschlag ist auch zulässig für Link- und Containerachsen.
  Der Zustand der zugeordneten Maschinenachse bleibt über Container-Drehungen hinweg erhalten. Dies gilt auch für modale Momentenbegrenzung mit FOCON.
- Fahren auf Festanschlag ist **nicht** möglich:
  - Bei Gantry-Achsen
  - Für konkurrierende Positionierachsen, die ausschließlich von der PLC gesteuert werden (die Anwahl von `FXS` muss aus dem NC-Programm erfolgen).
- Wird die Momentengrenze zu weit abgesenkt, kann die Achse der Sollwertvorgabe nicht mehr folgen, der Lageregler geht in die Begrenzung und die Konturabweichung steigt an. In diesem Betriebszustand kann es bei einer Erhöhung der Momentengrenze zu ruckartigen Bewegungen kommen. Um sicherzustellen, dass die Achse noch folgen kann, ist zu kontrollieren, dass die Konturabweichung nicht größer ist als bei unbegrenztem Moment.

## 2.14.7 Verweilzeit (G4)

Mit dem Befehl `G4` wird in einem Satz eine Zeit (Verweilzeit) programmiert, die abläuft, sobald der Satz im Hauptlauf in Bearbeitung ist. Der Satzwechsel in den nachfolgenden Satz erfolgt, sobald die Zeit vollständig abgelaufen ist.

**Hinweis**

`G4` unterbricht den Bahnsteuerbetrieb.

### Syntax

```
G4 F<Time>
G4 S<NumSpi>
G4 S<n> = <NumSpi>
```

### Bedeutung

| | | |
|---|---|---|
| `G4`: | Verweilzeit aktivieren | |
| | alleine im Satz: | ja |
| `F<Time>`: | Unter der Adresse `F` wird die Verweilzeit `<Time>` in Sekunden angegeben. | |
| `S<NumSpi>`: | Unter der Adresse S wird die Verweilzeit in Spindelumdrehungen `<NumSpi>` bezogen auf die aktuelle Hauptspindel angegeben. | |
| `S<n>=NumSpi>`: | Unter der Adresse S wird die Verweilzeit in Spindelumdrehungen `<NumSpi>` bezogen auf die mit der Adresserweiterung `<n>` adressierte Spindel angegeben. | |

**Hinweis**

Die im Verweilzeitsatz `G4` für die Zeitangabe verwendeten Adressen `F` und `S` beeinflussen die Vorschübe `F...` und die Spindeldrehzahlen `S...` des Programms nicht.

### Randbedingungen

#### Synchronaktionen

In einem Programm sind zwei Synchronaktionen so programmiert, dass der nachfolgende Satz mit Verweilzeit zum Aktionssatz wird, in dem die Synchronaktionen ausgeführt werden. Die eine Synchronaktion ist eine modale Synchronaktion. Die andere Synchronaktion ist eine satzweise Synchronaktion. Soll die satzweise Synchronaktion die modale Synchronaktion beeinflussen z.B. durch UNLOCK zur Bearbeitung freigeben, müssen als wirksame Verweilzeit **mindestens zwei Interpolatortakte** z.B. `G4 F<Interpolatortakt * 2>` zur Verfügung stehen.

Die wirksame Verweilzeit ist abhängig von der Einstellung im Maschinendatum MD10280 $MN_PROG_FUNCTION_MASK, Bit 4 = <Wert>

| Wert | Bedeutung |
|---|---|
| 0 | Die wirksame Verweilzeit ist gleich der programmierten Verweilzeit |
| 1 | Die wirksame Verweilzeit ist gleich der programmierten Verweilzeit gerundet auf das nächstgrößere Vielfache des Interpolatortaktes (MD10071 $MN_IPO_CYCLE_TIME) |

Programmbeispiel:

- MD10071 $MN_IPO_CYCLE_TIME == 8 ms
- MD10280 $MN_PROG_FUNCTION_MASK, Bit 4 = 1

```
Programmcode                                Kommentar
N10 WHEN TRUE DO LOCK(1)                    ; Satzweise SynAct: LOCK der
                                            ; modalen SynAct. ID=1
N20 G4 F2                                   ; Aktionssatz für SynAct aus N10
N30 WHEN TRUE DO UNLOCK(1)                  ; Satzweise SynAct: UNLOCK
                                            ; der modalen SynAct. ID=1
N40 ID=1 WHENEVER TRUE DO $R0=1 RDISABLE    ; Modale SynAct ID=1
                                            ; R-Parameter R0=1
                                            ; Setzen der Einlesesperre
N50 G4 F0.012                               ; Aktionssatz für SynAct aus N40 und N50
                                            ; siehe unten Absatz "Beschreibung"
N60 G4 F10
```

Beschreibung

Das gewünschte Verhalten ist, dass die satzweise Synchronaktion aus N30 die aktive Sperre (LOCK) der modalen Synchronaktion mit ID=1 aus N40 aufhebt und dadurch in N50 der R-Parameter geschrieben und die Einlesesperre wirksam wird. Dieses Verhalten wird aber nur erreicht, wenn die wirksame Verweilzeit mindestens zwei Interpolatortakte lang ist.

Die wirksame Verweilzeit ergibt sich aus der programmierten Verweilzeit, dem Interpolatortakt und der Einstellung in MD10280 $MN_PROG_FUNCTION_MASK, Bit 4. Damit die wirksame Verweilzeit mindestens zwei Interpolatortakt lang ist, muss folgende Verweilzeit programmiert werden:

- Bit 4 == 0: Programmierte Verweilzeit ≥ 2 * Interpolatortakt
- Bit 4 == 1: Programmierte Verweilzeit ≥ 1,5 * Interpolatortakt

Ist die wirksame Verweilzeit kleiner als zwei Interpolatortakte, wird das Schreiben des R-Parameters und die Einlesesperre erst im Satz N60 ausgeführt.

## Beispiel

```
Programmcode                Kommentar
N10 G1 F200 Z-5 S300 M3     ; Vorschub F, Spindeldrehzahl S
N20 G4 F3                   ; Verweilzeit: 3s
N30 X40 Y10
```

| Programmcode | Kommentar |
|---|---|
| N40 G4 S30 | ; 30 Umdrehungen der Spindel verweilen (entspricht bei S = 300 U/min und 100% Drehzahloverride: t = 0,1 min). |
| N50 X... | ; In N10 programmierter Vorschub und Spindeldrehzahl wirken weiterhin. |

## 2.14.8 Interner Vorlaufstopp

### Funktion

Beim Zugriff auf Zustandsdaten der Maschine ($A…) erzeugt die Steuerung internen Vorlaufstopp. Der nachfolgende Satz wird erst dann ausgeführt, wenn alle vorher aufbereiteten und gespeicherten Sätze vollständig abgearbeitet sind. Der vorherige Satz wird im Genauhalt (wie G9) angehalten.

### Beispiel

| Programmcode | Kommentar |
|---|---|
| ... | |
| N40 POSA[X]=100 | |
| N50 IF $AA_IM[X]==R100 GOTOF MARKE1 | ; Zugriff auf Zustandsdaten der Maschine ($A…), die Steuerung erzeugt internen Vorlaufstopp. |
| N60 G0 Y100 | |
| N70 WAITP(X) | |
| N80 MARKE1: | |
| ... | |

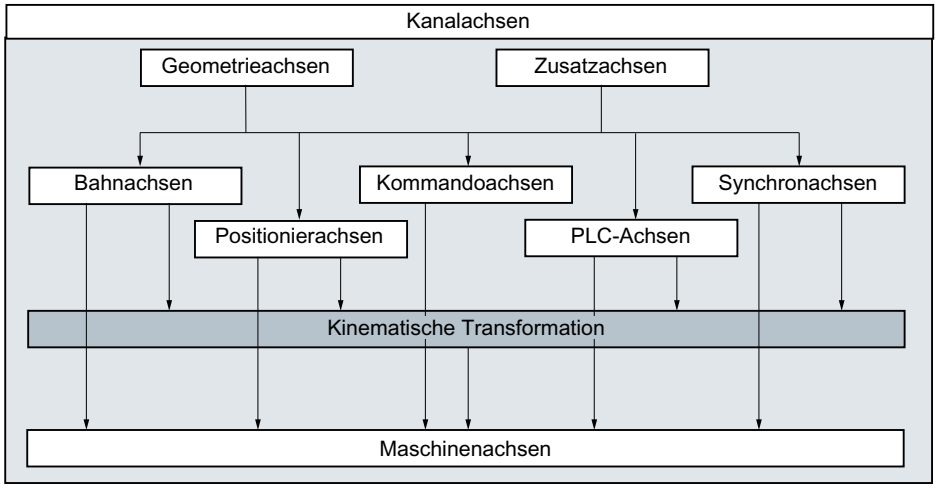
# 2.15 Sonstige Informationen

## 2.15.1 Achsen

### Achstypen

Im Rahmen der Programmierung werden folgende Achstypen unterschieden:

- Hauptachsen / Geometrieachsen
- Zusatzachsen
- Hauptspindel, Masterspindel
- Maschinenachsen
- Kanalachsen
- Bahnachsen
- Positionierachsen
- Synchronachsen
- Kommandoachsen
- PLC-Achsen / Konkurrierende Positionierachsen
- Link-Achsen
- Lead-Linkachsen




### 2.15.1.1 Hauptachsen/Geometrieachsen

Die Hauptachsen bestimmen ein rechtwinkliges, rechtsdrehendes Koordinatensystem. In diesem Koordinatensystem werden Werkzeugbewegungen programmiert.

In der NC-Technik werden die Hauptachsen als Geometrieachsen bezeichnet. Dieser Begriff wird in dieser Programmieranleitung ebenfalls verwendet.

### Umschaltbare Geometrieachsen

Mit der Funktion "Umschaltbare Geometrieachsen" (siehe Funktionshandbuch Arbeitsvorbereitung) lässt sich der über Maschinendatum konfigurierte Geometrieachsverbund vom Teileprogramm aus verändern. Dabei kann eine als synchrone Zusatzachse definierte Kanalachse eine beliebige Geometrieachse ersetzen.

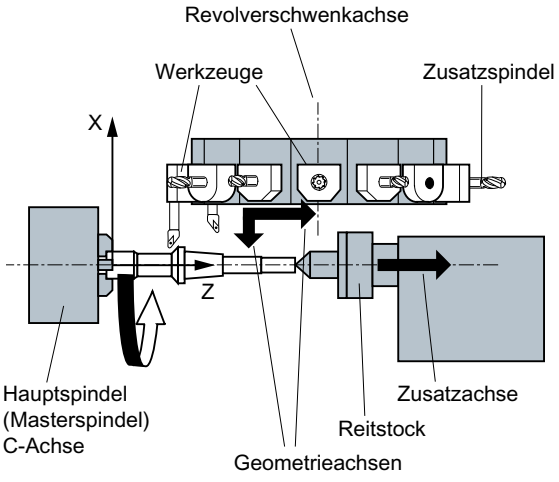
### Achsbezeichner

Der Name/Bezeichner einer Geometrieachse kann über das folgende Maschinendatum festgelegt werden:

MD20060 $MC_AXCONF_GEOAX_NAME_TAB (Geometrieachsname im Kanal)

Standardbezeichnung bei Drehmaschinen:

1. Geometrieachse: X
2. Geometrieachse:  Z




Standardbezeichnung bei Fräsmaschinen:

1. Geometrieachse: X
2. Geometrieachse: Y
3. Geometrieachse: Z

### Weitere Informationen

Maximal drei Geometrieachsen werden zur Programmierung der Frames und der Werkstückgeometrie (Kontur) verwendet.

Die Bezeichner für Geometrie- und Kanalachsen dürfen gleich sein, sofern eine Abbildung möglich ist.

Geometrie- und Kanalachsnamen können in allen Kanälen gleich sein. So kann ein Programm in jedem beliebigen Kanal abgearbeitet werden.

### 2.15.1.2 Zusatzachsen

Im Gegensatz zu den Geometrieachsen ist bei den Zusatzachsen kein geometrischer Zusammenhang zwischen den Achsen definiert.

Typische Zusatzachsen sind:

- Werkzeugrevolverachsen
- Schwenktischachsen
- Schwenkkopfachsen
- Laderachsen

#### Achsbezeichner

Bei einer Drehmaschine mit Revolvermagazin z. B.:

- Revolverposition U
- Reitstock V

#### Programmierbeispiel

| Programmcode | Kommentar |
|---|---|
| `N10 G1 X100 Y20 Z30 A40 F300` | `; Bahnachsbewegungen.` |
| `N20 POS[U]=10POS[X]=20 FA[U]=200 FA[X]=350` | `; Positionierachsbewegungen.` |
| `N30 G1 X500 Y80 POS[U]=150FA[U]=300 F550` | `; Bahn- und Positionierachse.` |
| `N40 G74 X1=0 Z1=0` | `; Referenzpunkt anfahren.` |

### 2.15.1.3 Hauptspindel, Masterspindel

Welche Spindel Hauptspindel ist, wird durch die Maschinenkinematik bestimmt. Diese Spindel wird in der Regel per Maschinendatum als Masterspindel deklariert.

Diese Zuordnung kann durch den Programmbefehl `SETMS(<Spindelnummer>)` geändert werden. Mit `SETMS` ohne Angabe der Spindelnummer kann auf die im Maschinendatum festgelegte Masterspindel zurückgeschaltet werden.

Für die Masterspindel gelten spezielle Funktionen, wie z. B. Gewindeschneiden.

#### Spindelbezeichner

S oder S0

### 2.15.1.4 Maschinenachsen

Maschinenachsen sind die physikalisch an der Maschine vorhandenen Achsen.

Die programmierte Bewegung einer Bahn- oder Zusatzachse kann durch eine im Kanal aktive Transformation (TRANSMIT, TRACYL oder TRAORI) auf mehrere Maschinenachsen wirken.

Maschinenachsen werden nur Sonderfällen direkt im Programm angesprochen (z. B. beim Referenzpunkt- oder Festpunktfahren).

#### Achsbezeichner

Der Name/Bezeichner einer Maschinenachse kann über das folgende NC-spez. Maschinendatum festgelegt werden:

MD10000 $MN_AXCONF_MACHAX_NAME_TAB (Maschinenachsname)

Standardeinstellung: X1, Y1, Z1, A1, B1, C1, U1, V1

Die Maschinenachsen haben darüber hinaus feste Achsbezeichner, die unabhängig vom im Maschinendatum eingestellten Namen immer verwendet werden können:

AX1, AX2, …, AX<n>

### 2.15.1.5 Kanalachsen

Als Kanalachsen werden alle Geometrie-, Zusatz- und Maschinenachsen bezeichnet, die einem Kanal zugeordnet sind.

#### Achsbezeichner

Der kanalspezifische Name/Bezeichner einer Geometrie- und Zusatzachsen kann über das folgende Maschinendatum festgelegt werden:

MD20080 $MC_AXCONF_CHANAX_NAME_TAB (Kanalachsname)

Standardeinstellung: X, Y, Z, A, B, C, U, V

Die Zuordnung, auf welche Maschinenachse eine Geometrie- oder Zusatzachsen im Kanal abgebildet wird, wird in folgendem Maschinendatum festgelegt:

MD20070 $MC_AXCONF_MACHAX_USED (verwendete Maschinenachsen)

### 2.15.1.6 Bahnachsen

Bahnachsen beschreiben den Bahnweg und somit die Werkzeugbewegung im Raum.

Der programmierte Vorschub wirkt entlang dieser Bahn. Die an dieser Bahn beteiligten Achsen erreichen ihre Position gleichzeitig. In der Regel sind das die Geometrieachsen.

Welche Achsen Bahnachsen und damit geschwindigkeitsbestimmend sind, wird jedoch per Voreinstellungen festgelegt.

Im NC-Programm können Bahnachsen mit `FGROUP` angegeben werden.

Mehr Informationen zu `FGROUP` siehe "Vorschub (G93, G94, G95, F, FGROUP, FL, FGREF) (Seite 108)".

### 2.15.1.7 Positionierachsen

Positionierachsen werden getrennt interpoliert, d. h. jede Positionierachse hat einen eigenen Achsinterpolator und einen eigenen Vorschub. Positionierachsen interpolieren nicht mit den Bahnachsen.

Positionierachsen werden aus dem NC-Programm oder von der PLC verfahren. Falls eine Achse gleichzeitig vom NC-Programm und der PLC verfahren werden soll, erscheint eine Fehlermeldung.

Typische Positionierachsen sind:

- Lader für Werkstückantransport
- Lader für Werkstückabtransport
- Werkzeugmagazin/Revolver

### Typen

Zu unterscheiden ist zwischen Positionierachsen mit Synchronisation zum Satzende oder über mehrere Sätze hinweg.

**POS-Achsen**

Der Satzwechsel erfolgt zum Satzende, wenn alle in diesem Satz programmierten Bahn- und Positionierachsen ihren programmierten Endpunkt erreicht haben.

**POSA-Achsen**

Die Bewegungen dieser Positionierachsen können über mehrere Sätze ablaufen.

**POSP-Achsen**

Die Bewegung dieser Positionierachsen zum Anfahren der Endposition erfolgt in Teilstücken.

**Hinweis**

Positionierachsen werden zu Synchronachsen, wenn sie ohne die besondere Kennung POS/POSA verfahren werden.

Ein Bahnsteuerbetrieb (G64) für Bahnachsen ist nur dann möglich, wenn die Positionierachsen (POS) vor den Bahnachsen ihre Endposition erreicht haben.

Bahnachsen, die mit `POS`/`POSA` programmiert werden, werden für diesen Satz aus dem Bahnachsverbund herausgenommen.

Mehr Informationen zu `POS`, `POSA` und `POSP` siehe "Positionierachsen verfahren (POS, POSA, POSP, FA, WAITP, WAITMC) (Seite 116)".

## 2.15.1.8 Synchronachsen

Synchronachsen fahren synchron zum Bahnweg von der Anfangsposition in die programmierte Endposition.

Der unter `F` programmierte Vorschub gilt für alle im Satz programmierten Bahnachsen, jedoch nicht für die Synchronachsen. Synchronachsen benötigen für ihren Weg die gleiche Zeit wie die Bahnachsen.

Eine Synchronachse kann zum Beispiel eine Rundachse sein, die synchron zur Bahninterpolation verfahren wird.

### 2.15.1.9 Kommandoachsen

Kommandoachsen werden aus Synchronaktionen auf Grund eines Ereignisses (Kommandos) gestartet. Sie können vollkommen asynchron zum Teileprogramm positioniert, gestartet und gestoppt werden. Eine Achse kann nicht gleichzeitig aus dem Teileprogramm und aus Synchronaktionen bewegt werden.

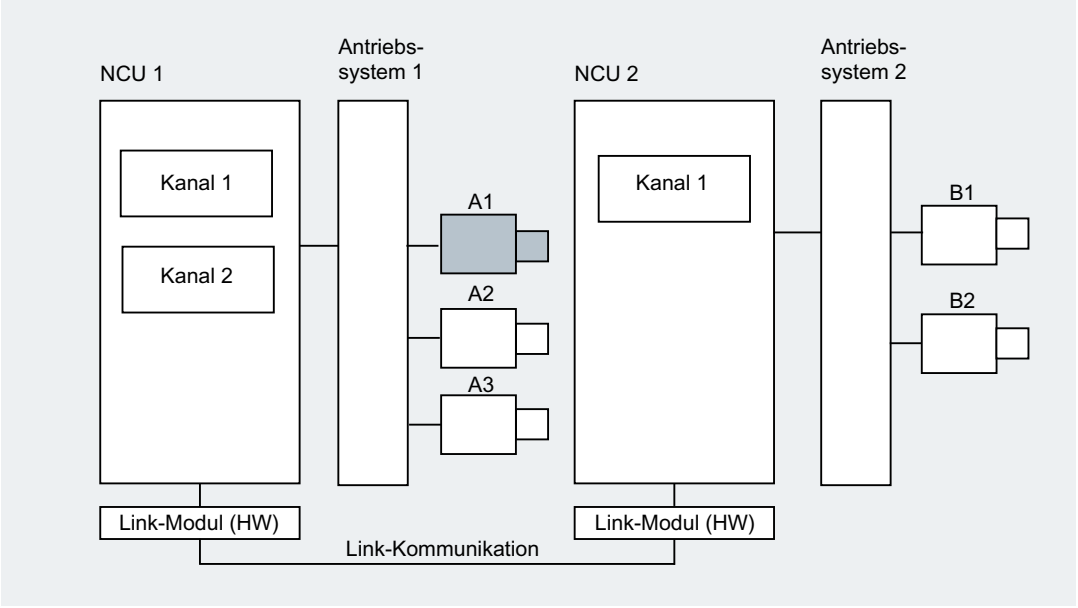
Kommandoachsen werden getrennt interpoliert, d. h. jede Kommandoachse hat einen eigenen Achsinterpolator und einen eigenen Vorschub.

### 2.15.1.10 PLC-Achsen

PLC-Achsen werden von der PLC über spezielle Funktionsbausteine im Grundprogramm verfahren und können sich asynchron zu allen übrigen Achsen bewegen. Die Fahrbewegungen laufen losgelöst von Bahn- und Synchronbewegungen ab.

### 2.15.1.11 Linkachsen

Link-Achsen sind Achsen, die an einer anderen NCU physikalisch angeschlossen sind und deren Lageregelung unterliegen. Link-Achsen können dynamisch Kanälen einer **anderen** NCU zugeordnet werden. Link-Achsen sind aus Sicht einer bestimmten NCU nicht-lokale Achsen.




Der dynamischen Änderung der Zuordnung zu einer NCU dient das Konzept der **Achscontainer**. Achstausch mit `GET` und `RELEASE` aus dem Teileprogramm ist für Link-Achsen **nicht** verfügbar.

## Weitere Informationen

### Voraussetzungen

- Die beteiligten NCUs NCU1 und NCU2 müssen über das Link-Modul mit schneller Link-Kommunikation verbunden sein.
- Die Achse muss durch Maschinendaten entsprechend konfiguriert werden.
- Die Option "Link-Achse" muss vorhanden sein.

### Beschreibung

Die Lageregelung erfolgt auf der NCU, auf der die Achse physikalisch mit dem Antrieb verbunden ist. Dort befindet sich auch die zugehörige Achs-VDI-Schnittstelle. Die Lagesollwerte werden bei Link-Achsen auf einer anderen NCU erzeugt und über NCU-Link kommuniziert.

Die Link-Kommunikation muss für das Zusammenspiel zwischen den Interpolatoren mit dem Lageregler bzw. PLC-Interface sorgen. Die von den Interpolatoren errechneten Sollwerte müssen an den Lageregelkreis auf der Heimat-NCU transportiert werden, bzw. die Istwerte müssen wieder zurücktransportiert werden.

### Achscontainer

Ein Achscontainer ist eine Ringpuffer-Datenstruktur, in der die Zuordnung von lokalen Achsen und/oder Link-Achsen zu Kanälen erfolgt. Die Einträge im Ringpuffer sind **zyklisch verschiebbar**.

Die Link-Achsen Konfiguration lässt im logischen Maschinenachsabbild neben dem direkten Verweis auf lokale Achsen oder Link-Achsen den Verweis auf Achscontainer zu. Ein solcher Verweis besteht aus:

- Container-Nummer **und**
- Slot (Ringpuffer-Platz innerhalb des entsprechenden Containers)

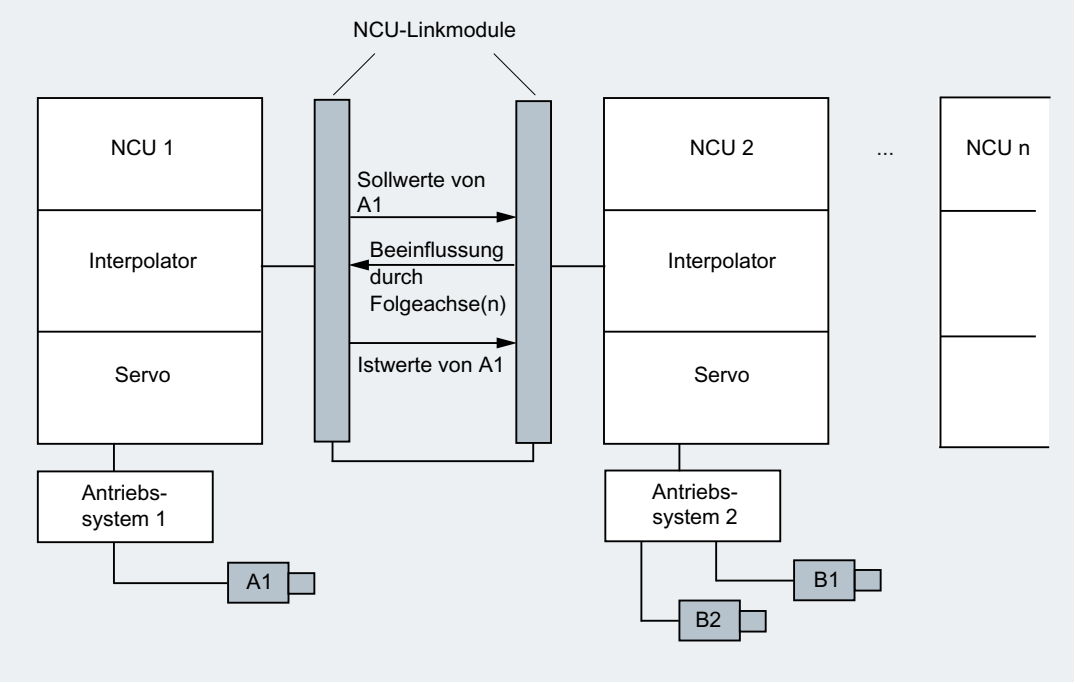
Als Eintrag in einem Ringpuffer-Platz steht:

- eine lokale Achse **oder**
- eine Link-Achse

Achscontainer-Einträge enthalten lokale Maschinenachsen oder Link-Achsen aus der Sicht einer einzelnen NCU. Die Einträge im logischen Maschinenachsabbild (MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB) einer einzelnen NCU sind fest.

### 2.15.1.12 Lead-Linkachsen

Eine Lead-Linkachse ist eine Achse, die von einer NCU interpoliert und einer oder mehreren anderen NCUs als Leitachse für das Führen von Folgeachsen benutzt wird.




Ein axialer Lageregler-Alarm wird an alle weiteren NCUs, die über eine Lead-Linkachse einen Bezug auf die betroffene Achse haben, weiterverteilt.

Die von der Lead-Linkachse abhängigen NCUs können folgende Kopplungen an die Lead-Linkachse benutzen:

- Leitwert (Soll-, Ist-Leitwert, Simulierter Leitwert)
- Mitschleppen
- Tangentiale Nachführung
- Elektronisches Getriebe (ELG)
- Synchronspindel

#### Programmierung

Leit-NCU:

Nur die NCU, der die Leitwert-Achse physikalisch zugeordnet ist, kann Verfahrbewegungen für diese Achse programmieren. Die Programmierung muss darüberhinaus keine Besonderheiten berücksichtigen.

NCUs der Folgeachsen:

Die Programmierung auf der NCU der Folgeachsen darf keine Verfahrbefehle für die Lead-Link-Achse (Leitwert-Achse) enthalten. Verstöße gegen diese Regel lösen einen Alarm aus.

Die Lead-Link-Achse wird über Kanalachs-Bezeichner in gewohnter Weise angesprochen. Die Zustände der Lead-Link-Achse werden durch ausgewählte Systemvariablen zugänglich.

## Weitere Informationen

### Voraussetzungen

- Die beteiligten NCUs NCU1 bis NCU<n> (<n> max. 8) müssen über das Link-Modul mit schneller Link-Kommunikation verbunden sein.
  Literatur:
  Gerätehandbuch Projektierung NCU
- Die Achse muss durch Maschinendaten entsprechend konfiguriert werden.
- Die Option "Link-Achse" muss vorhanden sein.
- Für alle beteiligten NCUs muss der gleiche Interpolatortakt konfiguriert sein.

### Einschränkungen

- Eine Leitachse als Lead-Link-Achse kann nicht Link-Achse sein, d. h. von anderen NCUs als ihrer Heimat-NCU verfahren werden.
- Eine Leitachse als Lead-Link-Achse kann nicht Container-Achse sein, d. h. wechselweise von verschiedenen NCUs angesprochen werden.
- Eine Lead-Link-Achse kann nicht programmierte Führungsachse eines Gantry-Verbandes sein.
- Kopplungen mit Lead-Link-Achsen können nicht mehrstufig hintereinandergeschaltet werden (Kaskadierung).
- Achstausch ist nur innerhalb der Heimat-NCU der Lead-Link-Achse möglich.

### Systemvariablen

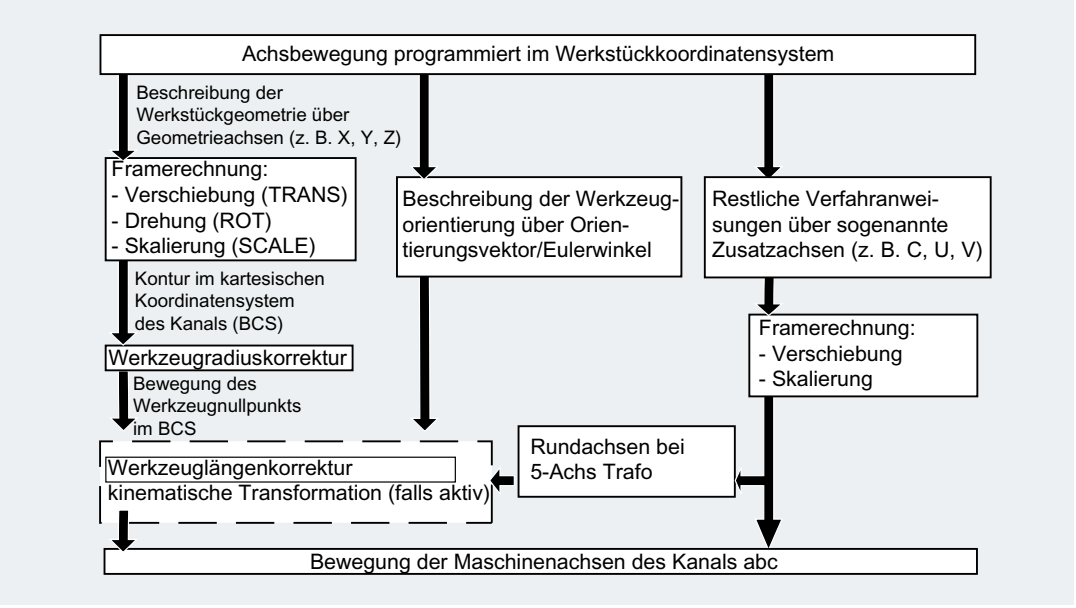
Folgende Systemvariablen können mit dem Kanalachsbezeichner der Lead-Link-Achse benutzt werden:

| Systemvariable | Bedeutung |
|---|---|
| $AA_LEAD_SP | Simulierter Leitwert - Position |
| $AA_LEAD_SV | Simulierter Leitwert - Geschwindigkeit |

Werden diese Systemvariablen durch die NCU der Leitachse aktualisiert, so werden die neuen Werte auch an die NCUs übertragen, die Folgeachsen abhängig von dieser Leitachse verfahren wollen, übertragen.

## 2.15.2 Vom Fahrbefehl zur Maschinenbewegung

Den Zusammenhang zwischen den programmierten Achsbewegungen (Fahrbefehlen) und den daraus resultierenden Maschinenbewegungen soll das folgende Bild veranschaulichen:

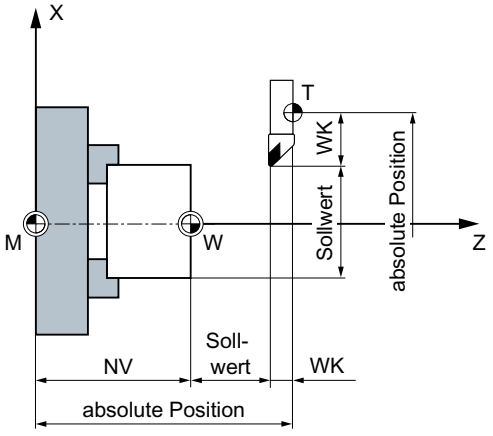



## 2.15.3 Wegberechnung

Die Wegberechnung ermittelt die in einem Satz zu verfahrende Wegstrecke unter Berücksichtigung aller Verschiebungen und Korrekturen.

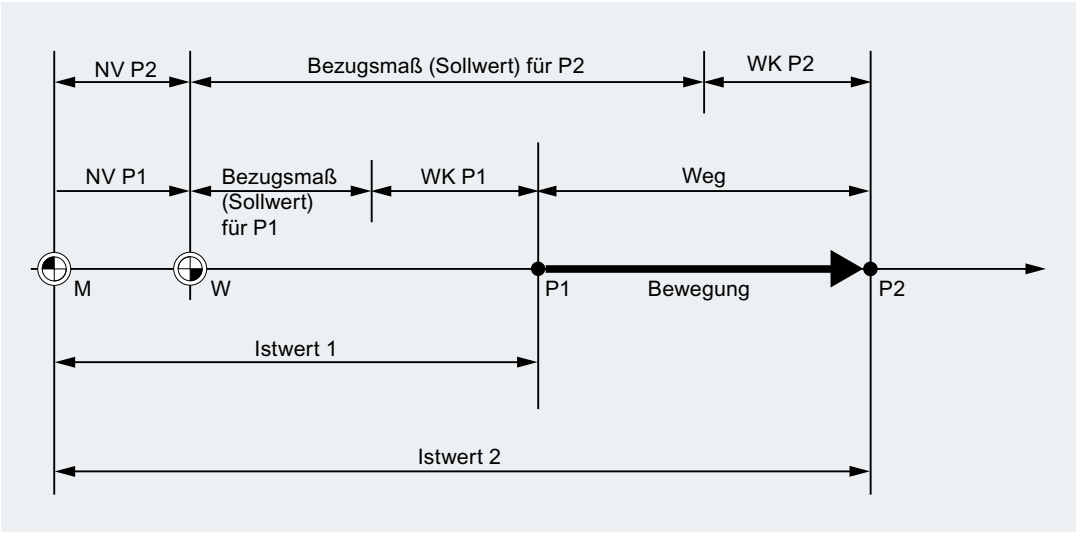
Allgemein gilt:

Weg = Sollwert - Istwert + Nullpunktverschiebung (NV) + Werkzeugkorrektur (WK)



Wird in einem neuen Programmsatz eine neue Nullpunktverschiebung und eine neue Werkzeugkorrektur programmiert, so gilt:

- bei Bezugsmaßeingabe:
  Weg = (Bezugsmaß P2 - Bezugsmaß P1) + (NV P2 - NV P1) + (WK P2 - WK P1).
- bei Kettenmaßeingabe:
  Weg = Kettenmaß + (NV P2 - NV P1) + (WK P2 - WK P1).




## 2.15.4 Adressen

### Feste Adressen

Diese Adressen sind fest eingerichtet, d. h., die Adresszeichen können nicht geändert werden.

Eine Auflistung findet sich in der Tabelle "Feste Adressen (Seite 1256)".

### Einstellbare Adressen

Diesen Adressen kann vom Maschinenhersteller über Maschinendatum ein anderer Name zugeordnet werden.

**Hinweis**

Einstellbare Adressen müssen innerhalb der Steuerung eindeutig sein, d. h., derselbe Adressname darf nicht für unterschiedliche Adresstypen (Achswerte und Endpunkte, Werkzeugorientierung, Interpolationsparameter, ...) verwendet werden.

Eine Auflistung findet sich in der Tabelle "Einstellbare Adressen (Seite 1260)".

## Modal / satzweise wirksame Adressen

Modal wirksame Adressen behalten mit dem programmierten Wert so lange ihre Gültigkeit (in allen Folgesätzen), bis unter der gleichen Adresse ein neuer Wert programmiert wird.

Satzweise wirksame Adressen gelten nur in dem Satz, in dem sie programmiert wurden.

Beispiel:

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 G01 F500 X10` | |
| `N20 X10` | `; Vorschub F aus N10 wirkt so lange, bis ein neuer eingegeben wird.` |

## Adressen mit axialer Erweiterung

Bei Adressen mit axialer Erweiterung steht ein Achsname in eckigen Klammern nach der Adresse, der die Zuordnung zu Achsen festlegt.

Beispiel:

| **Programmcode** | **Kommentar** |
|---|---|
| `FA[U]=400` | `; Achsspezifischer Vorschub für Achse U.` |

Siehe auch Tabelle "Feste Adressen (Seite 1256)".

## Erweiterte Adressschreibweise

Die erweiterte Adressschreibweise bietet die Möglichkeit, eine größere Anzahl von Achsen und Spindeln in eine Systematik einzuordnen.

Eine erweiterte Adresse besteht aus einer numerischen Erweiterung und einem mit "=" Zeichen zugewiesenen arithmetischen Ausdruck. Die numerische Erweiterung ist ein- oder zweistellig und immer positiv.

Die erweiterte Adressschreibweise ist nur für folgende einfache Adressen zulässig:

| Adresse | Bedeutung |
|---|---|
| X, Y, Z, … | Achsadressen |
| I, J, K | Interpolationsparameter |
| S | Spindeldrehzahl |
| SPOS, SPOSA | Spindelposition |
| M | Zusatzfunktionen |
| H | Hilfsfunktionen |
| T | Werkzeugnummer |
| F | Vorschub |

Beispiele:

| **Programmcode** | **Kommentar** |
|---|---|
| X7 | ; kein "=" erforderlich; 7 ist Wert; das Zeichen "=" ist aber auch hier möglich |
| X4=20 | ; Achse X4; "=" ist erforderlich |
| CR=7.3 | ; 2 Buchstaben ; "=" ist erforderlich |
| S1=470 | ; Drehzahl für 1. Spindel: 470 U/min |
| M3=5 | ; Spindel-Halt für 3. Spindel |

Bei den Adressen M, H, S sowie bei SPOS und SPOSA ist die numerische Erweiterung durch eine Variable ersetzbar. Der Variablenbezeichner steht dabei in eckigen Klammern.

Beispiele:

| **Programmcode** | **Kommentar** |
|---|---|
| S[SPINU]=470 | ; Drehzahl für die Spindel, deren Nummer in der Variablen SPINU hinterlegt ist. |
| M[SPINU]=3 | ; Rechtslauf für die Spindel, deren Nummer in der Variablen SPINU hinterlegt ist. |
| T[SPINU]=7 | ; Vorwahl des Werkzeugs für die Spindel, deren Nummer in der Variablen SPINU hinterlegt ist. |

## 2.15.5 Namen

Die Befehle nach DIN 66025 werden durch die NC-Hochsprache unter anderem mit benannten Objekten ergänzt.

Benannte Objekte können z. B. sein:

- Systemvariablen
- Anwenderdefinierte Variablen
- Achsen / Spindeln
- Unterprogramme
- Schlüsselwörter
- Sprungmarken
- Makros

**Hinweis**

Bezeichner müssen eindeutig sein. Derselbe Bezeichner darf **nicht** für verschiedene Objekte verwendet werden.

## Namensregeln

Ein Name kann unter Einhaltung folgender Regeln frei gewählt werden:

- Erlaubte Zeichen:
  - Buchstaben: A ... Z, a ... z
  - Ziffern: 0 ... 9
  - Unterstrich: _
- Die ersten beiden Zeichen sollten Buchstaben oder Unterstriche sein.
- Maximale Länge: :
  - Programmnamen (Seite 43): 24 Zeichen
  - Achsnamen: 8 Zeichen
  - Variablennamen: 31 Zeichen

**Hinweis**

Reservierte Schlüsselwörter dürfen nicht als Bezeichner verwendet werden.

## Zyklen

Zur Vermeidung von Namenskonflikten wird empfohlen folgende Festlegung bei der Vergabe von Namen für Anwender-Zyklen zu beachten:

| Zeichenfolge | reserviert für Namen von |
|---|---|
| • CYCLE<br>• CUST_<br>• GROUP_<br>• _<br>• S_<br>• E_<br>• F_ | SIEMENS-Zyklen |
| • CCS_ | SIEMENS-Compile-Zyklen |
| • CC_ | Anwender-Compile-Zyklen |

### Anwender-Zyklen

Für Namen von Anwender-Zyklen wird empfohlen diesen mit "U_" zu beginnen.

## Variablen

Informationen zur Namensgebung von Variablen finden sich in folgenden Kapiteln:

- Systemdaten (Seite 394)
- Definition von Anwendervariablen (DEF) (Seite 401)

## 2.15.6 Konstanten

### Konstante (allg.)

Eine Konstante ist ein Datenelement, dessen Wert sich bei der Ausführung eines Programms nicht ändert, z. B. eine Wertzuweisung an eine Adresse.

### Dezimal-Konstante

Der Zahlenwert einer Dezimal-Konstanten wird im Dezimalsystem dargestellt.

### INTEGER-Konstante

Eine INTEGER-Konstante ist ein ganzzahliger Wert, d. h. eine Ziffernfolge ohne Dezimalpunkt mit oder ohne Vorzeichen.

Beispiele:

| | |
|---|---|
| `X10` | Zuweisung des Werts +10 an die Adresse X |
| `X-35` | Zuweisung des Werts -35 an die Adresse X |
| `X0` | Zuweisung des Werts 0 an die Adresse X<br>**Hinweis:**<br>X0 kann nicht durch X ersetzt werden. |

### REAL-Konstante

Eine REAL-Konstante ist eine Ziffernfolge mit Dezimalpunkt mit oder ohne Vorzeichen sowie mit oder ohne Exponent.

Beispiele:

| | |
|---|---|
| `X10.25` | Zuweisung des Werts +10.25 an die Adresse X |
| `X-10.25` | Zuweisung des Werts -10.25 an die Adresse X |
| `X0.25` | Zuweisung des Werts +0.25 an die Adresse X |
| `X.25` | Zuweisung des Werts +0.25 an die Adresse X, ohne führende "0" |
| `X=-.1EX-3` | Zuweisung des Werts $-0.1*10^{-3}$ an die Adresse X |

**Hinweis**

Werden bei einer Adresse mit zulässiger Dezimalpunkteingabe nach dem Dezimalpunkt mehr Stellen geschrieben, als für diese Adresse vorgesehen sind, so wird sie auf die vorgesehene Stellenanzahl gerundet.

### Hexadezimal-Konstante

Möglich sind auch Konstanten, die hexadezimal, d. h. in der Basis 16, interpretiert werden. Dabei gelten die Buchstaben A bis F als hexadezimale Ziffern mit den dezimalen Werten 10 bis 15.

Hexadezimale Konstanten werden zwischen Hochkommata gesetzt und beginnen mit dem Buchstaben "H", gefolgt von dem hexadezimal geschriebenen Wert. Trennzeichen zwischen den Buchstaben und Ziffern sind erlaubt.

Beispiel:

| Programmcode | Kommentar |
|---|---|
| `$MC_TOOL_MANAGEMENT_MASK='H7F'` | `; Durch Zuweisung der Hexadezimal-Konstante werden in dem Maschinendatum Bit 0-7 gesetzt.` |

**Hinweis**

Die maximale Zeichenzahl ist durch den Wertebereich des ganzzahligen Datentyps begrenzt.

### Binär-Konstante

Möglich sind auch Konstanten, die binär interpretiert werden. Dabei werden nur die Ziffern "0" und "1" verwendet.

Binäre Konstanten werden zwischen Hochkommata gesetzt und beginnen mit dem Buchstaben "B", gefolgt von dem binär geschriebenen Wert. Trennzeichen zwischen den Ziffern sind erlaubt.

Beispiel:

| Programmcode | Kommentar |
|---|---|
| `$MN_AUXFU_GROUP_SPEC='B10000001'` | `; Durch Zuweisung der Binärkonstante werden in dem Maschinendatum Bit 0 und Bit 7 gesetzt.` |

**Hinweis**

Die maximale Zeichenzahl ist durch den Wertebereich des ganzzahligen Datentyps begrenzt.

## 2.15.7 Operatoren und Rechenfunktionen

### Operatoren

#### Rechenoperatoren

Systemvariable vom Typ REAL oder INT können durch folgende Operatoren miteinander verknüpft werden:

| Operator | Bedeutung |
|---|---|
| + | Addition |
| - | Subtraktion |
| * | Multiplikation |

| Operator | Bedeutung |
|---|---|
| / | • Division in Synchronaktionen: INT / INT ⇒ **INT**<br>• Division in Synchronaktionen mit REAL-Ergebnis durch Verwendung der Funktion ITOR():<br>ITOR( INT ) / ITOR( INT ) ⇒ **REAL**<br>• Division in NC-Programmen: INT / INT ⇒ **REAL** |
| DIV | Integer-Division: INT / INT ⇒ **INT** |
| MOD | Modulo-Division, (nur für Typ INT) liefert Rest einer INT-Division<br>Beispiel: 3 MOD 4 = 3 |

**Hinweis**

Es können nur Variablen gleichen Typs verknüpft werden.

### Vergleichsoperatoren

| Operator | Bedeutung |
|---|---|
| == | gleich |
| > | ungleich |
| < | kleiner |
| > | größer |
| <= | kleiner oder gleich |
| >= | größer oder gleich |

### Boole'sche Operatoren

| Operator | Bedeutung |
|---|---|
| NOT | NICHT |
| AND | UND |
| OR | ODER |
| XOR | Exclusiv-ODER |

### Bitweise logische Operatoren

| Operator | Bedeutung |
|---|---|
| B_OR | bitweise ODER |
| B_AND | bitweise UND |
| B_XOR | bitweise exklusives ODER |
| B_NOT | bitweise Negierung |

## Priorität der Operatoren

Die Operatoren haben bei der Abarbeitung in der Synchronaktion folgende Prioritäten (höchste Priorität: 1):

| Prio. | Operatoren | Bedeutung |
|---|---|---|
| 1 | NOT, B_NOT | Verneinung, bitweise Verneinung |
| 2 | *, /, DIV, MOD | Multiplikation, Division |
| 3 | +, - | Addition, Subtraktion |
| 4 | B_AND | bitweise UND |
| 5 | B_XOR | bitweise exklusives ODER |
| 6 | B_OR | bitweise ODER |
| 7 | AND | UND |
| 8 | XOR | exklusives ODER |
| 9 | OR | ODER |
| 10 | << | Verkettung von Strings, Ergebnistyp STRING |
| 11 | ==, <>, <, >, >=, <= | Vergleichsoperatoren |

### Hinweis

Es wird dringend empfohlen, bei der Verwendung von mehreren Operatoren in einem Ausdruck, die einzelnen Operatoren durch das Setzen von Klammern "( … )" eindeutig zu priorisieren.

Beispiel einer Bedingung mit einem Ausdruck mit mehreren Operatoren:

**Programmcode**

```
... WHEN ($AA_IM[X] > WERT) AND ($AA_IM[Y] > WERT1) DO ...
```

### Rechenfunktionen

| Operator | Bedeutung |
|---|---|
| SIN() | Sinus |
| COS() | Cosinus |
| TAN() | Tangens |
| ASIN() | Arcussinus |
| ACOS() | Arcuscosinus |
| ATAN2(,) | Arcustangens2 |
| SQRT() | Quadratwurzel |
| ABS() | Betrag |
| POT() | 2. Potenz (Quadrat) |
| TRUNC() | ganzzahliger Teil<br>Genauigkeiten bei Vergleichsbefehlen einstellbar mit TRUNC |
| ROUND() | Runden auf eine ganze Zahl |
| LN() | natürlicher Logarithmus |
| EXP() | Exponentialfunktion |

## Indizierung

Der Index einer Systemvariablen vom Typ "Feld von …" kann wiederum eine Systemvariable sein. Der Index wird dabei ebenfalls im Hauptlauf im Interpolatortakt ausgewertet.

Beispiel

**Programmcode**

```
... WHEN ... DO $AC_PARAM[ $AC_MARKER[1] ] = 3
```

### Einschränkungen

- Eine Schachtelung der Indizierung mit weiteren Systemvariablen ist nicht erlaubt.
- Der Index darf nicht über Vorlaufvariablen gebildet werden. Das folgende Beispiel ist somit **nicht** erlaubt, da $P_EP eine Vorlaufvariable ist:
  $AC_PARAM[ 1 ] = $P_EP[ $AC_MARKER[ 0 ] ]

## Siehe auch

Rechenfunktionen (Seite 445)

Operationen (Seite 445)

# Arbeitsvorbereitung 3

## 3.1 Flexible NC-Programmierung

### 3.1.1 Variablen

Durch die Verwendung von Variablen aus den Bereichen Systemdaten und Anwenderdaten, insbesondere in Verbindung mit Rechenfunktionen und Kontrollstrukturen, können NC-Programme und Zyklen flexibel gestaltet werden.

**⚠ WARNUNG**

**Sach- und Personenschäden durch veränderte Variablen**

Bei der Verwendung von Variablen im NC-Programm ist zu berücksichtigen, dass Maschinenbediener oder Unbefugte bei entsprechendem Zugriffsrecht die Variablen verändern und dadurch den Programmablauf beeinflussen können. Sach- und Personenschäden können die Folge sein.

- Um negative Auswirkungen auf den Programmablauf durch die Veränderung von Variablen zu vermeiden, sind im NC-Programm entsprechende Datenüberprüfungen ("Input validation") vorzusehen.

- Systemdaten
  In den Systemdaten liegen die im System vordefinierten Variablen. Diese Variablen haben eine definierte Bedeutung. Sie werden in erster Linie von der Systemsoftware verwendet. Vom Anwender können diese Variablen in NC-Programmen und Zyklen gelesen und geschrieben werden. Beispiel: Maschinendaten, Settingdaten, Systemvariablen.
  Obwohl die Bedeutung eines Systemdatums fest vorgegeben ist, können die Eigenschaften vom Anwender durch Redefinition in gewissem Umfang verändert werden.
  Siehe "Redefinition von Systemdaten, Anwenderdaten und NC-Sprachbefehlen (REDEF) (Seite 407)"
- Anwenderdaten
  In den Anwenderdaten liegen die vom Anwender definierten Variablen, deren Bedeutung ausschließlich durch den Anwender festgelegt wird. Sie werden vom System nicht ausgewertet.
  Die Anwenderdaten sind unterteilt in:
  - Vordefinierte Anwendervariablen
    Vordefinierte Anwendervariablen sind im System bereits definierte Variablen, deren Anzahl über Maschinendaten parametriert wird. Die Eigenschaften dieser Variablen können vom Anwender angepasst werden.
    Siehe "Redefinition von Systemdaten, Anwenderdaten und NC-Sprachbefehlen (REDEF) (Seite 407)".
  - Anwenderdefinierte Variablen
    Anwenderdefinierte Variablen sind Variablen, die vom Anwender definiert und vom System zur Laufzeit angelegt werden. Ihre Anzahl, Datentyp, Sichtbarkeit und alle weiteren Eigenschaften werden ausschließlich durch den Anwender festgelegt.
    Siehe "Definition von Anwendervariablen (DEF) (Seite 401)"

#### 3.1.1.1 Systemdaten

Die Systemdaten enthalten die im System vordefinierten Variablen, die in NC_Programmen und Zyklen Zugriff auf die aktuelle Parametrierung der Steuerung sowie auf Maschinen-, Steuerungs- und Prozesszustände ermöglichen.

##### Vorlaufvariablen

Vorlaufvariablen sind Systemdaten, die im Kontext des Vorlaufs, d. h. zum Zeitpunkt der Interpretation des Satzes, in dem die Variable programmiert ist, gelesen und geschrieben werden. Vorlaufvariable lösen keinen Vorlaufstopp aus.

##### Hauptlaufvariablen

Hauptlaufvariablen sind Systemdaten, die im Kontext des Hauptlaufs, d. h. zum Zeitpunkt der Ausführung des Satzes, in dem die Variable programmiert ist, gelesen oder geschrieben werden. Hauptlaufvariablen sind:

- Variable, die in Synchronaktionen programmiert werden können (Lesen/Schreiben)
- Variable, die im NC_Programm programmiert werden können und Vorlaufstopp auslösen (Lesen/Schreiben)
- Variable, die im NC_Programm programmiert werden können und deren Wert im Vorlauf ermittelt, aber erst im Hauptlauf geschrieben wird (Hauptlauf-synchron: nur Schreiben)

## Präfix-Systematik

Zur besonderen Kennzeichnung von Systemdaten ist dem Namen im Normalfall ein Präfix vorangestellt, der sich aus dem $-Zeichen, gefolgt von einem oder zwei Buchstaben und einem Unterstrich, zusammensetzt:

| **$ + 1. Buchstabe** | **Bedeutung: Datenart** |
|---|---|
| **Vorlaufdaten** (Systemdaten, die im Vorlauf gelesen / geschrieben werden) | |
| $M | Maschinendaten [1] |
| $S | Settingdaten, Schutzbereiche [1] |
| $T | Werkzeugverwaltungsdaten |
| $P | Programmierte Werte |
| $C | Zyklenvariablen der ISO-Hüllzyklen |
| $O | Optionsdaten |
| R | R-Parameter (Rechenparameter) [2] |
| **Hauptlaufdaten** (Systemdaten, die im Hauptlauf gelesen / geschrieben werden) | |
| $$M | Maschinendaten [1] |
| $$S | Settingdaten [1] |
| $A | Aktuelle Hauptlaufdaten |
| $V | Lageregler-Daten |
| $R | R-Parameter (Rechenparameter) [2] |
| [1] Ob Maschinen- und Settingdaten als Vor- oder Hauptlaufvariablen behandelt werden, hängt davon ab, ob sie mit einem oder zwei $-Zeichen geschrieben werden. Die Schreibweise ist anwendungsspezifisch frei wählbar.<br>[2] Bei der Verwendung eines R-Parameters im Teileprogramm / Zyklus als Vorlaufvariable wird kein Präfix geschrieben, z. B. R10. Bei der Verwendung in einer Synchronaktion als Hauptlaufvariable wird als Präfix ein $-Zeichen geschrieben, z. B. $R10. | |

| **2. Buchstabe** | **Bedeutung: Sichtbarkeit** |
|---|---|
| N | NC-globale Variable (**N**C) |
| C | kanalspezifische Variable (**C**hannel) |
| A | achsspezifische Variable (**A**xis) |

## Randbedingungen

### Ausnahmen in der Präfix-Systematik

Folgende Systemvariablen weichen von der oben genannten Präfix-Systematik ab:

- $TC_...: Der 2. Buchstabe C verweist hier nicht auf kanalspezifische, sondern auf Werkzeughalter-spezifische Systemvariablen (TC = Tool Carrier)
- $P_ ...: Kanalspezifische Systemvariablen

### Verwendung von Maschinen- und Settingdaten in Synchronaktionen

Bei der Verwendung von Maschinen- und Settingdaten in Synchronaktionen kann durch den Präfix bestimmt werden, ob das Maschinen- oder Settingdatum vorlauf- oder hauptlaufsynchron gelesen/geschrieben wird.

Bleibt das Datum während der Bearbeitung unverändert, kann vorlaufsynchron gelesen werden. Der Präfix des Maschinen- oder Settingdatums wird dazu mit einem $-Zeichen geschrieben:

```
ID=1 WHENEVER $AA_IM[z] < $SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

Wird das Datum während der Bearbeitung verändert, muss hauptlaufsynchron gelesen / geschrieben werden. Der Präfix des Maschinen- oder Settingdatums wird dazu mit zwei $-Zeichen geschrieben:

```
ID=1 WHENEVER $AA_IM[z] < $$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

**Hinweis**

**Schreiben von Maschinen- und Settingdaten**

Beim Schreiben eines Maschinen- oder Settingdatums ist darauf zu achten, dass die aktive Zugriffsstufe beim Ausführen des Teileprogramms / Zyklus den Schreibzugriff erlaubt und die Wirksamkeit des Datums "IMMEDIATE" ist.

## Siehe auch

Variablen (Seite 393)

### 3.1.1.2 Vordefinierte Anwendervariablen: Kanalspezifische Rechenparameter (R)

Kanalspezifische Rechenparameter oder R-Parameter sind vordefinierte Anwendervariablen mit der Bezeichnung R, definiert als Feld vom Datentyp REAL. Aus historischen Gründen ist für R-Parameter neben der Schreibweise mit Feldindex, z. B. `R[10]`, auch die Schreibweise ohne Feldindex, z. B. `R10`, erlaubt.

Bei der Verwendung in Synchronaktionen muss der Buchstabe $ vorangestellt werden, z. B. `$R10`.

## Syntax

Bei Verwendung als Vorlaufvariable:

```
R<n>
R[<Ausdruck>]
```

Bei Verwendung als Hauptlaufvariable:

```
$R<n>
$R[<Ausdruck>]
```

## Bedeutung

| | |
|---|---|
| `R`: | Bezeichner bei Verwendung als Vorlaufvariable, z. B. im Teileprogramm |
| `$R`: | Bezeichner bei Verwendung als Hauptlaufvariable, z. B. in Synchronaktionen |

| | | |
|---|---|---|
| | Typ: | REAL |
| | Wertebereich: | Bei nicht-exponentieller Schreibweise:<br>± (0.000 0001 ... 9999 9999)<br>**Hinweis:**<br>Es sind maximal 8 Dezimalstellen erlaubt |
| | | Bei exponentieller Schreibweise:<br>± ($1*10^{-300}$ ... $1*10^{+300}$)<br>**Hinweis:**<br>• Schreibweise: <Mantisse>EX<Exponent> z. B. 8.2EX-3<br>• Es sind maximal 10 Zeichen einschließlich Vorzeichen und Dezimalpunkt erlaubt. |
| <n>: | Nummer des R-Parameters | |
| | Typ: | INT |
| | Wertebereich: | 0 - MAX_INDEX<br>**Hinweis**<br>MAX_INDEX ergibt sich aus der parametrierten Anzahl an R-Parametern:<br>MAX_INDEX = (MD28050 $MN_MM_NUM_R_PARAM) - 1 |
| <Ausdruck>: | Feldindex<br>Als Feldindex kann ein beliebiger Ausdruck angegeben werden, solange das Ergebnis des Ausdrucks in den Datentyp INT gewandelt werden kann (INT, REAL, BOOL, CHAR) | |

## Beispiel

Zuweisungen an R-Parameter und Verwendung von R-Parametern in mathematischen Funktionen:

```
Programmcode                Kommentar
R0=3.5678                   ; Zuweisung im Vorlauf
R[1]=-37.3                  ; Zuweisung im Vorlauf
R3=-7                       ; Zuweisung im Vorlauf
$R4=-0.1EX-5                ; Zuweisung im Hauptlauf: R4 = -0.1 * 10^-5
$R[6]=1.874EX8              ; Zuweisung im Hauptlauf: R6 = 1.874 * 10^8
R7=SIN(25.3)                ; Zuweisung im Vorlauf

R[R2]=R10                   ; Indirekte Adressierung über R-Parameter
R[(R1+R2)*R3]=5             ; Indirekte Adressierung über math. Ausdruck

X=(R1+R2)                   ; Verfahre Achse X auf die Position die sich aus der Summe
                            von R1 und R2 ergibt
Z=SQRT(R1*R1+R2*R2)         ; Verfahre Achse Z auf Position Quadratwurzel(R1^2 + R2^2)
```

## Siehe auch

Variablen (Seite 393)

### 3.1.1.3 Vordefinierte Anwendervariablen: Globale Rechenparameter (RG)

## Funktion

Neben den kanalspezifischen R-Parametern stehen dem Anwender auch globale R-Parameter zur Verfügung. Sie existieren innerhalb der Steuerung einmal und können von allen Kanälen aus gelesen/geschrieben werden.

Globale R-Parameter werden z. B. verwendet, um Informationen von einem Kanal in den nächsten zu bekommen. Ein anderes Beispiel sind globale Einstellungen, die für alle Kanäle ausgewertet werden sollen, wie z. B. die Auskragung des Rohteils aus der Spindel.

Lesen und Schreiben der globalen R-Parameter erfolgt über die Bedienoberfäche oder im NC-Programm im Vorlauf. Eine Verwendung in Synchronaktionen oder Technologiezyklen ist nicht möglich.

**Hinweis**

Beim Lesen und Schreiben von globalen R-Parametern erfolgt **keine** Synchronisation zwischen den Kanälen.

Da das Lesen und Schreiben im Vorlauf erfolgt, ist der Zeitpunkt, zu dem ein geschriebener Wert von einem Kanal aus in einem anderen Kanal wirksam wird, nicht definiert.

Beispiel:

Im Kanal1 läuft eine Schleife mit einem globalen R-Parameter als Schleifenzähler. Kanal 2 schreibt einen Wert in diesen globalen R-Parameter, der zu einem Schleifenabbruch in Kanal 1 führt. Alle bis zu diesem Zeitpunkt im Kanal1 im Vorlauf interpretierten Schleifen werden jedoch noch ausgeführt. Wie viele Schleifen das sind ist nicht definiert und hängt unter anderem von der Auslastung des Kanals ab.

Eine Synchronisation zwischen den Kanälen muss der Anwender applikativ, z. B. mit WAIT-Marken, selbst realisieren!

## Syntax

**Schreiben im NC-Programm**

```
RG[<n>]=<Wert>
RG[<Ausdruck>]=<Wert>
```

**Lesen im NC-Programm**

```
R...=RG[<n>]
R...=RG[<Ausdruck>]
```

## Bedeutung

| | | |
|---|---|---|
| RG: | Default-Name der NC-Adresse für **globale** R-Parameter<br>**Hinweis:**<br>Der Name der NC-Adresse ist einstellbar über MD15800 $MN_R_PARAM_NCK_NAME | |
| <n>: | Nummer des globalen R-Parameters | |
| | Typ: | INT |
| | Wertebereich: | 0 ... MAX_INDEX<br>**Hinweis**<br>MAX_INDEX ergibt sich aus der parametrierten Anzahl an globalen R-Parametern:<br>MAX_INDEX = (MD18156 $MN_MM_NUM_R_PARAM_NCK) - 1 |
| <Ausdruck>: | Als Feldindex kann ein beliebiger Ausdruck angegeben werden, solange das Ergebnis des Ausdrucks in den Datentyp INT gewandelt werden kann (INT, REAL, BOOL, CHAR) | |
| <Wert>: | Wert des globalen R-Parameters | |
| | Typ: | REAL |
| | Wertebereich: | Bei nicht-exponentieller Schreibweise:<br>± (0.000 0001 ... 9999 9999)<br>**Hinweis:**<br>Es sind maximal 8 Dezimalstellen erlaubt |
| | | Bei exponentieller Schreibweise:<br>± ($1*10^{-300}$ ... $1*10^{+300}$)<br>**Hinweis:**<br>• Schreibweise: <Mantisse>EX<exponent> z.B. 8.2EX-3<br>• Es sind maximal 10 Zeichen einschließlich Vorzeichen und Dezimalpunkt erlaubt. |

### 3.1.1.4 Vordefinierte Anwendervariablen: Link-Variablen

Über Link-Variablen können im Rahmen der Funktion "NCU-Link" zyklisch Daten zwischen NCUs, die in einem Netzwerk miteinander verbunden sind, ausgetauscht werden. Sie ermöglichen dabei einen Datenformat-spezifischen Zugriff auf den Link-Variablen-Speicher. Der Link-Variablen-Speicher wird sowohl bezüglich der Größe und als auch der Datenstruktur vom Anwender / Maschinenhersteller anlagenspezifisch festgelegt.

Link-Variablen sind systemglobale Anwendervariablen, die bei projektierter Link-Kommunikation von allen NCUs des Link-Verbundes in Teileprogrammen und Zyklen gelesen und geschrieben werden können. Im Gegensatz zu globalen Anwendervariablen (GUD) können Link-Variablen auch in Synchronaktionen verwendet werden.

Bei Anlagen ohne aktiven NCU-Link können Link-Variablen Steuerungs-lokal neben den globalen Anwendervariablen (GUD) als zusätzliche globale Anwendervariablen verwendet werden.

## Syntax

```
$A_DLB[<Index>]
```

```
$A_DLW[<Index>]
$A_DLD[<Index>]
$A_DLR[<Index>]
```

## Bedeutung

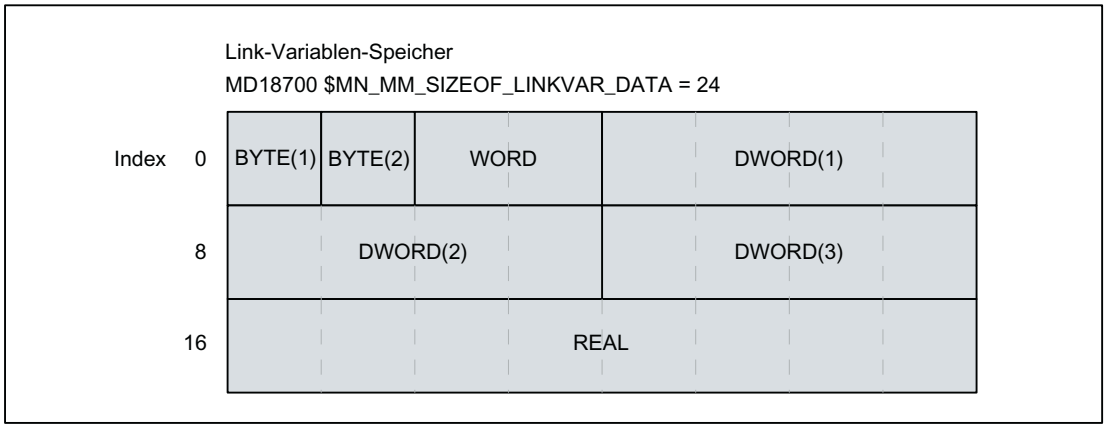
| | | |
|---|---|---|
| $A_DLB: | Link-Variable für Datenformat BYTE (1 Byte) | |
| | Datentyp: | UINT |
| | Wertebereich: | 0 ... 255 |
| $A_DLW: | Link-Variable für Datenformat WORD (2 Bytes) | |
| | Datentyp: | INT |
| | Wertebereich: | -32768 ... 32767 |
| $A_DLD: | Link-Variable für Datenformat DWORD (4 Bytes) | |
| | Datentyp: | INT |
| | Wertebereich: | -2147483648 ... 2147483647 |
| $A_DLR: | Link-Variable für Datenformat REAL (8 Bytes) | |
| | Datentyp: | REAL |
| | Wertebereich: | $\pm(2{,}2*10^{-308} \ldots 1{,}8*10^{+308})$ |
| <Index>: | Adressindex in Byte, gerechnet vom Anfang des Link-Variablen-Speichers | |
| | Datentyp: | INT |
| | Wertebereich: | 0 - MAX_INDEX<br>**Hinweis**<br>• MAX_INDEX ergibt sich aus der parametrierten Größe des Link-Variablen-Speichers: MAX_INDEX = (MD18700 $MN_MM_SIZEOF_LINKVAR_DATA) - 1<br>• Es dürfen nur Indizes programmiert werden, so dass die im Link-Variablen-Speicher adressierten Bytes auf einer Datenformatgrenze liegen ⇒<br>Index = n * Bytes, mit n = 0, 1, 2, ...<br>– $A_DLB[i]: i = 0, 1, 2, ...<br>– $A_DLW[i]: i = 0, 2, 4, ...<br>– $A_DLD[i]: i = 0, 4, 8, ...<br>– $A_DLR[i]: i = 0, 8, 16, ... |

## Beispiel

In der Automatisierungsanlage sind 2 NCUs (NCU1 und NCU2) vorhanden. An NCU1 ist Maschinenachse AX2 angeschlossen, die als Link-Achse von NCU2 verfahren wird.

NCU1 schreibt zyklisch den Stromistwert ($VA_CURR) der Achse AX2 in den Link-Variablen-Speicher. NCU2 liest zyklisch den per Link-Kommunikation übertragenen Stromistwert und zeigt bei Überschreitung des Grenzwertes Alarm 61000 an.

Die Datenstruktur im Link-Variablen-Speicher ist im folgenden Bild dargestellt. Der Stromistwert wird über den REAL-Wert übertragen.




### NCU1

NCU1 schreibt in einer statischen Synchronaktion zyklisch im IPO-Takt den Stromistwert der Achse AX2 über die Link-Variable $A_DLR[ 16 ] in den Link-Variablen-Speicher.

**Programmcode**

```
N111 IDS=1 WHENEVER TRUE DO $A_DLR[16]=$VA_CURR[AX2]
```

### NCU2

NCU2 liest in einer statischen Synchronaktion zyklisch im IPO-Takt den Stromistwert der Achse AX2 über die Link-Variable $A_DLR[ 16 ] aus dem Link-Variablen-Speicher. Ist der Stromistwert größer als 23.0 A, wird der Alarm 61000 angezeigt.

**Programmcode**

```
N222 IDS=1 WHEN $A_DLR[16] > 23.0 DO SETAL(61000)
```

## 3.1.1.5 Definition von Anwendervariablen (DEF)

Mit dem Befehl `DEF` können anwenderspezifische Variablen, oder Anwendervariablen (User Data), definiert und mit Werten belegt werden.

Entsprechend dem Gültigkeitsbereich, d. h. dem Bereich in dem die Variable sichtbar ist, gibt es folgende Kategorien von Anwendervariablen:

- Lokale Anwendervariablen (LUD)
  Lokale Anwendervariablen (LUD) sind Variablen, die in einem NC-Programm definiert sind, das zum Zeitpunkt der Abarbeitung nicht das Hauptprogramm ist. Sie werden beim Aufruf des NC-Programms angelegt und mit Programmende-Reset bzw. dem nächsten Steuerungshochlauf gelöscht. Auf LUD kann nur innerhalb des NC-Programms zugegriffen werden, in dem sie definiert sind.
- Programmglobale Anwendervariablen (PUD)
  Programmglobale Anwendervariablen (PUD) sind Variablen, die in einem als Hauptprogramm verwendeten NC-Programm definiert sind. Sie werden beim Aufruf des NC-Programms angelegt und mit Programmende-Reset bzw. dem nächsten Steuerungshochlauf gelöscht. Auf PUD kann im Hauptprogramm und in allen Unterprogrammen zugegriffen werden.

  **Hinweis**

  **Verfügbarkeit von programmglobalen Anwendervariablen (PUD)**

  Im Hauptprogramm definierte programmglobale Anwendervariablen (PUD) sind nur dann auch in den Unterprogrammen verfügbar, wenn folgendes Maschinendatum gesetzt ist:

  MD11120 $MN_LUD_EXTENDED_SCOPE = 1

  Mit MD11120 = 0 sind die im Hauptprogramm definierten programmglobalen Anwendervariablen nur im Hauptprogramm verfügbar.

- Globale Anwendervariablen (GUD)
  Globale Anwendervariablen (GUD) sind NC- bzw. Kanal-globale Variablen, die in einem Datenbaustein (SGUD, MGUD, UGUD, GUD4 ... GUD9) definiert sind und auch nach Programmende-Reset bzw. dem nächsten Steuerungshochlauf erhalten bleiben. Auf GUD kann in allen NC-Programmen zugegriffen werden.

Anwendervariablen müssen vor ihrer Verwendung (Lesen / Schreiben) definiert worden sein. Folgende Regeln sind dabei zu beachten:

- GUD müssen in einer Definitionsdatei, z. B. _N_DEF_DIR/_N_UGUD_DEF, definiert werden.
- PUD und LUD müssen im Definitionsteil eines NC-Programms definiert werden.
- Die Datendefinition muss in einem eigenen Satz erfolgen.
- Pro Datendefinition darf nur ein Datentyp verwendet werden.
- Pro Datendefinition können mehrere Variable des gleichen Datentyps definiert werden.

## Syntax

**LUD und PUD**

```
DEF <Typ> <Phys_Einheit> <Grenzwerte> <Name>[<Wert_1>, <Wert_2>,
<Wert_3>]=<Init_Wert>
```

**GUD**

```
DEF <Bereich> <VL_Stop> <Zugriffsrechte> <Datenklasse> <Typ>
<Phys_Einheit> <Grenzwerte> <Name>[<Wert_1>, <Wert_2>,
<Wert_3>]=<Init_Wert>
```

## Bedeutung

| | | |
|---|---|---|
| DEF: | Befehl zur Definition von Anwendervariablen GUD, PUD, LUD | |
| <Bereich>: | Gültigkeitsbereich, nur relevant für GUD: | |
| | NC: | NC-globale Anwendervariable |
| | CHAN: | Kanal-globale Anwendervariable |
| <VL_Stop>: | Vorlaufstop, nur relevant für GUD (optional) | |
| | SYNR: | Vorlaufstop beim Lesen |
| | SYNW: | Vorlaufstop beim Schreiben |
| | SYNRW: | Vorlaufstop beim Lesen/Schreiben |
| <Zugriffsrechte>: | Schutzstufe für das Lesen / Schreiben von GUD über NC-Programm oder BTSS (optional) | |
| | APRP <Schutzstufe>: | Lesen: NC-Programm |
| | APWP <Schutzstufe>: | Schreiben: NC-Programm |
| | APRB <Schutzstufe>: | Lesen: BTSS |
| | APWB <Schutzstufe>: | Schreiben: BTSS |
| | <Schutzstufe>: | Wertebereich: 0 ... 7 |
| | Siehe "Attribut: Zugriffsrechte (APR, APW, APRP, APWP, APRB, APWB) (Seite 417)" | |
| <Datenklasse>: | Datenklassenzuordnung (nur SINUMERIK 828D!) | |
| | DCM: | Datenklasse M (= Manufacturer) |
| | DCI: | Datenklasse I (= Individual) |
| | DCU: | Datenklasse U (= User) |
| <Typ>: | Datentyp: | |
| | INT: | Ganzzahliger Wert mit Vorzeichen |
| | REAL: | Real-Zahl (LONG REAL nach IEEE) |
| | BOOL: | Wahrheitswert TRUE (1) / FALSE (0) |
| | CHAR: | ASCII-Zeichen |
| | STRING[<MaxLänge>]: | Zeichenkette definierter Länge |
| | AXIS: | Achs-/Spindelbezeichner |
| | FRAME: | Geometrische Angaben für eine statische Koordinatentransformation |
| | Siehe "Datentypen (Seite 428)" | |
| <Phys_Einheit>: | Physikalische Einheit (optional) | |
| | PHU <Einheit>: | physikalische Einheit |
| | Siehe "Attribut: Physikalische Einheit (PHU) (Seite 415)" | |

| | | |
|---|---|---|
| <Grenzwerte>: | unterer / oberer Grenzwert (optional) | |
| | LLI <Grenzwert>: | unterer Grenzwert (lower limit) |
| | ULI <Grenzwert>: | oberer Grenzwert (upper limit) |
| | Siehe "Attribut: Grenzwerte (LLI, ULI) (Seite 413)" | |
| <Name>: | Name der Variablen<br>**Hinweis**<br>• Maximal 31 Zeichen<br>• Die beiden ersten Zeichen müssen ein Buchstabe und/oder ein Unterstrich sein.<br>• Das "$"-Zeichen ist für Systemvariablen reserviert und darf nicht verwendet werden. | |
| [<Wert_1>,<br><Wert_2>,<br><Wert_3>]: | Angabe der Feldgrößen für 1- bis max. 3-dimensionale Feldvariablen (optional)<br>Zur Initialisierung von Feldvariablen siehe" Definition und Initialisierung von Feldvariablen (DEF, SET, REP) (Seite 422)" | |
| <Init_Wert>: | Initialisierungswert (optional)<br>siehe "Attribut: Initialisierungswert (Seite 410)"<br>Zur Initialisierung von Feldvariablen siehe "Definition und Initialisierung von Feldvariablen (DEF, SET, REP) (Seite 422)" | |

## Beispiele

### Beispiel 1: Definitionen von Anwendervariablen im Datenbaustein für Maschinenhersteller

```
Programmcode                                                    Kommentar
%_N_MGUD_DEF                                                    ; GUD-Baustein: Maschinenhersteller
$PATH=/_N_DEF_DIR
DEF CHAN REAL PHU 24 LLI 0 ULI 10 STROM_1, STROM_2
; Beschreibung
; Definition zweier GUD: STROM_1, STROM_2
; Gültigkeitsbereich: Kanalweit
; Datentyp: REAL
; VL-Stop: nicht programmiert => Defaultwert = kein VL-Stop
; Phys. Einheit: 24 = [A]
; Grenzwerte: Low = 0.0, High = 10.0
; Zugriffsrechte: nicht programmiert => Defaultwert = 7 = Schlüsselschalterstellung 0
; Initialisierungswert: nicht programmiert => Defaultwert = 0.0

DEF NCK REAL PHU 13 LLI 10 APWP 3 APRP 3 APWB 0 APRB 2 ZEIT_1=12, ZEIT_2=45
; Beschreibung
; Definition zweier GUD: ZEIT_1, ZEIT_2
; Gültigkeitsbereich: NC-weit
; Datentyp: REAL
; VL-Stop: nicht programmiert => Defaultwert = kein VL-Stop
```

| **Programmcode** | **Kommentar** |
|---|---|

```
; Phys. Einheit: 13 = [s]
; Grenzwerte: Low = 10.0, High = nicht programmiert => obere Definitionsbereichsgrenze
; Zugriffsrechte:
; NC-Programm: Schreiben/Lesen = 3 = Endanwender
; BTSS: Schreiben = 0 = Siemens, Lesen = 3 = Endanwender
; Initialisierungswert: ZEIT_1 = 12.0, ZEIT_2 = 45.0

DEF NCK APWP 3 APRP 3 APWB 0 APRB 3 STRING[5] GUD5_NAME = "COUNTER"
; Beschreibung
; Definition eines GUD: GUD5_NAME
; Gültigkeitsbereich: NC-weit
; Datentyp: STRING, max. 5 Zeichen
; VL-Stop: nicht programmiert => Defaultwert = kein VL-Stop
; Phys. Einheit: nicht programmiert => Defaultwert = 0 = keine phys. Einheit
; Grenzwerte: nicht programmiert => Definitionsbereichsgrenzen: Low = 0, High = 255
; Zugriffsrechte:
; NC-Programm: Schreiben/Lesen = 3 = Endanwender
; BTSS: Schreiben = 0 = Siemens, Lesen = 3 = Endanwender
; Initialisierungswert: "COUNTER"
M30
```

### Beispiel 2: Programm-globale und -lokale Anwendervariablen (PUD / LUD)

| **Programmcode** | **Kommentar** |
|---|---|
| PROC MAIN | ; Hauptprogramm |
| DEF INT VAR1 | ; PUD-Definition |
| ... | |
| SUB2 | ; Unterprogrammaufruf |
| ... | |
| M30 | |

| **Programmcode** | **Kommentar** |
|---|---|
| PROC SUB2 | ; Unterprogramm SUB2 |
| DEF INT VAR2 | ; LUD-DEFINITION |
| ... | |
| IF (VAR1==1) | ; PUD lesen |
| VAR1=VAR1+1 | ; PUD lesen und schreiben |
| VAR2=1 | ; LUD schreiben |
| ENDIF | |
| SUB3 | ; Unterprogrammaufruf |
| ... | |
| M17 | |

| Programmcode | Kommentar |
|---|---|
| PROC SUB3 | ; Unterprogramm SUB3 |
| ... | |
| IF (VAR1==1) | ; PUD lesen |
| VAR1=VAR1+1 | ; PUD lesen und schreiben |
| VAR2=1 | ; Fehler: LUD aus SUB2 nicht bekannt |
| ENDIF | |
| ... | |
| M17 | |

### Beispiel 3: Definition und Verwendung von Anwendervariablen vom Datentyp AXIS

| Programmcode | Kommentar |
|---|---|
| DEF AXIS ABSZISSE | ; 1. Geometrieachse |
| DEF AXIS SPINDLE | ; Spindel |
| ... | |
| IF ISAXIS(1) == FALSE GOTOF WEITER | |
| ABSZISSE = $P_AXN1 | |
| WEITER: | |
| ... | |
| SPINDLE=(S1) | ; 1. Spindel |
| OVRA[SPINDLE]=80 | ; Spindeloverride = 80% |
| SPINDLE=(S3) | ; 3. Spindel |

## Randbedingungen

### Globale Anwendervariablen (GUD)

Im Rahmen der Definition von globalen Anwendervariablen (GUD) sind folgende Maschinendaten zu berücksichtigen:

| Nr. | Bezeichner: $MN_ | Bedeutung |
|---|---|---|
| 11140 | GUD_AREA_ SAVE_TAB | zusätzliche Sicherung für GUD-Bausteine |
| 18118 [1)] | MM_NUM_GUD_MODULES | Anzahl GUD-Dateien im aktiven Filesystem |
| 18120 [1)] | MM_NUM_GUD_NAMES_NCK | Anzahl der globalen GUD-Namen |
| 18130 [1)] | MM_NUM_GUD_NAMES_CHAN | Anzahl der kanalspez. GUD-Namen |
| 18150 [1)] | MM_GUD_VALUES_MEM | Speicherplatz für globale GUD-Werte |
| 18660 [1)] | MM_NUM_SYNACT_GUD_REAL | Anzahl projektierbare GUD Datentyp REAL |
| 18661 [1)] | MM_NUM_SYNACT_GUD_INT | Anzahl projektierbare GUD Datentyp INT |
| 18662 [1)] | MM_NUM_SYNACT_GUD_BOOL | Anzahl projektierbare GUD Datentyp BOOL |
| 18663 [1)] | MM_NUM_SYNACT_GUD_AXIS | Anzahl projektierbare GUD Datentyp AXIS |
| 18664 [1)] | MM_NUM_SYNACT_GUD_CHAR | Anzahl projektierbare GUD Datentyp CHAR |
| 18665 [1)] | MM_NUM_SYNACT_GUD_STRING | Anzahl projektierbare GUD Datentyp STRING |

[1)] MD bei SINUMERIK 828D nur lesbar!

#### Kanalübergreifende Verwendung einer NC-globalen Anwendervariablen vom Datentyp AXIS

Eine NC-globale Anwendervariable vom Datentyp `AXIS`, die bei der Definition im Datenbaustein mit einem Achsbezeichner initialisiert wurde, kann nur dann in unterschiedlichen Kanälen der NC verwendet werden, wenn die Achse in diesen Kanälen die gleiche Kanalachsnummer hat.

Ist dies nicht der Fall, muss die Variable am NC-Programmanfang geladen oder, wie im folgenden Beispiel, die Funktion AXNAME(...) (siehe "Achsfunktionen (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Seite 891)") verwendet werden.

| **Programmcode** | **Kommentar** |
|---|---|
| `DEF NCK STRING[5] ACHSE="X"` | `; Definition im Datenbaustein` |
| `...` | |
| `N100 AX[AXNAME(ACHSE)]=111 G00` | `; Verwendung im NC-Programm` |

### 3.1.1.6 Redefinition von Systemdaten, Anwenderdaten und NC-Sprachbefehlen (REDEF)

Mit dem Befehl `REDEF` können die Attribute von Systemdaten, Anwenderdaten und NC-Sprachbefehle geändert werden. Grundvoraussetzung für eine Redefinition ist, dass sie zeitlich nach der entsprechenden Definition ausgeführt wird.

Bei einer Redefinition können nicht mehrere Attribute gleichzeitig geändert werden. Für jedes zu ändernde Attribut muss eine eigene `REDEF`-Anweisung programmiert werden.

Werden mehrere konkurrierende Attributänderungen programmiert, wird immer die letzte Änderung aktiv.

#### Attributwerte zurücksetzen

Die mit `REDEF` geänderten Attribute für Zugriffsrechte und Initialisierungszeitpunkt können durch erneute Programmierung von `REDEF`, gefolgt von dem Namen der Variablen oder des NC-Sprachbefehls, auf ihren Defaultwert zurückgesetzt werden:

- Zugriffsrechte: Schutzstufe 7
- Initialisierungszeitpunkt: keine Initialisierung bzw. Beibehalten des aktuellen Werts

#### Redefinierbare Attribute

Siehe "Übersicht definierbarer und redefinierbarer Attribute (Seite 421)".

#### Lokale Anwendervariablen (PUD / LUD)

Für lokale Anwendervariablen (PUD / LUD) dürfen keine Redefinitionen vorgenommen werden.

## Syntax

```
REDEF <Name> <VL_Stop>
REDEF <Name> <Phys_Einheit>
REDEF <Name> <Grenzwerte>
REDEF <Name> <Zugriffsrechte>
REDEF <Name> <Init_Zeitpunkt>
```

```
REDEF <Name> <Init_Zeitpunkt> <Init_Wert>
REDEF <Name> <Datenklasse>
REDEF <Name>
```

## Bedeutung

| | | |
|---|---|---|
| REDEF: | Befehl zur Redefinition eines bestimmten Attributs bzw. zum Zurücksetzen der Attribute "Zugriffsrechte" und/oder "Initialisierungszeitpunkt" von Systemvariablen, Anwendervariablen und NC-Sprachbefehlen | |
| <Name>: | Name einer bereits definierten Variablen oder eines NC-Sprachbefehls | |
| <VL-Stop>: | Vorlaufstop | |
| | SYNR: | Vorlaufstop beim Lesen |
| | SYNW: | Vorlaufstop beim Schreiben |
| | SYNRW: | Vorlaufstop beim Lesen/Schreiben |
| <Phys_Einheit>: | Physikalische Einheit | |
| | PHU <Einheit>: | physikalische Einheit |
| | Siehe "Attribut: Physikalische Einheit (PHU) (Seite 415)".<br>**Hinweis**<br>Nicht redefinierbar für:<br>• Systemvariablen<br>• Globale Anwenderdaten (GUD) der Datentypen: BOOL, AXIS, STRING, FRAME | |
| <Grenzwerte>: | unterer / oberer Grenzwert | |
| | LLI <Grenzwert>: | unterer Grenzwert (lower limit) |
| | ULI <Grenzwert>: | oberer Grenzwert (upper limit) |
| | Siehe "Attribut: Grenzwerte (LLI, ULI) (Seite 413)".<br>**Hinweis**<br>Nicht redefinierbar für:<br>• Systemvariablen<br>• Globale Anwenderdaten (GUD) der Datentypen: BOOL, AXIS, STRING, FRAME | |
| <Zugriffsrechte>: | Zugriffsrechte für das Lesen / Schreiben über Teileprogramm oder BTSS | |
| | APX <Schutzstufe>: | Ausführen: NC-Sprachelement |
| | APRP <Schutzstufe>: | Lesen: Teileprogramm |
| | APWP <Schutzstufe>: | Schreiben: Teileprogramm |
| | APRB <Schutzstufe>: | Lesen: BTSS |
| | APWB <Schutzstufe>: | Schreiben: BTSS |
| | <Schutzstufe>: | Wertebereich: 0 ... 7 |
| | Siehe "Attribut: Zugriffsrechte (APR, APW, APRP, APWP, APRB, APWB) (Seite 417)". | |

| | | |
|---|---|---|
| <Init_Zeitpunkt>: | Zeitpunkt zu dem die Variable reinitialisiert wird | |
| | INIPO: | Power On |
| | INIRE: | Hauptprogrammende, NC-Reset oder Power On |
| | INICF: | NEWCONF oder Hauptprogrammende, NC-Reset oder Power On |
| | PRLOC: | Hauptprogrammende, NC-Reset nach lokaler Änderung oder Power On |
| | Siehe "Attribut: Initialisierungswert (Seite 410)". | |
| <Init_Wert>: | Initialisierungswert<br>Bei Redefinition des Initialisierungswerts muss immer auch ein Initialisierungszeitpunkt (siehe <Init_Zeitpunkt>) angegeben werden.<br>Siehe "Attribut: Initialisierungswert (Seite 410)".<br>Zur Initialisierung von Feldvariablen siehe "Definition und Initialisierung von Feldvariablen (DEF, SET, REP) (Seite 422)".<br>**Hinweis**<br>Nicht redefinierbar für Systemvariablen, außer Settingdaten. | |
| <Datenklasse>: | Datenklassenzuordnung (nur SINUMERIK 828D!) | |
| | DCM: | Datenklasse M (= Manufacturer) |
| | DCI: | Datenklasse I (= Individual) |
| | DCU: | Datenklasse U (= User) |

## Beispiel

### Redefinitionen der Systemvariable $TC_DPCx im Datenbaustein für Maschinenhersteller

**Programmcode**

```
%_N_MGUD_DEF                                ; GUD-Baustein: Maschinenhersteller
N100 REDEF $TC_DPC1 APWB 2 APWP 3
N200 REDEF $TC_DPC2 PHU 21
N300 REDEF $TC_DPC3 LLI 0 ULI 200
N400 REDEF $TC_DPC4 INIPO (100, 101, 102, 103)
N800 REDEF $TC_DPC1
N900 REDEF $TC_DPC4
M30
```

zu N100: Zugriffsrecht Schreiben: BTSS = Schutzstufe 2, Teileprogramm = Schutzstufe 3

zu N200: Physikalische Einheit [ % ]

zu N300: unterer Grenzwert = 0, oberer Grenzwert = 200

| | |
|---|---|
| zu N400: | Die Feldvariable wird bei PowerOn mit den vier Werten initialisiert. |
| zu N800 / N900 | Zurücksetzen der Attributwerte "Zugriffsrechte" und/oder "Initialisierungszeitpunkt" |

**Hinweis**

**Verwendung von ACCESS-Dateien**

Bei Verwendung von ACCESS-Dateien muss die Redefinition der Zugriffsrechte von _N_MGUD_DEF nach _N_MACCESS_DEF verlagert werden.

## Randbedingungen

**Granularität**

Eine Redefinition bezieht sich immer auf die gesamte, durch ihren Namen eindeutig gekennzeichnete Variable. Es ist nicht möglich z. B. bei Feldvariablen für einzelne Feldelemente unterschiedliche Attributwerte zuzuweisen.

### 3.1.1.7 Attribut: Initialisierungswert

## Definition (DEF) von Anwendervariablen

Bei der Definition kann für folgende Anwendervariablen ein Initialisierungswert vorgegeben werden:

- globale Anwendervariablen (GUD)
- programmglobale Anwendervariablen (PUD)
- lokale Anwendervariablen (LUD)

## Redefinition (REDEF) von System- und Anwendervariablen

Bei der Redefinition kann für folgende Variablen ein Initialisierungswert vorgegeben werden:

- Systemdaten
  - Settingdaten
- Anwenderdaten
  - R-Parameter
  - Synchronaktionsvariable ($AC_MARKER, $AC_PARAM, $AC_TIMER)
  - Synchronaktions-GUD (SYG_xy[ ], mit x=R, I, B, A, C, S und y=S, M, U, 4, ..., 9)
  - EPS–Parameter
  - Werkzeugdaten-OEM
  - Magazindaten-OEM
  - globale Anwendervariablen (GUD)

### Reinitialisierungszeitpunkt

Bei der Redefinition kann der Zeitpunkt angegeben werden, zu dem die Variable reinitialisiert, d. h. wieder auf den Initialisierungswert gesetzt werden soll:

- `INIPO` (Power On)
  Die Variable wird bei Power On reinitialisiert.
- `INIRE` (Reset)
  Die Variable wird bei NC-Reset, BAG-Reset, Teileprogrammende (`M02` / `M30`) oder Power On reinitialisiert.
- `INICF` (NEWCONF)
  Die Variable wird bei der Funktion "Maschinendaten wirksam setzen" über HMI, Teileprogramm-Befehl `NEWCONF` oder NC-Reset, BAG-Reset, Teileprogrammende (M02 / M30) oder Power On reinitialisiert.
- `PRLOC` (programmlokale Änderung)
  Die Variable wird nur dann bei NC-Reset, BAG-Reset oder Teileprogrammende (M02 / M30) reinitialisiert, wenn sie im Rahmen des aktuellen Teileprogramms verändert worden ist.
  Das Attribut `PRLOC` darf nur in Zusammenhang mit programmierbaren Settingdaten (siehe folgende Tabelle) verwendet werden.

Tabelle 3-1 Programmierbare Settingdaten

| Nummer | Bezeichner | G-Befehl [1)] |
|---|---|---|
| 42000 | $SC_THREAD_START_ANGLE | `SF` |
| 42010 | $SC_THREAD_RAMP_DISP | `DITS` / `DITE` |
| 42400 | $SA_PUNCH_DWELLTIME | `PDELAYON` |
| 42800 | $SA_SPIND_ASSIGN_TAB | `SETMS` |
| 43210 | $SA_SPIND_MIN_VELO_G25 | `G25` |
| 43220 | $SA_SPIND_MAX_VELO_G26 | `G26` |
| 43230 | $SA_SPIND_MAX_VELO_LIMS | `LIMS` |
| 43300 | $SA_ASSIGN_FEED_PER_REV_SOURCE | `FPRAON` |

| Nummer | Bezeichner | G-Befehl [1)] |
|---|---|---|
| 43420 | $SA_WORKAREA_LIMIT_PLUS | G26 |
| 43430 | $SA_WORKAREA_LIMIT_MINUS | G25 |
| 43510 | $SA_FIXED_STOP_TORQUE | FXST |
| 43520 | $SA_FIXED_STOP_WINDOW | FXSW |
| 43700 | $SA_OSCILL_REVERSE_POS1 | OSP1 |
| 43710 | $SA_OSCILL_REVERSE_POS2 | OSP2 |
| 43720 | $SA_OSCILL_DWELL_TIME1 | OST1 |
| 43730 | $SA_OSCILL_DWELL_TIME2 | OST2 |
| 43740 | $SA_OSCILL_VELO | FA |
| 43750 | $SA_OSCILL_NUM_SPARK_CYCLES | OSNSC |
| 43760 | $SA_OSCILL_END_POS | OSE |
| 43770 | $SA_OSCILL_CTRL_MASK | OSCTRL |
| 43780 | $SA_OSCILL_IS_ACTIVE | OS |
| 43790 | $SA_OSCILL_START_POS | OSB |
| 1) mit diesem G-Befehl wird das Settingdatum angesprochen | | |

## Randbedingungen

### Initialisierungswert: globale Anwendervariablen (GUD)

- Für globale Anwendervariable (GUD) mit dem Gültigkeitsbereich NC kann als Initialisierungszeitpunkt nur INIPO (Power On) vorgegeben werden.
- Für globale Anwendervariablen (GUD) mit dem Gültigkeitsbereich CHAN kann als Initialisierungszeitpunkt neben INIPO (Power On) auch INIRE (Reset) oder INICF (NEWCONF) vorgegeben werden.
- Bei globalen Anwendervariablen (GUD) mit dem Gültigkeitsbereich CHAN und Initialisierungszeitpunkt INIRE (Reset) oder INICF (NEWCONF) werden bei NC-Reset, BAG-Reset und "Maschinendaten wirksam setzen" die Variablen nur in den Kanälen neu initialisiert, in denen die genannten Ereignisse ausgelöst wurden.

### Initialisierungswert: Datentyp FRAME

Für Variablen vom Datentyp FRAME darf kein Initialisierungswert angegeben werden. Variablen vom Datentyp FRAME werden implizit immer mit dem Defaultframe initialisiert.

### Initialisierungswert: Datentyp CHAR

Für Variablen vom Datentyp CHAR kann statt des ASCII-Codes (0...255) auch das entsprechende ASCII-Zeichen in Anführungszeichen programmiert werden, z.B. "A"

### Initialisierungswert: Datentyp STRING

Bei Variablen vom Datentyp STRING muss die Zeichenkette in Anführungszeichen gesetzt werden z.B.: ...= "MASCHINE_1"

### Initialisierungswert: Datentyp AXIS

Für Variablen vom Datentyp AXIS muss bei erweiterter Adressschreibweise der Achsbezeichner in Klammern gesetzt werden, z.B.: ...=(X3)

**Initialisierungswert: Systemvariable**

Für Systemvariable können durch Redefinition keine anwenderspezifischen Initialisierungswerte vorgegeben werden. Die Initialisierungswerte der Systemvariablen sind vom System fest vorgegeben. Durch Redefinition kann aber der Zeitpunkt (`INIRE`, `INICF`) zu dem die Systemvariable reinitialisiert wird geändert werden.

**Impliziter Initialisierungswert: Datentyp AXIS**

Für Variablen vom Datentyp `AXIS` wird folgender implizite Initialisierungswert verwendet:

- Systemdaten: "erste Geometrieachse"
- Synchronaktions-GUD (Bezeichnung: SYG_A*), PUD, LUD:
  Achsbezeichner aus Maschinendatum: MD20082
  $MC_AXCONF_CHANAX_DEFAULT_NAME

**Impliziter Initialisierungswert: Werkzeug- und Magazindaten**

Für Werkzeug- und Magazindaten können Initialisierungswerte über folgendes Maschinendatum vorgegeben werden: MD17520 $MN_TOOL_DEFAULT_DATA_MASK

**Hinweis**

**Synchronisation**

Die Synchronisation von Ereignissen die eine Reinitialisierung einer globalen Variable auslösen mit dem Lesen dieser Variable an anderer Stelle, liegt ausschließlich in der Verantwortung des Anwenders / Maschinenherstellers.

## Siehe auch

Variablen (Seite 393)

### 3.1.1.8 Attribut: Grenzwerte (LLI, ULI)

Ein oberer und unterer Grenzwert des Definitionsbereichs kann nur für folgende Datentypen vorgegeben werden:

- INT
- REAL
- CHAR

## Definition (DEF) von Anwendervariablen: Grenzwerte und implizite Initialisierungswerte

Wird bei der Definition einer Anwendervariablen von einem der oben genannten Datentypen kein expliziter Initialisierungswert definiert, wird die Variable auf den impliziten Initialisierungswert des Datentyps gesetzt:

- INT: 0
- REAL: 0.0
- CHAR: 0

Liegt der implizite Initialisierungswert außerhalb des durch die programmierten Grenzwerte festgelegten Definitionsbereichs, wird die Variable mit dem Grenzwert initialisiert, der dem impliziten Initialisierungswert am nächsten liegt:

- impliziter Initialisierungswert < unterer Grenzwert (LLI) ⇒ Initialisierungswert = unterer Grenzwert
- impliziter Initialisierungswert > oberer Grenzwert (ULI) ⇒ Initialisierungswert = oberer Grenzwert

Beispiele:

| **Programmcode** | **Kommentar** |
|---|---|
| `DEF REAL GUD1` | `; unterer Grenzwert = Definitionsbereichsgrenze`<br>`; oberer Grenzwert = Definitionsbereichsgrenze`<br>`; kein Initialisierungswert programmiert`<br>`; => impliziter Initialisierungswert = 0.0` |
| `DEF REAL LLI 5.0 GUD2` | `; unterer Grenzwert = 5.0`<br>`; oberer Grenzwert = Definitionsbereichsgrenze`<br>`; => Initialisierungswert = 5.0` |
| `DEF REAL ULI -5 GUD3` | `; unterer Grenzwert = Definitionsbereichsgrenze`<br>`; oberer Grenzwert = -5.0`<br>`; => Initialisierungswert = -5.0` |

## Redefinition (REDEF) von Anwendervariablen: Grenzwerte und aktuelle Istwerte

Werden bei der Redefinition der Grenzwerte einer Anwendervariablen diese so geändert, dass der aktuelle Istwert außerhalb des neuen Definitionsbereichs liegt, erfolgt ein Alarm und die Grenzwerte werden nicht übernommen.

**Hinweis**

**Redefinition (REDEF) von Anwendervariablen**

Bei der Redefinition der Grenzwerte einer Anwendervariablen ist auf das konsistente Ändern der folgenden Werte zu achten:

- Grenzwerte
- Istwert
- Initialisierungswert beim Redefinieren und beim automatischen Reinitialisieren aufgrund von INIPO, INIRE oder INICF

## Siehe auch

Variablen (Seite 393)

### 3.1.1.9 Attribut: Physikalische Einheit (PHU)

Eine physikalische Einheit kann nur für Variablen von folgende Datentypen vorgegeben werden:

- INT
- REAL

## Programmierbare physikalische Einheiten (PHU)

Die Angabe der physikalische Einheit erfolgt als Festkommazahl: `PHU <Einheit>`

Folgende physikalische Einheiten können programmiert werden:

| <Einheit> | Bedeutung | Physikalische Einheit |
|---|---|---|
| 0 | keine physikalische Einheit | - |
| 1 | Linear- oder Winkel–Position [1)2)] | [ mm ], [ inch ], [ Grad ] |
| 2 | Linear-Position [2)] | [ mm ], [ inch ] |
| 3 | Winkel-Position | [ Grad ] |
| 4 | Linear- oder Winkel-Geschwindigkeit [1)2)] | [ mm/min ], [ inch/min ], [ U/min ] |
| 5 | Linear-Geschwindigkeit [2)] | [ mm/min ] |
| 6 | Winkel-Geschwindigkeit | [ U/min ] |
| 7 | Linear- oder Winkel-Beschleunigung [1)2)] | [ $m/s^2$ ], [ $inch/s^2$ ], [ $U/s^2$ ] |
| 8 | Linear-Beschleunigung [2)] | [ $m/s^2$ ], [ $inch/s^2$ ] |
| 9 | Winkel-Beschleunigung | [ $U/s^2$ ] |
| 10 | Linear- oder Winkel-Ruck [1)2)] | [ $m/s^3$ ], [ $inch/s^3$ ], [ $U/s^3$ ] |
| 11 | Linear-Ruck [2)] | [ $m/s^3$ ], [ $inch/s^3$ ] |
| 12 | Winkel-Ruck | [ $U/s^3$ ] |
| 13 | Zeit | [ s ] |
| 14 | Lageregler-Verstärkung | [ 16.667/s ] |
| 15 | Umdrehungsvorschub [2)] | [ mm/U ], [ inch/U ] |
| 16 | Temperaturkompensation [1)2)] | [ mm ], [ inch ] |
| 18 | Kraft | [ N ] |
| 19 | Masse | [ kg ] |
| 20 | Trägheitsmoment [3)] | [ $kgm^2$ ] |
| 21 | Prozent | [ % ] |
| 22 | Frequenz | [ Hz ] |
| 23 | Spannung | [ V ] |
| 24 | Strom | [ A ] |
| 25 | Temperatur | [ °C ] |
| 26 | Winkel | [ Grad ] |
| 27 | KV | [ 1000/min ] |
| 28 | Linear- oder Winkel-Position [3)] | [ mm ], [ inch ], [ Grad ] |
| 29 | Schnittgeschwindigkeit [2)] | [ m/min ], [ feet/min ] |
| 30 | Umfangsgeschwindigkeit [2)] | [ m/s], [ feet/s ] |
| 31 | Widerstand | [ Ohm ] |

| <Einheit> | Bedeutung | Physikalische Einheit |
|---|---|---|
| 32 | Induktivität | [ mH ] |
| 33 | Drehmoment 3) | [ Nm ] |
| 34 | Drehmomentkonstante 3) | [ Nm/A ] |
| 35 | Stromreglerverstärkung | [ V/A ] |
| 36 | Drehzahlreglerverstärkung 3) | [ Nm/(rad*s) ] |
| 37 | Drehzahl | [ U/min ] |
| 42 | Leistung | [ kW ] |
| 43 | Strom, klein | [ µA ] |
| 46 | Drehmoment, klein 3) | [ µNm ] |
| 48 | Promille | - |
| 49 | - | [ Hz/s ] |
| 65 | Durchfluss | [ l/min ] |
| 66 | Druck | [ bar ] |
| 67 | Volumen 3) | [ $cm^3$ ] |
| 68 | Streckenverstärkung 3) | [ mm/(V*min) ] |
| 69 | Streckenverstärkung Kraftregler | [ N/V ] |
| 155 | Gewindesteigung 3) | [ mm/U ], [ inch/U] |
| 156 | Gewindesteigungsänderung 3) | [ mm/U / U ], [ inch/U / U] |
| 1) Die physikalische Einheit ist abhängig vom Achstyp: Linear- oder Rundachse | | |
| 2) Maßsystem-Umschaltung<br>G70/G71(inch/metrisch)<br>Nach einer Umschaltung des Grundsystems (MD10240 $MN_SCALING_SYSTEM_IS_METRIC) mit G70/G71 erfolgt bei Schreib/Lesezugriffen auf längenbehaftete System- und Anwendervariablen **keine** Umrechnung der Werte (Istwert, Defaultwert und Grenzwerte)<br>G700/G710(inch/metrisch)<br>Nach einer Umschaltung des Grundsystems (MD10240 $MN_SCALING_SYSTEM_IS_METRIC) mit G700/G710 erfolgt bei Schreib/Lesezugriffen auf längenbehaftete System- und Anwendervariablen **eine** Umrechnung der Werte (Istwert, Defaultwert und Grenzwerte) | | |
| 3) Die Variable wird **nicht** automatische in das aktuelle Maßsystem der NC (inch/metrisch) umgerechnet. Die Umrechnung liegt ausschließlich in der Verantwortung des Anwenders / Maschinenherstellers. | | |

**Hinweis**

**Ebenenüberlauf durch Formatumrechnung**

Das interne Ablageformat für alle Anwendervariablen (GUD / PUD / LUD) mit längenbehafteten physikalischen Einheiten ist metrisch. Eine exzessive Verwendung derartiger Variablen im Hauptlauf des NCK, z. B. in Synchronaktionen, kann bei einer Maßsystemumschaltung zu einem Rechenzeitüberlauf der Interpolatorebene, Alarm 4240, führen.

**Hinweis**

**Kompatibilität von Einheiten**

Bei der Verwendung von Variablen (Zuweisung, Vergleich, Berechnung etc.) erfolgt keine Prüfung auf Kompatibilität der beteiligten Einheiten. Eine gegebenenfalls erforderliche Umrechnung liegt ausschließlich in der Verantwortung des Anwenders / Maschinenherstellers.

## Siehe auch

Variablen (Seite 393)

### 3.1.1.10 Attribut: Zugriffsrechte (APR, APW, APRP, APWP, APRB, APWB)

#### Bezeichnung

Die Bezeichnung der Zugriffsattribute AP... setzt sich zusammen aus:

1. A: **A**ccess
2. P: **P**rotection
3. R / W: **R**ead / **W**rite
4. P / B: **P**rogram / **B**TSS

#### Zugriffsrechte / Schutzstufen

Den Zugriffsrechten entsprechen folgende bei der Programmierung anzugebende Schutzstufen:

| Zugriffsrecht | Schutzstufe |
|---|---|
| Kennwort System | 0 |
| Kennwort Maschinenhersteller | 1 |
| Kennwort Service | 2 |
| Kennwort Endanwender | 3 |
| Schlüsselschalter Stellung 3 | 4 |
| Schlüsselschalter Stellung 2 | 5 |
| Schlüsselschalter Stellung 1 | 6 |
| Schlüsselschalter Stellung 0 | 7 |

## Definition (DEF) von Anwenderdaten

Zugriffsrechte (APR... / APW...) können für folgende Daten definiert werden:

- globale Anwenderdaten (GUD)

## Redefinition (REDEF) von System- und Anwenderdaten

Zugriffsrechte (APR... / APW...) können für folgende Daten redefiniert werden:

- Systemdaten
  - Maschinendaten

    **Hinweis**

    **Redefinition der Leserechte von Maschinendaten**

    Die Schutzstufe für das Lesen von Maschinendaten kann nur mit dem Schlüsselwort APR gemeinsam für Teileprogramm und BTSS gesetzt werden.

    Die Schlüsselwörter APRP und APRB werden bei der Redefinition der Leserechte nicht unterstützt und führen zur Meldung von Alarm 12490 "Zugriffsrecht APRP/APRB <Schutzstufe> wurde nicht eingestellt".

  - Settingdaten
  - Systemvariable
  - Prozessdaten
  - Magazindaten
  - Werkzeugdaten
- Anwenderdaten
  - R-Parameter
  - Synchronaktionsvariable ($AC_MARKER, $AC_PARAM, $AC_TIMER)
  - Synchronaktions-GUD (SYG_xy[ ], mit x=R, I, B, A, C, S und y=S, M, U, 4, ..., 9)
  - EPS–Parameter
  - Werkzeugdaten-OEM
  - Magazindaten-OEM
  - globale Anwendervariablen (GUD)

    **Hinweis**

    Bei der Redefinition kann das Zugriffsrecht auf eine Variable zwischen der niedrigsten Schutzstufe 7 und der eigenen Schutzstufe, z. B. 1 (Maschinenhersteller), frei vergeben werden.

## Redefinition (REDEF) von NC-Sprachbefehlen

Das Zugriffs- bzw. Ausführungsrecht (APX) kann für folgende NC-Sprachbefehle redefiniert werden:

- G-Befehle / Wegbedingungen (Seite 173)
- Vordefinierte Funktionen (Seite 1312)
- Vordefinierte Unterprogrammaufrufe

- Anweisung DO bei Synchronaktionen
- Programmbezeichner von Zyklen
  Der Zyklus muss in einem Zyklenverzeichnis abgelegt sein und eine PROC-Anweisung enthalten.

### Zugriffsrechte bezüglich NC-Programmen und Zyklen (APRP, APWP)

Die unterschiedlichen Zugriffsrechte haben für den Zugriff aus einem NC-Programm bzw. Zyklus folgende Auswirkungen:

- APRP 0 / APWP 0
  - Beim Abarbeiten des NC-Programms muss das System-Kennwort gesetzt sein
  - Der Zyklus muss im Verzeichnis _N_CST_DIR (System) abgelegt sein
  - Für das Verzeichnis _N_CST_DIR muss im MD11160 $MN_ACCESS_EXEC_CST das Ausführungsrecht auf System eingestellt sein
- APRP 1 / APWP 1 bzw. APRP 2 / APWP 2
  - Beim Abarbeiten des NC-Programms muss das Maschinenhersteller- bzw. Service-Kennwort gesetzt sein
  - Der Zyklus muss im Verzeichnis _N_CMA_DIR (Maschinenhersteller) oder _N_CST_DIR abgelegt sein
  - Für die Verzeichnisse _N_CMA_DIR bzw. _N_CST_DIR müssen in den Maschinendaten MD11161 $MN_ACCESS_EXEC_CMA bzw. MD11160 $MN_ACCESS_EXEC_CST die Ausführungsrechte mindestens auf Maschinenhersteller eingestellt sein
- APRP 3 / APWP 3
  - Beim Abarbeiten des NC-Programms muss das Endanwender-Kennwort gesetzt sein
  - Der Zyklus muss im Verzeichnis _N_CUS_DIR (Anwender), _N_CMA_DIR oder _N_CST_DIR abgelegt sein
  - Für die Verzeichnisse _N_CUS_DIR, _N_CMA_DIR bzw. _N_CST_DIR müssen in den Maschinendaten MD11162 $MN_ACCESS_EXEC_CUS, MD11161 $MN_ACCESS_EXEC_CMA bzw. MD11160 $MN_ACCESS_EXEC_CST die Ausführungsrechte mindestens auf Endanwender eingestellt sein
- APRP 4...7 / APWP 4...7
  - Beim Abarbeiten des NC-Programms muss Schlüsselschalterstellung 3 ... 0 eingestellt sein
  - Der Zyklus muss im Verzeichnis _N_CUS_DIR, _N_CMA_DIR oder _N_CST_DIR abgelegt sein
  - Für die Verzeichnisse _N_CUS_DIR, _N_CMA_DIR bzw. _N_CST_DIR müssen in den Maschinendaten MD11162 $MN_ACCESS_EXEC_CUS, MD11161 $MN_ACCESS_EXEC_CMA bzw. MD11160 $MN_ACCESS_EXEC_CST die Ausführungsrechte mindestens auf die entsprechende Schlüsselschalterstellung eingestellt sein

## Zugriffsrechte bezüglich BTSS (APRB, APWB)

Die Zugriffsrechte (APRB, APWB) beschränken den Zugriff auf System- und Anwendervariablen über BTSS für alle Systemkomponenten (HMI, PLC, externe Rechner, EPS-Dienste, etc.) gleichermaßen.

**Hinweis**

**HMI-lokale Zugriffsrechte**

Bei Änderungen von Zugriffsrechten von Systemdaten muss darauf geachtet werden, dass diese konsistent zu den über HMI-Mechanismen festgelegten Zugriffsrechten erfolgt.

## Zugriffsattribute APR / APW

Aus Kombatibilitätsgründen werden die Attribute APR und APW implizit auf die Attribute APRP / APRB und APWP / APWB abgebildet:

- APR x ⇒ APRP x APRB x
- APW y ⇒ APWP y APWB y

## Zugriffsrechte über ACCESS-Dateien

Bei der Verwendung von ACCESS-Dateien für die Vergabe von Zugriffsrechten, dürfen Redefinitionen von Zugriffsrechten für Systemdaten, Anwenderdaten und NC-Sprachbefehlen ausschließlich in ACCESS-Dateien erfolgen. Eine Ausnahme bilden globale Anwenderdaten (GUD). Für diese muss eine Redefinition der Zugriffsrechte weiterhin in den entsprechenden Definitionsdateien *_DEF erfolgen.

Für einen durchgehenden Zugriffsschutz müssen die Maschinendaten für die Ausführungsrechte und den Zugriffsschutz der entsprechenden Verzeichnisse konsistenten angepasst werden.

Daraus ergibt sich folgende prinzipielle Vorgehensweise:

1. Erstellen der benötigten Definitionsdateien:
   - _N_DEF_DIR/_N_SACCESS_DEF
   - _N_DEF_DIR/_N_MACCESS_DEF
   - _N_DEF_DIR/_N_UACCESS_DEF
2. Parametrieren des Schreibrechts für die Definitionsdateien auf den für die Redefinition erforderlichen Wert:
   - MD11170 $MN_ACCESS_WRITE_SACCESS = <Schutzstufe>
   - MD11171 $MN_ACCESS_WRITE_MACCESS = <Schutzstufe>
   - MD11172 $MN_ACCESS_WRITE_UACCESS = <Schutzstufe>

3. Für Zugriffe auf geschützte Elemente aus Zyklen heraus müssen die Ausführungs- und Schreibrechte der Zyklenverzeichnisse _N_CST_DIR, _N_CMA_DIR und _N_CST_DIR angepasst werden:
   Ausführungsrechte
   - MD11160 $MN_ACCESS_EXEC_CST = <Schutzstufe>
   - MD11161 $MN_ACCESS_EXEC_CMA = <Schutzstufe>
   - MD11162 $MN_ACCESS_EXEC_CUS = <Schutzstufe>

   Schreibrechte
   - MD11165 $MN_ACCESS_WRITE_CST = <Schutzstufe>
   - MD11166 $MN_ACCESS_WRITE_CMA = <Schutzstufe>
   - MD11167 MN_ACCESS_WRITE_CUS = <Schutzstufe>

   Das Ausführungsrecht muss mindestens auf die gleiche Schutzstufe wie die höchste Schutzstufe des verwendeten Elements gesetzt werden.
   Das Schreibrecht muss mindestens auf die gleiche Schutzstufe wie das Ausführungsrecht gesetzt werden.
4. Die Schreibrechte der HMI-lokalen Zyklenverzeichnisse müssen auf die gleiche Schutzstufe wie die der NC-lokalen Zyklenverzeichnisse gesetzt werden.

**Unterprogrammaufrufe in ACCESS-Dateien**

Für die weitere Strukturierung des Zugriffsschutzes können in den ACCESS-Dateien auch Unterprogramme (Kennung SPF oder MPF) aufgerufen werden. Die Unterprogramme erben dabei die Ausführungsrechte der aufrufenden ACCESS-Datei.

**Hinweis**

In den ACCESS-Dateien können nur die Zugriffsrechte redefiniert werden. Alle anderen Attribute müssen weiterhin in den entsprechenden Definitionsdateien programmiert bzw. redefiniert werden.

## Siehe auch

Variablen (Seite 393)

### 3.1.1.11 Übersicht definierbarer und redefinierbarer Attribute

Die folgenden Tabellen zeigen bei welchen Datenarten welche Attribute definiert (`DEF`) und/ oder redefiniert (`REDEF`) werden können.

## Systemdaten

| Datenart | Init.Wert | Grenzwerte | phys. Einheit | Zugriffsrechte | Datenklasse (nur 828D) |
|---|---|---|---|---|---|
| Maschinendaten | --- | --- | --- | REDEF | REDEF |
| Settingdaten | REDEF | --- | --- | REDEF | --- |

| Datenart | Init.Wert | Grenzwerte | phys. Einheit | Zugriffsrechte | Datenklasse (nur 828D) |
|---|---|---|---|---|---|
| FRAME-Daten | --- | --- | --- | REDEF | --- |
| Prozessdaten | --- | --- | --- | REDEF | --- |
| Spindelsteigungsfehlerkomp. (EEC) | --- | --- | --- | REDEF | --- |
| Durchhangkompensation (CEC) | --- | --- | --- | REDEF | --- |
| Quadrantenfehlerkompensation (QEC) | --- | --- | --- | REDEF | --- |
| Magazindaten | --- | --- | --- | REDEF | --- |
| Werkzeugdaten | --- | --- | --- | REDEF | --- |
| Schutzbereiche | --- | --- | --- | REDEF | --- |
| orientierbare Werkzeugträger | --- | --- | --- | REDEF | --- |
| kinematische Ketten | --- | --- | --- | REDEF | --- |
| 3D-Schutzbereiche | --- | --- | --- | REDEF | --- |
| Arbeitsfeldbegrenzung | --- | --- | --- | REDEF | --- |

## Anwenderdaten

| Datenart | Init.Wert | Grenzwerte | phys. Einheit | Zugriffsrechte | Datenklasse |
|---|---|---|---|---|---|
| R–Parameter | REDEF | REDEF | REDEF | REDEF | --- |
| Synchronaktionsvariable ($AC_...) | REDEF | REDEF | REDEF | REDEF | --- |
| Synchronaktions–GUD (SYG_...) | REDEF | REDEF | REDEF | REDEF | --- |
| EPS–Parameter | REDEF | REDEF | REDEF | REDEF | --- |
| Werkzeugdaten–OEM | REDEF | REDEF | REDEF | REDEF | --- |
| Magazindaten–OEM | REDEF | REDEF | REDEF | REDEF | --- |
| globale Anwendervariablen (GUD) | DEF / REDEF | DEF | DEF | DEF / REDEF | DEF / REDEF |
| lokale Anwendervariablen (PUD / LUD) | DEF | DEF | DEF | --- | --- |

## Siehe auch

Variablen (Seite 393)

### 3.1.1.12 Definition und Initialisierung von Feldvariablen (DEF, SET, REP)

Eine Anwendervariable kann als 1- bis maximal 3-dimensionales Feld (Array) definiert werden:

- 1-dimensional: `DEF <Datentyp> <Variablenname>[<n>]`
- 2-dimensional: `DEF <Datentyp> <Variablenname>[<n>,<m>]`
- 3-dimensional: `DEF <Datentyp> <Variablenname>[<n>,<m>,<o>]`

**Hinweis**

Anwendervariable vom Datentyp STRING können maximal als 2-dimensionales Feld definiert werden.

### Datentypen

Anwendervariable können als Felder für folgende Datentypen definiert werden: BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME

### Wertzuweisung an Feldelemente

Wertzuweisungen an Feldelemente können zu folgenden Zeitpunkten vorgenommen werden:

- bei der Felddefinition (Initialisierungswerte)
- während des Programmablaufs

Wertzuweisung können dabei erfolgen über:

- explizite Angabe eines Feldelements
- explizite Angabe eines Feldelements als Startelement und Angabe einer Werteliste (`SET`)
- explizite Angabe eines Feldelements als Startelement und Angabe eines Wertes und der Häufigkeit seiner Wiederholung (`REP`)

**Hinweis**

Anwendervariablen vom Datentyp FRAME können keine Initialisierungswerte zugewiesen werden.

## Syntax (`DEF`)

```
DEF <Datentyp> <Variablenname>[<n>,<m>,<o>]
DEF  STRING[<Stringlänge>] <Variablenname>[<n>,<m>]
```

## Syntax (`DEF...=SET...`)

Verwendung einer Werteliste:

- bei der Definition:
  ```
  DEF <Datentyp> <Variablenname>[<n>,<m>,<o>] =
  SET(<Wert1>,<Wert2>,...)
  ```
  gleichbedeutend mit:
  ```
  DEF <Datentyp> <Variablenname>[<n>,<m>,<o>] =
  (<Wert1>,<Wert2>,...)
  ```

**Hinweis**

Bei der Initialisierung über eine Werteliste ist die Angabe von `SET` optional.

- bei einer Wertzuweisung:
  ```
  <Variablenname>[<n>,<m>,<o>] = SET(<WERT1>,<Wert2>,...)
  ```

## Syntax (DEF...=REP...)

Verwendung eines Werte mit Wiederholung

- bei der Definition:
  ```
  DEF <Datentyp> <Variablenname>[<n>,<m>,<o>] = REP(<Wert>)
  DEF <Datentyp> <Variablenname>[<n>,<m>,<o>] =
  REP(<Wert>,<Anzahl_Feldelemente>)
  ```
- bei einer Wertzuweisung:
  ```
  <Variablenname>[<n>,<m>,<o>] = REP(<Wert>)
  <Variablenname>[<n>,<m>,<o>] = REP(<Wert>,<Anzahl_Feldelemente>)
  ```

## Bedeutung

| | | |
|---|---|---|
| DEF: | Befehl zur Definition von Variablen | |
| <Datentyp>: | Datentyp der Variablen | |
| | Wertebereich:<br>• bei Systemvariablen:<br>BOOL, CHAR, INT, REAL, STRING, AXIS<br>• bei GUD- oder LUD-Variablen:<br>BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME | |
| <Stringlänge>: | Maximale Anzahl der Zeichen beim Datentyp STRING | |
| <Variablenname>: | Variablenname | |
| [<n>,<m>,<o>]: | Feldgrößen bzw. Feldindizes | |
| <n>: | Feldgröße bzw. Feldindex für 1. Dimension | |
| | Typ: | INT (bei Systemvariablen auch AXIS) |
| | Wertebereich: | Max. Feldgröße: 65535<br>Feldindex: 0 ≤ n ≤ 65534 |
| <m>: | Feldgröße bzw. Feldindex für 2. Dimension | |
| | Typ: | INT (bei Systemvariablen auch AXIS) |
| | Wertebereich: | Max. Feldgröße: 65535<br>Feldindex: 0 ≤ m ≤ 65534 |
| <o>: | Feldgröße bzw. Feldindex für 3. Dimension | |
| | Typ: | INT (bei Systemvariablen auch AXIS) |
| | Wertebereich: | Max. Feldgröße: 65535<br>Feldindex: 0 ≤ o ≤ 65534 |
| SET: | Wertzuweisung über die angegebenen Werteliste | |
| (<Wert1>,<Wert2>,...): | Werteliste | |
| REP: | Wertzuweisung über den angegebenen <Wert> | |

| | |
|---|---|
| <Wert>: | Wert, mit dem die Feldelemente bei der Initialisierung mit REP beschrieben werden sollen. |
| <Anzahl_Feldelemente>: | Anzahl der Feldelemente, die mit dem angegebenen <Wert> beschrieben werden sollen. Für die restlichen Feldelemente gilt abhängig vom Zeitpunkt:<br>• Initialisierung bei der Felddefinition:<br>→ Die restlichen Feldelemente werden mit Null beschrieben<br>• Zuweisung während des Programmlaufs:<br>→ Die aktuellen Werte der Feldelemente bleiben unverändert.<br>Ist der Parameter nicht programmiert, werden alle Feldelemente mit <Wert> beschrieben.<br>Ist der Parameter gleich Null, gilt abhängig vom Zeitpunkt:<br>• Initialisierung bei der Felddefinition:<br>→ Alle Elemente werden mit Null vorbelegt<br>• Zuweisung während des Programmlaufs:<br>→ Die aktuellen Werte der Feldelemente bleiben unverändert. |

## Feldindex

Die implizite Reihenfolge der Feldelemente z. B. bei einer Wertzuweisung über SET oder REP erfolgt durch Iteration der Feldindizes von rechts nach links.

Beispiel: Initialisierung eines 3-dimensionalen Feldes mit 24 Feldelementen:

```
DEF INT FELD[2,3,4] = REP(1,24)
  FELD[0,0,0] = 1      1. Feldelement
  FELD[0,0,1] = 1      2. Feldelement
  FELD[0,0,2] = 1      3. Feldelement
  FELD[0,0,3] = 1      4. Feldelement
  ...
  FELD[0,1,0] = 1      5. Feldelement
  FELD[0,1,1] = 1      6. Feldelement
  ...
  FELD[0,2,3] = 1      12. Feldelement
  FELD[1,0,0] = 1      13. Feldelement
  FELD[1,0,1] = 1      14. Feldelement
  ...
  FELD[1,2,3] = 1      24. Feldelement
```

entsprechend:

```
FOR n=0 TO 1
  FOR m=0 TO 2
    FOR o=0 TO 3
```

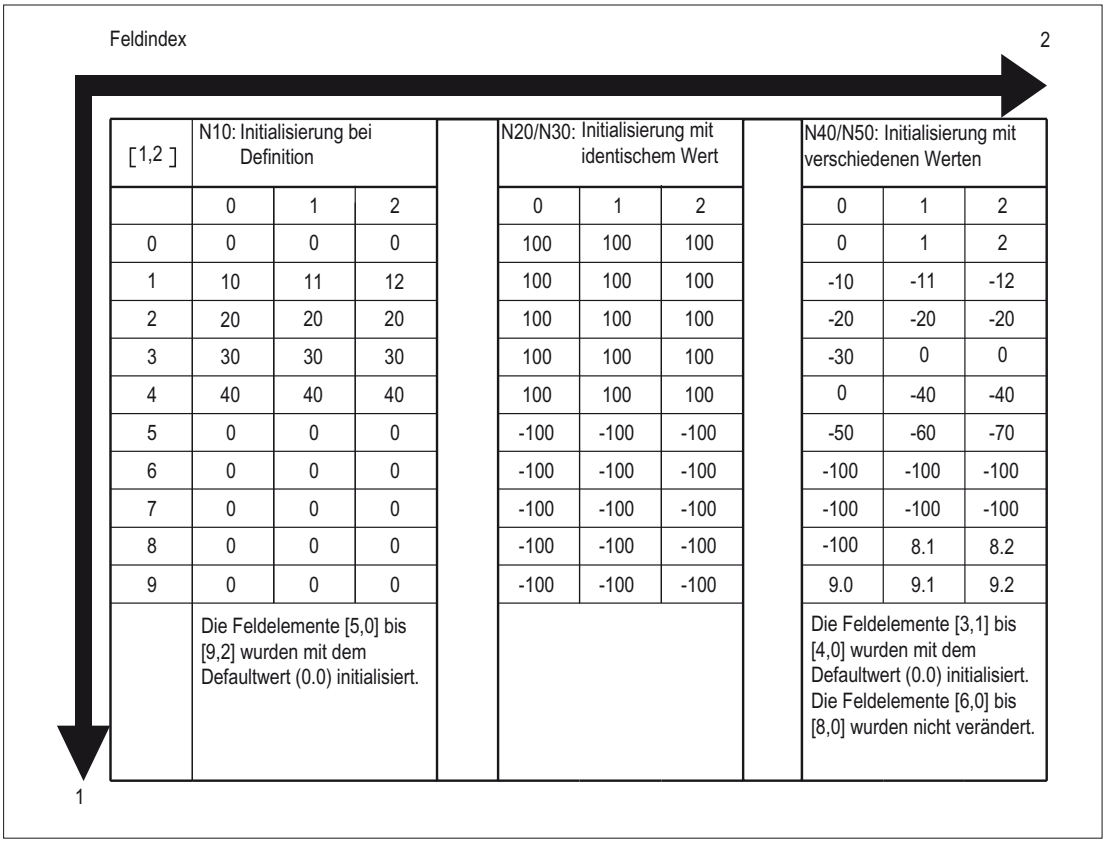
```
      FELD[n,m,o] = 1
    ENDFOR
  ENDFOR
ENDFOR
```

## Beispiel: Initialisierung kompletter Variablenfelder

Aktuelle Belegung siehe Abbildung.

**Programmcode**

```
N10 DEF REAL FELD1[10,3]=SET(0,0,0,10,11,12,20,20,20,30,30,30,40,40,40,)
N20 FELD1[0,0]=REP(100)
N30 FELD1[5,0]=REP(-100)
N40 FELD1[0,0]=SET(0,1,2,-10,-11,-12,-20,-20,-20,-30, , , ,-40,-40,-50,-60,-70)
N50 FELD1[8,1]=SET(8.1,8.2,9.0,9.1,9.2)
```




| [1,2 ] | N10: Initialisierung bei Definition | | | N20/N30: Initialisierung mit identischem Wert | | | N40/N50: Initialisierung mit verschiedenen Werten | | |
|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 |
| 0 | 0 | 0 | 0 | 100 | 100 | 100 | 0 | 1 | 2 |
| 1 | 10 | 11 | 12 | 100 | 100 | 100 | -10 | -11 | -12 |
| 2 | 20 | 20 | 20 | 100 | 100 | 100 | -20 | -20 | -20 |
| 3 | 30 | 30 | 30 | 100 | 100 | 100 | -30 | 0 | 0 |
| 4 | 40 | 40 | 40 | 100 | 100 | 100 | 0 | -40 | -40 |
| 5 | 0 | 0 | 0 | -100 | -100 | -100 | -50 | -60 | -70 |
| 6 | 0 | 0 | 0 | -100 | -100 | -100 | -100 | -100 | -100 |
| 7 | 0 | 0 | 0 | -100 | -100 | -100 | -100 | -100 | -100 |
| 8 | 0 | 0 | 0 | -100 | -100 | -100 | -100 | 8.1 | 8.2 |
| 9 | 0 | 0 | 0 | -100 | -100 | -100 | 9.0 | 9.1 | 9.2 |
| | Die Feldelemente [5,0] bis [9,2] wurden mit dem Defaultwert (0.0) initialisiert. | | | | | | Die Feldelemente [3,1] bis [4,0] wurden mit dem Defaultwert (0.0) initialisiert. Die Feldelemente [6,0] bis [8,0] wurden nicht verändert. | | |

## Siehe auch

Definition und Initialisierung von Feldvariablen (DEF, SET, REP) (Seite 422)

Variablen (Seite 393)

## Weitere Informationen (SET)

### Initialisierung bei der Definition

- Es werden, beginnend beim 1. Feldelement, so viele Feldelemente mit den Werten aus der Werteliste initialisiert, wie Elemente in der Werteliste programmiert sind.
- Feldelemente ohne explizit angegebene Werte in der Werteliste (Lücken in der Werteliste) werden mit 0 belegt.
- Bei Variablen vom Datentyp AXIS sind Lücken in der Werteliste nicht zugelassen.
- Enthält die Werteliste mehr Werte als Feldelemente definiert sind, wird ein Alarm angezeigt.

### Wertzuweisung im Programmablauf

Bei der Wertzuweisung im Programmablauf gelten die oben bei der Definition beschriebenen Regeln. Zusätzlich gibt es folgende Möglichkeiten:

- Als Elemente in der Werteliste sind auch Ausdrücke erlaubt.
- Die Wertzuweisung beginnt bei dem programmierten Feldindex. Hierdurch lassen sich gezielt Teilfelder mit Werten belegen.

Beispiel:

| **Programmcode** | **Kommentar** |
|---|---|
| `DEF INT FELD[5,5]` | `; Felddefinition` |
| `FELD[0,0]=SET(1,2,3,4,5)` | `; Wertzuweisung an die ersten 5 Feldelemente [0,0] - [0,4]` |
| `FELD[0,0]=SET(1,2, , ,5)` | `; Wertzuweisung mit Lücke an die ersten 5 Feldelemente [0,0] - [0,4], Feldelemente [0,2] und [0,3] = 0` |
| `FELD[2,3]=SET(VARIABLE,4*5.6)` | `; Wertzuweisung mit Variable und Ausdruck ab Feldindex [2,3]:`<br>`[2,3] = VARIABLE`<br>`[2,4] = 4 * 5.6 = 22.4` |

## Weitere Informationen (REP)

### Initialisierung bei der Definition

- Alle oder die optional angegebene Anzahl an Feldelementen werden mit dem angegebenen Wert (Konstante) initialisiert.
- Variablen vom Datentyp FRAME können nicht initialisiert werden.

Beispiel:

| **Programmcode** | **Kommentar** |
|---|---|
| `DEF REAL varName[10]=REP(3.5,4)` | `; Felddefinition und Feldelemente [0] bis [3] mit Wert 3,5 initialisieren` |

### Wertzuweisung im Programmablauf

Bei der Wertzuweisung im Programmablauf gelten die oben bei der Definition beschriebenen Regeln. Zusätzlich gibt es folgende Möglichkeiten:

- Als Elemente in der Werteliste sind auch Ausdrücke erlaubt.
- Die Wertzuweisung beginnt bei dem programmierten Feldindex. Hierdurch lassen sich gezielt Teilfelder mit Werten belegen.

Beispiele:

| Programmcode | Kommentar |
|---|---|
| `DEF REAL varName[10]` | `; Felddefinition` |
| `varName[5]=REP(4.5,3)` | `; Feldelemente [5] bis [7] = 4,5` |
| `R10=REP(2.4,3)` | `; R-Parameter R10 bis R12 = 2,4` |
| `DEF FRAME FRM[10]` | `; Felddefinition` |
| `FRM[5]=REP(CTRANS(X,5))` | `; Feldelemente [5] bis [9] = CTRANS(X,5)` |

## Siehe auch

Definition und Initialisierung von Feldvariablen (DEF, SET, REP) (Seite 422)

## 3.1.1.13 Datentypen

Folgende Datentypen stehen in der NC zur Verfügung:

| Datentyp | Bedeutung | Wertebereich |
|---|---|---|
| INT | ganzzahliger Wert mit Vorzeichen | -2147483648 ... +2147483647 |
| REAL | Real-Zahl (LONG REAL nach IEEE) | ±(~2,2*10$^{-308}$ … ~1,8*10$^{+308}$) |
| BOOL | Wahrheitswert TRUE (1) und FALSE (0) | 1, 0 |
| CHAR | ASCII-Zeichen | ASCII-Code 0 ... 255 |
| STRING | Zeichenkette definierter Länge | maximal 200 Zeichen (keine Sonderzeichen) |
| AXIS | Achs-/Spindelbezeichner | Kanalachsbezeichner |
| FRAME | Geometrische Angaben für eine statische Koordinatentransformation (Verschieben, Drehen, Skalieren, Spiegeln) | --- |

## Implizite Datentypwandlungen

Folgende Datentypwandlungen sind möglich und werden bei Zuweisungen und Parameterübergaben implizit vorgenommen:

| von ↓/ nach → | REAL | INT | BOOL |
|---|---|---|---|
| REAL | x | o | & |
| INT | x | x | & |

| von ↓/ nach → | REAL | INT | BOOL |
|---|---|---|---|
| BOOL | x | x | x |
| x: ohne Einschränkungen möglich<br>o: Datenverlust durch Überschreitung des Wertebereichs möglich ⇒ Alarm;<br>Rundung: Nachkommawert ≥ 0,5 ⇒ aufrunden, Nachkommawert < 0,5 ⇒ abrunden<br>&: Wert ≠ 0 ⇒ TRUE, Wert == 0 ⇒ FALSE | | | |

## Siehe auch

Variablen (Seite 393)

### 3.1.1.14 Minimum, Maximum und Bereich von Variablen (MINVAL, MAXVAL, BOUND)

Mit den Befehlen `MINVAL` und `MAXVAL` können die Werte zweier Variablen miteinander verglichen werden. Als Ergebnis wird der kleinere Wert (bei `MINVAL`) bzw. größere Wert (bei `MAXVAL`) zurückgeliefert.

Mit dem Befehl `BOUND` kann geprüft werden, ob der Wert einer Prüfvariablen innerhalb eines definierten Wertebereichs liegt.

## Syntax

```
<Kleinerer Wert>=MINVAL(<Variable1>,<Variable2>)
<Größerer Wert>=MAXVAL(<Variable1>,<Variable2>)
<Rückgabewert>=<BOUND>(<Minimum>,<Maximum>,<Prüfvariable>)
```

## Bedeutung

| | |
|---|---|
| `MINVAL`: | Ermittelt den **kleineren** Wert zweier Variablen (`<Variable1>`, `<Variable2>`) |
| `<Kleinerer Wert>`: | Ergebnisvariable für den Befehl `MINVAL`<br>Wird auf den kleineren Variablenwert gesetzt. |
| `MAXVAL`: | Ermittelt den **größeren** Wert zweier Variablen (`<Variable1>`, `<Variable2>`) |
| `<Größerer Wert>`: | Ergebnisvariable für den Befehl `MAXVAL`<br>Wird auf den größeren Variablenwert gesetzt. |
| `BOUND`: | Prüft, ob eine Variable (`<Prüfvariable>`) innerhalb eines definierten Wertebereichs liegt. |
| `<Minimum>`: | Variable, die den Minimalwert des Wertebereichs definiert |

| | |
|---|---|
| <Maximum>: | Variable, die den Maximalwert des Wertebereichs definiert |
| <Rückgabewert>: | Ergebnisvariable für den Befehl BOUND<br>Wenn der Wert der Prüfvariablen innerhalb des definierten Wertebereichs liegt, dann wird die Ergebnisvariable auf den Wert der Prüfvariablen gesetzt.<br>Wenn der Wert der Prüfvariablen größer als der Maximalwert ist, dann wird die Ergebnisvariable auf den Maximalwert des Definitionsbereichs gesetzt.<br>Wenn der Wert der Prüfvariablen kleiner als der Minimalwert ist, dann wird die Ergebnisvariable auf den Minimalwert des Definitionsbereichs gesetzt. |

**Hinweis**

MINVAL, MAXVAL und BOUND können auch in Synchronaktionen programmiert werden.

**Hinweis**

**Verhalten bei Gleichheit**

Bei Gleichheit wird bei MINVAL/MAXVAL dieser gleiche Wert geliefert. Bei BOUND wird der Wert der zu prüfenden Variablen wieder zurückgegeben.

## Beispiel

```
Programmcode                                    Kommentar
DEF REAL rVar1=10.5, rVar2=33.7, rVar3, rVar4, rVar5, rValMin, rValMax, rRetVar
rValMin=MINVAL(rVar1,rVar2)                     ; rValMin wird auf den Wert 10.5 gesetzt.
rValMax=MAXVAL(rVar1,rVar2)                     ; rValMax wird auf den Wert 33.7 gesetzt.
rVar3=19.7
rRetVar=BOUND(rVar1,rVar2,rVar3)                ; rVar3 liegt innerhalb der Grenzen, rRetVar wird auf
                                                19.7 gesetzt.
rVar3=1.8
rRetVar=BOUND(rVar1,rVar2,rVar3)                ; rVar3 liegt unterhalb der Minimumgrenze, rRetVar wird
                                                auf 10.5 gesetzt.
rVar3=45.2
rRetVar=BOUND(rVar1,rVar2,rVar3)                ; rVar3 liegt oberhalb der Maximumgrenze, rRetVar wird
                                                auf 33.7 gesetzt.
```

### 3.1.1.15 Vorhandensein einer Variablen prüfen (ISVAR)

Mit der vordefinierten Funktion ISVAR kann geprüft werden, ob eine System-/ Anwendervariable (z. B. Maschinendatum, Settingdatum, Systemvariable, allgemeine Variable wie GUD) in der NC bekannt ist.

### Variable

Die abzufragende Variable folgendermaßen aufgebaut sein:

| | |
|---|---|
| Dimensionslose Variable: | <Variable> |
| Eindimensionale Variable ohne Feldindex: | <Variable>[ ] |
| Eindimensionale Variable mit Feldindex n: | <Variable>[<n>] |
| Zweidimensionale Variable ohne Feldindex: | <Variable>[ , ] |
| Zweidimensionale Variable mit Feldindizes n und m: | <Variable>[<n>,<m>] |

## Syntax

```
<Ergebnis>=ISVAR(<Variable>[<n>,<m>])
```

## Bedeutung

| | | | |
|---|---|---|---|
| <Ergebnis>: | Rückgabewert | | |
| | Datentyp: | BOOL | |
| | Wertebereich: | 1 | Variable vorhanden |
| | | 0 | Variable unbekannt |
| ISVAR: | Prüft, ob die angegebene System-/Anwendervariable in der NC bekannt ist. | | |
| <Variable>: | Name der System- / Anwendervariablen | | |
| | Datentyp: | STRING | |
| <n>: | Feldindex der ersten Dimension (**optional**) | | |
| | Datentyp: | INT | |
| <m>: | Feldindex der zweiten Dimension (**optional**) | | |
| | Datentyp: | INT | |

Entsprechend dem Übergabeparameter werden folgende Prüfungen durchgeführt:

- Ist der Name bekannt
- Ist die Variable ein Feld
- Ist es ein ein- oder zweidimensionales Feld
- Ist der jeweilige Feldindex im erlaubten Bereich

Wenn alle Prüfungen positiv sind, wird TRUE (1) zurückgeliefert.

Ist eine Prüfung negativ oder ist ein Syntaxfehler aufgetreten, wird FALSE (0) zurückgeliefert.

## Beispiele

| **Programmcode** | **Kommentar** |
|---|---|
| `DEF INT VAR1` | |
| `DEF BOOL IS_VAR=FALSE` | |
| `N10 IS_VAR=ISVAR("VAR1")` | `; IS_VAR ist in diesem Fall TRUE.` |

| Programmcode | Kommentar |
|---|---|
| DEF REAL VARARRAY[10,10] | |
| DEF BOOL IS_VAR=FALSE | |
| N10 IS_VAR=ISVAR("VARARRAY[,]") | ; IS_VAR ist in diesem Fall TRUE, ist ein zweidimensionales Feld. |
| N20 IS_VAR=ISVAR("VARARRAY") | ; IS_VAR ist TRUE, Variable existiert. |
| N30 IS_VAR=ISVAR("VARARRAY[8,11]") | ; IS_VAR ist FALSE, Feldindex ist nicht erlaubt. |
| N40 IS_VAR=ISVAR("VARARRAY[8,8") | ; IS_VAR ist FALSE, "]" fehlt (Syntaxfehler). |
| N50 IS_VAR=ISVAR("VARARRAY[,8]") | ; IS_VAR ist TRUE, Feldindex ist erlaubt. |
| N60 IS_VAR=ISVAR("VARARRAY[8,]") | ; IS_VAR ist TRUE, Feldindex ist erlaubt. |

| Programmcode | Kommentar |
|---|---|
| DEF BOOL IS_VAR=FALSE | |
| N100 IS_VAR=ISVAR("$MC_GCODE_RESET_VALUES[1]" | ; Übergabeparameter ist ein Maschinendatum, IS_VAR ist TRUE. |

| Programmcode | Kommentar |
|---|---|
| DEF BOOL IS_VAR=FALSE | |
| N10 IS_VAR=ISVAR("$P_EP") | ; IS_VAR ist in diesem Fall TRUE. |
| N20 IS_VAR=ISVAR("$P_EP[X]") | ; IS_VAR ist in diesem Fall TRUE. |

### 3.1.1.16 Attributwerte/Datentyp lesen (GETVARPHU, GETVARAP, GETVARLIM, GETVARDIM, GETVARDFT, GETVARTYP)

Mit den vordefinierten Funktionen GETVARPHU, GETVARAP, GETVARLIM, GETVARDIM und GETVARDFT können die Attributwerte von System-/Anwendervariablen gelesen werden, mit GETVARTYP der Datentyp einer System-/Anwendervariablen.

## Physikalische Einheit lesen

**Syntax:**
<Ergebnis>=GETVARPHU(<Name>)

**Bedeutung:**

| | | | |
|---|---|---|---|
| <Ergebnis>: | Zahlenwert der physikalischen Einheit | | |
| | Datentyp: | INT | |
| | Wertebereich: | siehe Tabelle in "Attribut: Physikalische Einheit (PHU) (Seite 415)" | |
| | | Im Fehlerfall: | |
| | | - 2 | Der angegebene Variablenname ist keinem Systemparameter und keiner Anwendervariablen zugeordnet. |
| GETVARPHU: | Lesen der physikalischen Einheit einer System- /Anwendervariablen | | |

| | | |
|---|---|---|
| <Name>: | Name der System- /Anwendervariablen | |
| | Datentyp: | STRING |

**Beispiel:**

Der NC enthalte folgende GUD-Variable:

DEF CHAN REAL PHU 42 LLI 0 ULI 10000 electric

```
Programmcode                                    Kommentar
DEF INT result=0
result=GETVARPHU("electric")                    ; Ermittle die physikalische Einheit der GUD-
                                                Variablen.
IF (result < 0) GOTOF error
```

Als Ergebnis wird der Wert 42 zurückgeliefert. Dies entspricht der physikalischen Einheit [kW].

**Hinweis**

Mit GETVARPHU lässt sich z. B. prüfen, ob bei einer Variablenzuweisung a = b beide Variablen die erwarteten physikalischen Einheiten haben.

## Zugriffsrecht lesen

**Syntax:**
<Ergebnis>=GETVARAP(<Name>,<Zugriff>)

**Bedeutung:**

| | | | |
|---|---|---|---|
| <Ergebnis>: | Schutzstufe für den angegebenen <Zugriff> | | |
| | Datentyp: | INT | |
| | Wertebereich: | 0 ... 7 | Siehe "Attribut: Zugriffsrechte (APR, APW, APRP, APWP, APRB, APWB) (Seite 417)". |
| | | Im Fehlerfall: | |
| | | - 1 | nicht schreibbar (nur bei den Zugriffsarten "WP" und "WB" relevant) |
| | | - 2 | Der angegebene Variablenname ist keinem Systemparameter und keiner Anwendervariablen zugeordnet. |
| | | - 3 | falscher Wert für den Parameter <Zugriff> |
| GETVARAP: | Lesen des Zugriffsrechts auf eine System- /Anwendervariable | | |
| <Name>: | Name der System- /Anwendervariablen | | |
| | Datentyp: | STRING | |

| | | | |
|---|---|---|---|
| <Zugriff>: | Art des Zugriffs | | |
| | Datentyp: | STRING | |
| | Wertebereich: | "RP" | Lesen über Teileprogramm |
| | | "WP" | Schreiben über Teileprogramm |
| | | "RB" | Lesen über BTSS |
| | | "WB" | Schreiben über BTSS |

**Beispiel:**

| **Programmcode** | **Kommentar** |
|---|---|
| DEF INT result=0 | |
| result=GETVARAP("$TC_MAP8","WB") | ; Ermittle den Zugriffschutz für den Systemparameter "Magazinposition" bezüglich Schreiben über BTSS. |
| IF (result < 0) GOTOF error | |

Als Ergebnis wird der Wert 7 zurückgeliefert. Dies entspricht der Schlüsselschalter-Stellung 0 (= kein Zugriffsschutz).

**Hinweis**

Mit GETVARAP kann z. B. ein Prüfprogramm realisiert werden, das die von der Applikation erwarteten Zugriffsrechte prüft.

## Grenzwerte lesen

**Syntax:**
<Status>=GETVARLIM(<Name>,<Grenzwert>,<Ergebnis>)

**Bedeutung:**

| | | | |
|---|---|---|---|
| <Status>: | Funktionsstatus | | |
| | Datentyp: | INT | |
| | Wertebereich: | 1 | o. k. |
| | | -1 | kein Grenzwert definiert<br>(bei Variablen vom Typ AXIS, STRING, FRAME) |
| | | -2 | Der angegebene Variablenname ist keinem Systemparameter und keiner Anwendervariablen zugeordnet. |
| | | -3 | falscher Wert für den Parameter <Grenzwert> |
| GETVARLIM: | Lesen des unteren/oberen Grenzwerts einer System- /Anwendervariablen | | |
| <Name>: | Name der System- /Anwendervariablen | | |
| | Datentyp: | STRING | |
| <Grenzwert>: | Gibt an, welcher Grenzwert ausgelesen werden soll | | |
| | Datentyp: | CHAR | |
| | Wertebereich: | "L" | unterer Grenzwert |
| | | "U" | oberer Grenzwert |

| | | |
|---|---|---|
| <Ergebnis>: | Rückgabe des Grenzwerts | |
| | Datentyp: | VAR REAL |

### Beispiel:

| Programmcode | Kommentar |
|---|---|
| DEF INT state=0 | |
| DEF REAL result=0 | |
| state=GETVARLIM("$MA_MAX_AX_VELO","L",result) | ; Ermittle den unteren Grenzwert für MD32000 $MA_MAX_AX_VELO. |
| IF (result < 0) GOTOF error | |

## Dimensionswert lesen

### Syntax:
<Ergebnis>=GETVARDIM(<Name>,<Index>)

### Bedeutung:

| | | | |
|---|---|---|---|
| <Ergebnis>: | Größe der durch den Parameter <Index> angegebenen Dimension | | |
| | Datentyp: | INT | |
| GETVARDIM: | Lesen der Größe der 1., 2. oder 3. Dimension des Feldes einer System- /Anwendervariablen | | |
| <Name>: | Name der System- /Anwendervariablen | | |
| | Datentyp: | STRING | |
| <Index>: | Feldindex | | |
| | Datentyp: | INT | |
| | Wertebereich: | 1 ... 3 | |
| | | 1 | Index für 1. Dimension des Feldes |
| | | 2 | Index für 2. Dimension des Feldes |
| | | 3 | Index für 3. Dimension des Feldes |

### Beispiel:

| Programmcode | Kommentar |
|---|---|
| N5 DEF REAL myReal[5,4] | |
| N10 R1=GETVARDIM("myReal",1) | ; Ermittle die Größe der 1. Dimension des Feldes. |
| | ; Ergebnis: R1 = 5 |
| N15 R2=GETVARDIM("myReal",2) | ; Ermittle die Größe der 2. Dimension des Feldes. |
| | ; Ergebnis: R2 = 4 |

## Standardwert lesen

### Syntax:
<Status>=GETVARDFT(<Name>,<Ergebnis>[,<Index_1>,<Index_2>,<Index_3>])

### Bedeutung:

| | | | |
|---|---|---|---|
| <Status>: | Funktionsstatus | | |
| | Datentyp: | INT | |
| | Wertebereich: | 1 | o. k. |
| | | -1 | kein Standardwert verfügbar<br>(z. B. weil die Datentyp-Definition der Ergebnisvariablen nicht zum Datentyp der System-/Anwendervariablen passt) |
| | | -2 | Der angegebene Variablenname ist keinem Systemparameter und keiner Anwendervariablen zugeordnet. |
| | | -3 | falscher Wert für Parameter <Index_1>, Dimension kleiner eins (⇒ kein Feld, sondern skalare Variable) |
| | | -4 | falscher Wert für Parameter <Index_2> |
| | | -5 | falscher Wert für Parameter <Index_3> |
| GETVARDFT: | Lesen des Standardwerts einer System- /Anwendervariablen | | |
| <Name>: | Name der System-/Anwendervariablen | | |
| | Datentyp: | STRING | |
| <Ergebnis>: | Rückgabe des Standardwerts | | |
| | Datentyp: | VAR REAL<br>(beim Lesen des Standardwerts von Variablen der Typen INT, REAL, BOOL, AXIS) | |
| | | VAR STRING<br>(beim Lesen des Standardwerts von Variablen der Typen STRING und CHAR) | |
| | | VAR FRAME<br>(beim Lesen des Standardwerts von Variablen des Typs FRAME) | |
| <Index_1>: | Index für 1. Dimension des Feldes (optional) | | |
| | Datentyp: | INT | |
| | Nicht programmiert bedeutet = 0 | | |
| <Index_2>: | Index für 2. Dimension des Feldes (optional) | | |
| | Datentyp: | INT | |
| | Nicht programmiert bedeutet = 0 | | |
| <Index_3>: | Index für 3. Dimension des Feldes (optional) | | |
| | Datentyp: | INT | |
| | Nicht programmiert bedeutet = 0 | | |

### Beispiel:

```
Programmcode                                                    Kommentar
DEF INT state=0
DEF REAL resultR=0                                              ; Variable zur Aufnahme der Standardwerte
                                                                der Typen INT, REAL, BOOL, AXIS.
DEF FRAME resultF=0                                             ; Variable zur Aufnahme der Standardwerte
                                                                des Typs FRAME


IF (GETVARTYP("$MA_MAX_AX_VELO") <> 4) GOTOF error


state=GETVARDFT("$MA_MAX_AX_VELO",resultR, AXTOINT(X))          ; Ermittle den Standardwert der "X"-Achse.
```

| Programmcode | Kommentar |
|---|---|
| IF (resultR < 0) GOTOF error | |
| | |
| IF (GETVARTYP("$TC_TP8") <> 3) GOTOF error | |
| state=GETVARDFT("$TC_TP8", resultR) | |
| | |
| IF (GETVARTYP("$P_UBFR") <> 7) GOTOF error | |
| state=GETVARDFT("$P_UBFR", resultF ) | |

## Datentyp lesen

**Syntax:**
<Ergebnis>=GETVARTYP(<Name>)

**Bedeutung:**

| | | | |
|---|---|---|---|
| <Ergebnis>: | Datentyp der angegebenen System- /Anwendervariablen | | |
| | Datentyp: | INT | |
| | Wertebereich: | 1 | = BOOL |
| | | 2 | = CHAR |
| | | 3 | = INT |
| | | 4 | = REAL |
| | | 5 | = STRING |
| | | 6 | = AXIS |
| | | 7 | = FRAME |
| | | Im Fehlerfall: | |
| | | < 0 | Der angegebene Variablenname ist keinem Systemparameter und keiner Anwendervariablen zugeordnet. |
| GETVARTYP: | Lesen des Datentyps einer System- /Anwendervariablen | | |
| <Name>: | Name der System- /Anwendervariablen | | |
| | Datentyp: | STRING | |

**Beispiel:**

| Programmcode | Kommentar |
|---|---|
| DEF INT result=0 | |
| DEF STRING name="R" | |
| result=GETVARTYP(name) | ; Ermittle den Typ des R-Parameters. |
| IF (result < 0) GOTOF error | |

Als Ergebnis wird der Wert 4 zurückgeliefert. Dies entspricht dem Datentyp REAL.

### 3.1.1.17 Mögliche Typenkonvertierungen

Der konstante Zahlenwert, die Variable oder der Ausdruck, der einer Variablen zugewiesen wird, muss mit dem Typ dieser Variablen verträglich sein. Ist dies gegeben, so wird bei der Zuweisung der Typ automatisch umgewandelt.

### Mögliche Typkonvertierungen

| nach | REAL | INT | BOOL | CHAR | STRING | AXIS | FRAME |
|---|---|---|---|---|---|---|---|
| von | | | | | | | |
| REAL | ja | ja* | ja[1]) | ja* | – | – | – |
| INT | ja | ja | ja[1]) | ja[2]) | – | – | – |
| BOOL | ja | ja | ja | ja | ja | – | – |
| CHAR | ja | ja | ja[1]) | ja | ja | – | – |
| STRING | – | – | ja[4]) | ja[3]) | ja | – | – |
| AXIS | – | – | – | – | – | ja | – |
| FRAME | – | – | – | – | – | – | ja |

**Erklärungen**

* Bei Typumwandlung von REAL nach INT wird bei gebrochenem Wert >= 0.5 aufgerundet, ansonsten wird abgerundet (vgl. Funktion ROUND)

1) Wert <> 0 entspricht TRUE, Wert == 0 entspricht FALSE

2) Wenn der Wert im zulässigen Zahlenbereich liegt

3) Wenn nur 1 Zeichen

4) Stringlänge 0 = >FALSE, ansonsten TRUE

**Hinweis**

Ist beim Konvertieren ein Wert größer als der Zielbereich, erfolgt eine Fehlermeldung.

Treten in einem Ausdruck gemischte Typen auf, so wird eine Typanpassung automatisch durchgeführt. Typumwandlungen sind auch in Synchronaktionen möglich, siehe Kapitel Bewegungssynchronaktionen, Implizite Typwandlung".

## 3.1.2 Indirekte Programmierung

### 3.1.2.1 Indirekte Programmierung von Adressen

Bei der indirekten Programmierung von Adressen (Seite 1255) wird die erweiterte Adresse (<Index>) durch eine Variable geeigneten Typs ersetzt.

**Hinweis**

Die indirekte Programmierung von Adressen ist nicht möglich bei:

- N (Satznummer)
- L (Unterprogramm)
- Einstellbaren Adressen
  (z. B. X[1] anstelle von X1 ist nicht zulässig)

## Syntax

```
<ADRESSE>[<Index>]
```

## Bedeutung

| | |
|---|---|
| <ADRESSE>[...]: | Feste Adresse mit Erweiterung (Index) |
| <Index>: | Variable z. B. für Spindelnummer, Achse, ... |

## Beispiele

### Beispiel 1: Indirekte Programmierung einer Spindelnummer

Direkte Programmierung:

| Programmcode | Kommentar |
|---|---|
| S1=300 | ; Drehzahl 300 U/min für die Spindel mit Nummer 1. |

Indirekte Programmierung:

| Programmcode | Kommentar |
|---|---|
| DEF INT SPINU=1 | ; Definition der Variablen vom Typ INT und Wertzuweisung. |
| S[SPINU]=300 | ; Drehzahl 300 U/min für die Spindel, deren Nummer in der Variablen SPINU abgelegt ist (in diesem Beispiel die Spindel mit Nummer 1). |

### Beispiel 2: Indirekte Programmierung einer Achse

Direkte Programmierung:

| Programmcode | Kommentar |
|---|---|
| FA[U]=300 | ; Vorschub 300 für die Achse "U". |

Indirekte Programmierung:

| Programmcode | Kommentar |
|---|---|
| DEF AXIS AXVAR2=U | ; Definition einer Variablen vom Typ AXIS und Wertzuweisung. |
| FA[AXVAR2]=300 | ; Vorschub 300 für die Achse, deren Adressname in der Variablen mit dem Namen AXVAR2 abgelegt ist. |

### Beispiel 3: Indirekte Programmierung einer Achse

Direkte Programmierung:

| Programmcode | Kommentar |
|---|---|
| $AA_MM[X] | ; Messtaster-Messwert (MKS) der Achse "X" lesen. |

Indirekte Programmierung:

| Programmcode | Kommentar |
|---|---|
| DEF AXIS AXVAR3=X | ; Definition einer Variablen vom Typ AXIS und Wertzuweisung. |

| Programmcode | Kommentar |
|---|---|
| $AA_MM[AXVAR3] | ; Messtaster-Messwert (MKS) lesen für die Achse, deren Name in der Variablen AXVAR3 abgelegt ist. |

### Beispiel 4: Indirekte Programmierung einer Achse

Direkte Programmierung:

| Programmcode |
|---|
| X1=100 X2=200 |

Indirekte Programmierung:

| Programmcode | Kommentar |
|---|---|
| DEF AXIS AXVAR1 AXVAR2 | ; Definition zweier Variablen vom Typ AXIS. |
| AXVAR1=(X1) AXVAR2=(X2) | ; Zuweisung der Achsnamen. |
| AX[AXVAR1]=100 AX[AXVAR2]=200 | ; Verfahren der Achsen, deren Adressnamen in den Variablen mit den Namen AXVAR1 und AXVAR2 abgelegt sind. |

### Beispiel 5: Indirekte Programmierung einer Achse

Direkte Programmierung:

| Programmcode |
|---|
| G2 X100 I20 |

Indirekte Programmierung:

| Programmcode | Kommentar |
|---|---|
| DEF AXIS AXVAR1=X | ; Definition einer Variablen vom Typ AXIS und Wertzuweisung. |
| G2 X100 IP[AXVAR1]=20 | ; Indirekte Programmierung der Mittelpunktsangabe für die Achse, deren Adressname in der Variablen mit dem Namen AXVAR1 abgelegt ist |

### Beispiel 6: Indirekte Programmierung von Feldelementen

Direkte Programmierung:

| Programmcode | Kommentar |
|---|---|
| DEF INT FELD1[4,5] | ; Definition von Feld 1. |

Indirekte Programmierung:

| Programmcode | Kommentar |
|---|---|
| DEFINE DIM1 AS 4 | ; Bei Felddimensionen müssen Feldgrößen als feste Werte angegeben werden. |
| DEFINE DIM2 AS 5 | |
| DEF INT FELD[DIM1,DIM2] | |
| FELD[DIM1-1,DIM2-1]=5 | |

**Beispiel 7: Indirekter Unterprogrammaufruf**

| Programmcode | Kommentar |
|---|---|
| `CALL "L" << R10` | `; Aufruf des Programms, dessen Nummer in R10 steht (String-verkettung).` |

### 3.1.2.2 Indirekte Programmierung von G-Befehlen

Die indirekte Programmierung von G-Befehlen (Seite 1267) ermöglicht eine effektive Zyklenprogrammierung.

## Syntax

```
G[<Gruppe>]=<Nummer>
```

## Bedeutung

| | | |
|---|---|---|
| `G[...]`: | G-Befehl mit Erweiterung (Index) | |
| `<Gruppe>`: | Index-Parameter: G-Gruppe | |
| | Typ: | INT |
| `<Nummer>`: | Variable für die G-Befehl-Nummer | |
| | Typ: | INT oder REAL |

**Hinweis**

Es können i. d. R. nur nicht-syntaxbestimmende G-Befehle indirekt programmiert werden.

Von den syntaxbestimmenden G-Befehlen sind nur die der G-Gruppe 1 möglich.
Die syntaxbestimmenden G-Befehle der G-Gruppen 2, 3 und 4 sind nicht möglich.

**Hinweis**

In der indirekten G-Befehl-Programmierung sind keine Arithmetik-Funktionen erlaubt. Eine notwendige Berechnung der G-Befehls-Nummer muss in einer eigenen Teileprogrammzeile vor der indirekten G-Befehl-Programmierung erfolgen.

## Beispiele

**Beispiel 1: Einstellbare Nullpunktverschiebung (G-Gruppe 8)**

```
Programmcode                              Kommentar
N1010 DEF INT INT_VAR
N1020 INT_VAR=2
...
N1090 G[8]=INT_VAR G1 X0 Y0               ; G54
N1100 INT_VAR=INT_VAR+1                   ; G-Befehl-Berechnung
N1110 G[8]=INT_VAR G1 X0 Y0               ; G55
```

**Beispiel 2: Ebenenanwahl (G-Gruppe 6)**

| Programmcode | Kommentar |
|---|---|
| N2010 R10=$P_GG[6] | ; Aktiver G-Befehl der G-Gruppe 6 lesen |
| ... | |
| N2090 G[6]=R10 | |

### 3.1.2.3 Indirekte Programmierung von Positionsattributen (GP)

Positionsattribute, wie z. B. die inkrementelle oder absolute Programmierung der Achsposition, können in Verbindung mit dem Schlüsselwort GP indirekt als Variablen programmiert werden.

## Anwendung

Die indirekte Programmierung von Positionsattributen findet Verwendung in **Ersetzungszyklen**, da hier folgender Vorteil gegenüber der Programmierung von Positionsattributen als Schlüsselwort (z. B. IC, AC, ...) besteht:

Durch die indirekte Programmierung als Variablen wird **keine** CASE-Anweisung benötigt, die über alle möglichen Positionsattribute verzweigt.

## Syntax

```
<POSITIONIERBEFEHL>[<Achse/Spindel>]=
GP(<Position>,<Positionsattribut)
<Achse/Spindel>=GP(<Position>,<Positionsattribut)
```

## Bedeutung

| | |
|---|---|
| <POSITIONIERBEFEHL>[]: | Folgende Positionierbefehle können zusammen mit dem Schlüsselwort GP programmiert werden:<br>POS, POSA, SPOS, SPOSA<br>Außerdem möglich:<br>• alle im Kanal vorhandenen Achs-/Spindelbezeichner: <Achse/Spindel><br>• variabler Achs-/Spindelbezeichner AX |
| <Achse/Spindel>: | Achse/Spindel, die positioniert werden soll |
| GP(): | Schlüsselwort zur Positionierung |
| <Position>: | Parameter 1<br>Achs-/Spindelposition als Konstante oder Variable |
| <Positionsattribut>: | Parameter 2<br>Positionsattribut (z. B. Positionsanfahrmodus) als Variable (z. B. $P_SUB_SPOSMODE) oder als Schlüsselwort (IC, AC, ...) |

Die von den Variablen gelieferten Werte haben folgende Bedeutung:

| Wert | Bedeutung | Zulässig bei: |
|---|---|---|
| 0 | Keine Änderung des Positionsattributs | |
| 1 | AC | POS, POSA,SPOS, SPOSA,AX, Achsadresse |
| 2 | IC | POS, POSA,SPOS, SPOSA,AX, Achsadresse |
| 3 | DC | POS, POSA,SPOS, SPOSA,AX, Achsadresse |
| 4 | ACP | POS, POSA,SPOS, SPOSA,AX, Achsadresse |
| 5 | ACN | POS, POSA,SPOS, SPOSA,AX, Achsadresse |
| 6 | OC | - |
| 7 | PC | - |
| 8 | DAC | POS, POSA,AX, Achsadresse |
| 9 | DIC | POS, POSA,AX, Achsadresse |
| 10 | RAC | POS, POSA,AX, Achsadresse |
| 11 | RIC | POS, POSA,AX, Achsadresse |
| 12 | CAC | POS, POSA |
| 13 | CIC | POS, POSA |
| 14 | CDC | POS, POSA |
| 15 | CACP | POS, POSA |
| 16 | CACN | POS, POSA |

## Beispiel

Bei einer aktiven Synchronspindelkopplung zwischen der Leitspindel S1 und der Folgespindel S2 wird durch den `SPOS`-Befehl im Hauptprogramm der folgende Ersetzungszyklus zur Positionierung der Spindeln aufgerufen.

Die Positionierung erfolgt über die Anweisung in N2230:
```
SPOS[1]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE)
SPOS[2]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE)
```

Die anzufahrende Position wird aus der Systemvariablen $P_SUB_SPOSIT, der Positionsanfahrmodus wird aus der Systemvariablen $P_SUB_SPOSMODE gelesen.

| Programmcode | Kommentar |
|---|---|
| `N1000 PROC LANG_SUB DISPLOF SBLOF` | |
| `...` | |
| `N2100 IF($P_SUB_AXFCT==2)` | |
| `N2110` | `; Ersetzung des SPOS / SPOSA / M19-Befehls bei aktiver Synchronspindelkopplung` |
| `N2185 DELAYFSTON` | `; Beginn Stopp-Delay-Bereich` |
| `N2190 COUPOF(S2,S1)` | `; Synchronspindelkopplung deaktivieren` |
| `N2200` | `; Leit- und Folgespindel positionieren` |
| `N2210 IF($P_SUB_SPOS==TRUE) OR ($P_SUB_SPOSA==TRUE)` | |
| `N2220` | `; Spindel mit SPOS positionieren:` |
| `N2230 SPOS[1]=GP($P_SUB_SPOSIT, $P_SUB_SPOSMODE)` | |

| Programmcode | Kommentar |
|---|---|
| SPOS[2]=GP($P_SUB_SPOSIT, $P_SUB_SPOSMODE) | |
| N2250 ELSE | |
| N2260 | ; Spindel mit M19 positionieren: |
| N2270 M1=19 M2=19 | ; Leit- und Folgespindel positionieren |
| N2280 ENDIF | |
| N2285 DELAYFSTOF | ; Ende Stopp-Delay-Bereich |
| N2290 COUPON(S2,S1) | ; Synchronspindelkopplung aktivieren |
| N2410 ELSE | |
| N2420 | ; Abfrage auf weitere Ersetzungen |
| ... | |
| N3300 ENDIF | |
| ... | |
| N9999 RET | |

## Randbedingungen

In Synchronaktionen ist die indirekte Programmierung von Positionsattributen nicht möglich.

### 3.1.2.4 Indirekte Programmierung von Teileprogrammzeilen (EXECSTRING)

Mit dem Teileprogrammbefehl `EXECSTRING` ist es möglich, eine zuvor erzeugte String-Variable als Teileprogrammzeile auszuführen.

## Syntax

`EXECSTRING` wird in einer eigenen Teileprogrammzeile programmiert:
`EXECSTRING(<String-Variable>)`

## Bedeutung

| | |
|---|---|
| `EXECSTRING`: | Befehl zur Ausführung einer String-Variablen als Teileprogrammzeile |
| `<String-Variable>`: | Variable vom Typ STRING, die die eigentlich auszuführende Teileprogrammzeile enthält |

### Hinweis

Mit `EXECSTRING` können mit Ausnahme von Kontrollstrukturen (Seite 480) alle Teileprogramm-Konstrukte abgesetzt werden, die **im Programmteil eines Teileprogramms** programmiert werden können. Ausgeschlossen sind damit `PROC`- und `DEF`-Anweisungen sowie generell die Verwendung in INI- und DEF-Dateien.

### Beispiel

| Programmcode | Kommentar |
|---|---|
| N100 DEF STRING[100] MY_BLOCK | ; Definition der String-Variablen zur Aufnahme der auszuführenden Teileprogrammzeile. |
| N110 DEF STRING[10] MFCT1="M7" | |
| ... | |
| N200 EXECSTRING(MFCT1 << "M4711") | ; Teileprogrammzeile "M7 M4711" ausführen. |
| ... | |
| N300 R10=1 | |
| N310 MY_BLOCK="M3" | |
| N320 IF(R10) | |
| N330 MY_BLOCK = MY_BLOCK << MFCT1 | |
| N340 ENDIF | |
| N350 EXECSTRING(MY_BLOCK) | ; Teileprogrammzeile "M3 M7" ausführen. |

## 3.1.3 Operationen

### 3.1.3.1 Rechenfunktionen

| Operator / Rechenfunktion | Bedeutung |
|---|---|
| + | Addition |
| - | Subtraktion |
| * | Multiplikation |
| / [1] | Division [1] |
| DIV [1] | Ganzzahl-Division [1] |
| MOD [1] | Modulo-Division (Liefert den Rest der Ganzzahl-Division) [1] |
| : | Kettungsoperator für FRAME-Variablen |
| SIN() | Sinus |
| COS() | Cosinus |
| TAN() | Tangens |
| ASIN() | Arcussinus |
| ACOS() | Arcuscosinus |
| ATAN2(,) [1] | Arcustangens2 [1] |
| SQRT() | Quadratwurzel |
| ABS() | Betrag |
| POT() | 2. Potenz (Quadrat) |
| TRUNC() | ganzzahliger Teil<br>Genauigkeiten bei Vergleichsbefehlen einstellbar mit TRUNC (siehe "Genauigkeitskorrektur bei Vergleichsfehlern (TRUNC) (Seite 450)") |
| ROUND() | Runden auf ein Ganzzahliges |

| | |
|---|---|
| `LN()` | natürlicher Logarithmus |
| `EXP()` | Exponentialfunktion |
| `MINVAL()` | kleinerer Wert zweier Variablen<br>(siehe "Minimum, Maximum und Bereich von Variablen (MINVAL, MAXVAL, BOUND) (Seite 429)" ) |
| `MAXVAL()` | größerer Wert zweier Variablen<br>(siehe "Minimum, Maximum und Bereich von Variablen (MINVAL, MAXVAL, BOUND) (Seite 429)") |
| `BOUND()` | Variablenwert, der im definierten Wertebereich liegt<br>(siehe "Minimum, Maximum und Bereich von Variablen (MINVAL, MAXVAL, BOUND) (Seite 429)") |
| `CTRANS()` | Verschiebung |
| `CROT()` | Drehung |
| `CSCALE()` | Maßstabsveränderung |
| `CMIRROR()` | Spiegeln |
| 1) Siehe Absatz "Beispiele" | |

## Programmierung

Bei den Rechenfunktionen gilt die übliche mathematische Schreibweise. Prioritäten in der Abarbeitung werden durch runde Klammern gesetzt. Für die trigonometrischen und deren inverse Funktionen gilt die Gradangabe (rechter Winkel = 90°).

## Beispiele

### Division: /

(Typ REAL) = Typ INT oder Typ REAL) / (Typ INT oder Typ REAL);

Beispiel: 3 / 4 = 0.75

### Ganzzahl-Division: DIV

(Typ INT) = (Typ INT oder REAL) / (Typ INT oder REAL);

Beispiel: 7 DIV 4.1 = 1

### Modulo-Division (Liefert den Rest der Ganzzahl-Division): MOD

(Typ REAL) = (Typ INT oder REAL) MOD (Typ INT oder REAL);

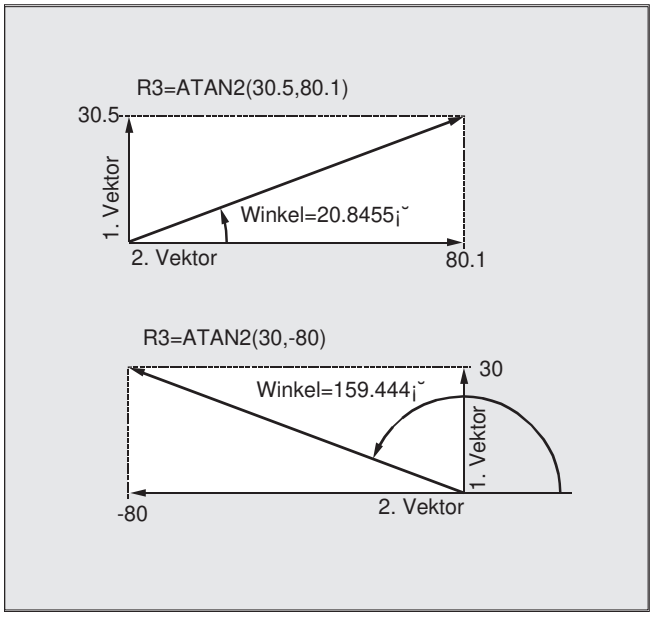
Beispiel: 7 MOD 4.1 = 2.9

### Arcustangens2: ATAN2

Die Rechenfunktion ATAN2 berechnet aus zwei aufeinander senkrecht stehenden Vektoren den Winkel des Summenvektors.

Das Ergebnis liegt im Bereich von vier Quadranten (-180° < 0 < +180°).

Basis für den Winkelbezug ist immer der 2. Wert in positiver Richtung.




**Programmierbeispiele**

| **Programmcode** | **Kommentar** |
|---|---|
| `R1=R1+1` | `; Neues R1 = altes R1 + 1` |
| `R1=R2+R3 R4=R5-R6 R7=R8*R9` | |
| `R10=R11/R12 R13=SIN(25.3)` | |
| `R14=R1*R2+R3` | `; Punktrechnung geht vor Strichrechnung.` |
| `R14=(R1+R2)*R3` | `; Geklammerte Ausdrücke werden zuerst berechnet.` |
| `R15=SQRT(POT(R1)+POT(R2))` | `; Innere Klammern werden zuerst aufgelöst:` |
| | `R15 = Quadratwurzel( (R1^2 + R2^2) )` |
| `RESFRAME=FRAME1:FRAME2` | `; FRAME-Verknüpfung mit Kettungsoperator` |
| `FRAME3=CTRANS(…):CROT(…)` | `Wertzuweisung an eine FRAME-Komponente` |

### 3.1.3.2 Vergleichs- und logische Operationen

**Vergleichsoperationen** können z. B. zur Formulierung einer Sprungbedingung benutzt werden. Vergleichbar sind dabei auch komplexe Ausdrücke.

Die Vergleichsoperationen sind für Variable vom Typ `CHAR, INT, REAL` und `BOOL` anwendbar. Beim Typ `CHAR` wird der Codewert verglichen.
Bei den Typen `STRING`, `AXIS` und `FRAME` sind möglich: **==** und **<>**, die für Operationen vom Typ `STRING` auch in Synchronaktionen angewendet werden können.

Das Ergebnis von vergleichenden Operationen ist immer vom Typ `BOOL`.

**Logische Operatoren** dienen zur Verknüpfung von Wahrheitswerten.

Die logischen Operationen sind nur auf Variable vom Typ `BOOL` anwendbar. Über interne Typenkonvertierung sind sie auch auf die Datentypen `CHAR, INT`, und `REAL` anwendbar.

Bei den logischen (boolschen) Operationen gilt für die Datentypen BOOL, CHAR, INT und REAL:

- 0 entspricht: FALSE
- ungleich 0 entspricht: TRUE

**Bitweise logische Operatoren**

Mit den Variablen vom Typ CHAR und INT können auch bitweise logische Operationen vorgenommen werden. Gegebenenfalls erfolgt eine Typkonvertierung automatisch.

## Programmierung

| Vergleichsoperator | Bedeutung |
|---|---|
| == | gleich |
| <> | ungleich |
| > | größer |
| < | kleiner |
| >= | größer oder gleich |
| <= | kleiner oder gleich |

| Logischer Operator | Bedeutung |
|---|---|
| AND | UND |
| OR | ODER |
| NOT | Negation |
| XOR | Exklusiv-ODER |

| Bitweise logischer Operator | Bedeutung |
|---|---|
| B_AND | bitweises UND |
| B_OR | bitweises ODER |
| B_NOT | bitweise Negation |
| B_XOR | bitweises Exklusiv-ODER |

**Hinweis**

In arithmetischen Ausdrücken kann durch runde Klammern die Abarbeitungsreihenfolge aller Operatoren festgelegt und damit von den normalen Prioritätsregeln abgewichen werden.

**Hinweis**

Zwischen BOOLSCHEN Operanden und Operatoren müssen Zwischenräume geschrieben werden.

**Hinweis**

Der Operator B_NOT bezieht sich auf nur einen Operanden. Dieser steht nach dem Operator.

## Beispiele

**Beispiel 1: Vergleichsoperatoren**

```
IF R10>=100 GOTOF ZIEL
```

oder

```
R11=R10>=100
IF R11 GOTOF ZIEL
```

Das Ergebnis des Vergleichs `R10>=100` wird zunächst in `R11` zwischengespeichert.

**Beispiel 2: Logische Operatoren**

```
IF (R10<50) AND ($AA_IM[X]>=17.5) GOTOF ZIEL
```

oder

```
IF NOT R10 GOTOB START
```

`NOT` bezieht sich nur auf einen Operanden.

**Beispiel 3: Bitweise logische Operatoren**

```
IF $MC_RESET_MODE_MASK B_AND 'B10000' GOTOF ACT_PLANE
```

### 3.1.3.3 Priorität der Operationen

Jedem Operator ist eine Priorität zugeordnet. Bei der Auswertung eines Ausdrucks werden stets die Operatoren höherer Priorität zuerst angewandt. Bei gleichrangigen Operatoren erfolgt die Auswertung von links nach rechts.

In arithmetischen Ausdrücken kann durch runde Klammern die Abarbeitungsreihenfolge aller Operatoren festgelegt und damit von den normalen Prioritätsregeln abgewichen werden.

## Reihenfolge der Operatoren

**Von der höchsten zur niedrigsten Priorität**

| | | |
|---|---|---|
| 1. | NOT, B_NOT | Verneinung, bitweise Verneinung |
| 2. | *, /, DIV, MOD | Multiplikation, Division |
| 3. | +, – | Addition, Subtraktion |
| 4. | B_AND | bitweises UND |
| 5. | B_XOR | bitweises exklusives ODER |
| 6. | B_OR | bitweises ODER |
| 7. | AND | UND |
| 8. | XOR | exklusives ODER |
| 9. | OR | ODER |
| 10. | << | Verkettung von Strings, Ergebnistyp STRING |
| 11. | ==, <>, >, <, >=, <= | Vergleichsoperatoren |

**Hinweis**

Der Kettungsoperator ":" für Frames darf nicht mit anderen Operatoren in einem Ausdruck vorkommen. Eine Prioritätseinstufung für diesen Operator ist deshalb nicht erforderlich.

### Beispiel If-Anweisung

```
If (otto==10) and (anna==20) gotof end
```

## 3.1.3.4 Genauigkeitskorrektur bei Vergleichsfehlern (TRUNC)

Der TRUNC-Befehl schneidet den mit einem Genauigkeitsfaktor multiplizierten Operanden ab.

#### Einstellbare Genauigkeit bei Vergleichsbefehlen

Teileprogrammdaten vom Typ REAL werden intern im IEEE-Format mit 64 Bit dargestellt. Aufgrund dieser Darstellungsform können Dezimalzahlen ungenau abgebildet werden, die bei einem Vergleich mit ideal gerechneten Werten zu unerwarteten Ergebnissen führen können.

#### Relative Gleichheit

Damit die durch die Darstellungsform hervorgerufenen Ungenauigkeiten den Programmfluß nicht verfälschen, wird bei den Vergleichsbefehlen nicht auf absolute Gleichheit, sondern auf eine relative Gleichheit geprüft.

### Syntax

#### Genauigkeitskorrektur bei Vergleichsfehlern

```
TRUNC (R1*1000)
```

### Bedeutung

| | |
|---|---|
| `TRUNC`: | Abschneiden der Nachkommastellen |

#### Berücksichtigte relative Gleichheit von $10^{-12}$ bei

- Gleichheit: (==)
- Ungleichheit: (<>)
- Größer-Gleich: (>=)
- Kleiner-Gleich: (<=)
- Größer/Kleiner: (><) mit absoluter Gleichheit
- Größer: (>)
- Kleiner: (<)

### Kompatibilität

Aus Kompatibilitätsgründen kann die Prüfung auf relative Gleichheit bei (>) und (<) durch Setzen von Maschinendatum MD10280 $MN_ PROG_FUNCTION_MASK Bit0 = 1 deaktiviert werden.

### Hinweis

Vergleiche mit Daten vom Typ REAL sind aus den genannten Gründen generell mit einer gewissen Ungenauigkeit behaftet. Bei nicht akzeptablen Abweichungen muss auf INTEGER-Rechnung ausgewichen werden, indem die Operanden mit einem Genauigkeitsfaktor multipliziert und danach mit TRUNC abgeschnitten werden.

### Synchronaktionen

Das beschriebene Verhalten der Vergleichsbefehle gilt auch bei Synchronaktionen.

## Beispiele

### Beispiel 1: Genauigkeitsbetrachtungen

| Programmcode | Kommentar |
|---|---|
| N40 R1=61.01 R2=61.02 R3=0.01 | ; Zuweisung der Anfangswerte |
| N41 IF ABS(R2-R1) > R3 GOTOF FEHLER | ; Sprung würde bisher ausgeführt werden |
| N42 M30 | ; Programmende |
| N43 FEHLER: SETAL(66000) | |
| R1=61.01 R2=61.02 R3=0.01 | ; Zuweisung der Anfangswerte |
| R11=TRUNC(R1*1000) R12=TRUNC(R2*1000) R13=TRUNC(R3*1000) | ; Genauigkeitskorrektur |
| IF ABS(R12-R11) > R13 GOTOF FEHLER | ; Sprung wird nicht mehr ausgeführt |
| M30 | ; Programmende |
| FEHLER: SETAL(66000) | |

### Beispiel 2: Quotient beider Operanden bilden und auswerten

| Programmcode | Kommentar |
|---|---|
| R1=61.01 R2=61.02 R3=0.01 | ; Zuweisung der Anfangswerte |
| IF ABS((R2-R1)/R3)-1) > 10EX-5 GOTOF FEHLER | ; Sprung wird nicht ausgeführt |
| M30 | ; Programmende |
| FEHLER: SETAL(66000) | |

## 3.1.3.5 Aufrunden (ROUNDUP)

Mit der Funktion "ROUNDUP" können Eingabewerte vom Typ REAL (gebrochene Zahlen mit Dezimalpunkt) auf die nächste größere ganze Zahl aufgerundet werden.

### Syntax

```
ROUNDUP(<Wert>)
```

### Bedeutung

| | |
|---|---|
| ROUNDUP: | Befehl zum Aufrunden eines Eingabewerts |
| <Wert>: | Eingabewert vom Typ REAL |

**Hinweis**

Eingabewerte vom Typ INTEGER (eine ganze Zahl) werden unverändert zurückgeliefert.

### Beispiele

**Beispiel 1: Verschiedene Eingabewerte und deren Rundungsergebnisse**

| Beispiel | Rundungsergebnis |
|---|---|
| ROUNDUP(3.1) | 4.0 |
| ROUNDUP(3.6) | 4.0 |
| ROUNDUP(-3.1) | -3.0 |
| ROUNDUP(-3.6) | -3.0 |
| ROUNDUP(3.0) | 3.0 |
| ROUNDUP(3) | 3.0 |

**Beispiel 2: ROUNDUP im NC-Programm**

**Programmcode**

```
N10 X=ROUNDUP(3.5) Y=ROUNDUP(R2+2)
N15 R2=ROUNDUP($AA_IM[Y])
N20 WHEN X=100 DO Y=ROUNDUP($AA_IM[X])
...
```

## 3.1.4 Stringoperationen

### Stringoperationen

Neben den klassischen Operationen "Zuweisung" und "Vergleich" sind folgende Stringoperationen möglich:

- Typenkonvertierung nach STRING (AXSTRING) (Seite 453)
- Typenkonvertierung von STRING (NUMBER, ISNUMBER, AXNAME) (Seite 454)
- Verkettung von Strings (<<) (Seite 455)

- Wandlung in Klein-/Großbuchstaben (TOLOWER, TOUPPER) (Seite 456)
- Länge eines Strings bestimmen (STRLEN) (Seite 457)
- Zeichen/String in String suchen (INDEX, RINDEX, MINDEX, MATCH) (Seite 457)
- Auswahl eines Teilstrings (SUBSTR) (Seite 458)
- Lesen und Schreiben von einzelnen Zeichen (Seite 459)
- String formatieren (SPRINT) (Seite 461)

### Sonderbedeutung des 0-Zeichens

Das 0-Zeichen wird intern als Enderkennung eines Strings interpretiert. Wird ein Zeichen durch das 0-Zeichen ersetzt, wird der String damit verkürzt.

Beispiel:

```
Programmcode                                    Kommentar
DEF STRING[20] STRG="Achse . steht"
STRG[6]="X"
MSG(STRG)                                       ; Liefert die Meldung "Achse X steht".
STRG[6]=0
MSG(STRG)                                       ; Liefert die Meldung "Achse".
```

## 3.1.4.1 Typenkonvertierung nach STRING (AXSTRING)

Durch die Funktion "Typkonvertierung nach STRING" lassen sich Variablen unterschiedlichen Typs als Bestandteil einer Meldung (MSG) nutzen.

Erfolgt bei Verwendung des Operators << implizit für die Datentypen INT, REAL, CHAR und BOOL (siehe " Verkettung von Strings (<<) (Seite 455) ").

Ein INT-Wert wird in die normal lesbare Form umgewandelt. Bei REAL-Werten werden bis zu 10 Nachkommastellen angegeben.

Mit dem Befehl `AXSTRING` können Variable vom Typ AXIS nach STRING gewandelt werden.

### Syntax

```
<STRING_ERG> = << <bel._Typ>
<STRING_ERG> = AXSTRING(<Achsbezeichner>)
```

### Bedeutung

| | | |
|---|---|---|
| `<STRING_ERG>`: | Variable für das Ergebnis der Typkonvertierung | |
| | Typ: | STRING |
| `<bel._Typ>`: | Variablen-Typen INT, REAL, CHAR, STRING und BOOL | |
| `AXSTRING`: | Der Befehl `AXSTRING` liefert den angegebenen Achsbezeichner als String. | |
| `<Achsbezeichner>`: | Variable für Achsbezeichner | |
| | Typ: | AXIS |

**Hinweis**

FRAME-Variablen können nicht konvertiert werden.

### 3.1.4.2 Typenkonvertierung von STRING (NUMBER, ISNUMBER, AXNAME)

Mit dem Befehl `NUMBER` wird von STRING nach REAL konvertiert. Die Konvertierbarkeit kann mit dem Befehl `ISNUMBER` überprüft werden.

Mit dem Befehl `AXNAME` wird ein String in den Datentyp AXIS konvertiert.

## Syntax

```
<REAL_ERG>=NUMBER("<String>")
<BOOL_ERG>=ISNUMBER("<String>")
<AXIS_ERG>=AXNAME("<String>")
```

## Bedeutung

| | | | |
|---|---|---|---|
| `NUMBER`: | Der Befehl `NUMBER` liefert die durch den `<String>` dargestellte Zahl als REAL-Wert zurück. | | |
| `<String>`: | Zu konvertierende Variable vom Typ STRING | | |
| `<REAL_ERG>`: | Variable für das Ergebnis der Typkonvertierung mit `NUMBER` | | |
| | Typ: | REAL | |
| `ISNUMBER`: | Mit dem Befehl `ISNUMBER` kann überprüft werden, ob der `<String>` in eine gültige Zahl gewandelt werden kann. | | |
| `<BOOL_ERG>`: | Variable für das Ergebnis der Abfrage mit `ISNUMBER` | | |
| | Typ: | BOOL | |
| | Wert: | TRUE | `ISNUMBER` liefert den Wert TRUE, wenn der `<String>` eine nach den Regeln der Sprache gültige REAL-Zahl darstellt. |
| | | FALSE | Liefert `ISNUMBER` den Wert FALSE, wird bei Aufruf von `NUMBER` mit dem gleichen `<String>` Alarm ausgelöst. |
| `AXNAME`: | Der Befehl `AXNAME` wandelt den angegebenen `<String>` in einen Achsbezeichner.<br>**Hinweis:**<br>Kann der `<String>` keinem projektierten Achsbezeichner zugeordnet werden, wird ein Alarm ausgelöst. | | |
| `<AXIS_ERG>`: | Variable für das Ergebnis der Typkonvertierung mit `AXNAME` | | |
| | Typ: | AXIS | |

## Beispiel

| **Programmcode** | **Kommentar** |
|---|---|
| `DEF BOOL BOOL_ERG` | |

| **Programmcode** | **Kommentar** |
|---|---|
| DEF REAL REAL_ERG | |
| DEF AXIS AXIS_ERG | |
| BOOL_ERG=ISNUMBER("1234.9876Ex-7") | ; BOOL_ERG == TRUE |
| BOOL_ERG=ISNUMBER("1234XYZ") | ; BOOL_ERG == FALSE |
| REAL_ERG=NUMBER("1234.9876Ex-7") | ; REAL_ERG == 1234.9876Ex-7 |
| AXIS_ERG=AXNAME("X") | ; AXIS_ERG == X |

### 3.1.4.3 Verkettung von Strings (<<)

Die Funktion "Verkettung von Strings" schafft die Möglichkeit, einen String aus einzelnen Bestandteilen zusammensetzen zu können.

Realisiert wird die Verkettung über den Operator "**<<**". Dieser Operator hat für alle Kombinationen der Basistypen CHAR, BOOL, INT, REAL und STRING als Zieltyp STRING. Eine eventuell notwendige Konvertierung wird nach den bestehenden Regeln vorgenommen.

#### Syntax

```
<bel._Typ> << <bel._Typ>
```

#### Bedeutung

| | |
|---|---|
| <bel._Typ>: | Variable vom Typ CHAR, BOOL, INT, REAL oder STRING |
| << : | Operator für die Verkettung von Variablen (<bel._Typ>) zu einer zusammengesetzten Zeichenkette (Typ STRING).<br>Dieser Operator ist auch alleinig als sog. "unäre" Variante verfügbar. So ist es möglich, eine explizite Typwandlung nach STRING auszuführen (nicht für FRAME und AXIS):<br><< <bel._Typ> |

Beispielsweise lässt sich so eine Meldung oder ein Kommando aus Textlisten zusammensetzen und Parameter (etwa ein Bausteinname) einfügen:
`MSG(STRG_TAB[LOAD_IDX]<<BAUSTEIN_NAME)`

**Hinweis**

Die Zwischenergebnisse bei der Stringverkettung dürfen die maximale Stringlänge nicht überschreiten.

**Hinweis**

Die Typen FRAME und AXIS können nicht zusammen mit dem Operator "**<<**" verwendet werden.

## Beispiele

### Beispiel 1: Verkettung von Strings

```
Programmcode                                          Kommentar
DEF INT IDX=2
DEF REAL VALUE=9.654
DEF STRING[20] STRG="INDEX:2"
IF STRG=="Index:"<<IDX GOTOF NO_MSG
MSG("Index:"<<IDX<<"/Wert:"<<VALUE)                   ; Anzeige:
                                                      "Index:2/Wert:9.654"
NO_MSG:
```

### Beispiel 2: Explizite Typkonvertierung mit <<

```
Programmcode              Kommentar
DEF REAL VALUE=3.5
<<VALUE                   ; Die angegebene Variable vom Typ REAL wird in den Typ
                          STRING konvertiert.
```

## 3.1.4.4 Wandlung in Klein-/Großbuchstaben (TOLOWER, TOUPPER)

Die Funktion "Wandlung in Klein-/Großbuchstaben" erlaubt es, alle Buchstaben einer Zeichenkette in eine einheitliche Darstellung zu wandeln.

## Syntax

```
<STRING_ERG>=TOUPPER("<String>")
<STRING_ERG>=TOLOWER("<String>")
```

## Bedeutung

| | | |
|---|---|---|
| TOUPPER: | Mit dem Befehl TOUPPER werden alle Buchstaben einer Zeichenkette in **Groß**buchstaben umgewandelt. | |
| TOLOWER: | Mit dem Befehl TOLOWER werden alle Buchstaben einer Zeichenkette in **Klein**buchstaben umgewandelt. | |
| <String>: | Zeichenkette, die umgewandelt werden soll | |
| | Typ: | STRING |
| <STRING_ERG>: | Variable für das Ergebnis der Umwandlung | |
| | Typ: | STRING |

## Beispiel

Da es auch möglich ist, Benutzereingaben an der Bedienoberfläche anzustoßen, kann eine einheitliche Darstellung mit Klein- oder Großbuchstaben erreicht werden:

```
Programmcode
DEF STRING [29] STRG
```

**Programmcode**

```
...
IF "LEARN.CNC"==TOUPPER(STRG) GOTOF LOAD_LEARN
```

### 3.1.4.5 Länge eines Strings bestimmen (STRLEN)

Mit dem Befehl `STRLEN` ist es möglich, die Länge einer Zeichenkette zu bestimmen.

#### Syntax

```
<INT_ERG>=STRLEN("<STRING>")
```

#### Bedeutung

| | | |
|---|---|---|
| `STRLEN`: | Mit dem Befehl `STRLEN` wird die Länge der angegebenen Zeichenkette bestimmt.<br>Es wird die Anzahl der Zeichen zurückgegeben, die - vom Anfang der Zeichenkette an gezählt - keine 0-Zeichen sind. | |
| `<String>`: | Zeichenkette, deren Länge bestimmt werden soll | |
| | Typ: | STRING |
| `<INT_ERG>`: | Variable für das Ergebnis der Bestimmung | |
| | Typ: | INT |

#### Beispiel

Die Funktion im Zusammenhang mit dem Einzelzeichenzugriff ermöglicht es, das Ende einer Zeichenkette zu bestimmen:

**Programmcode**

```
IF (STRLEN(BAUSTEIN_NAME)>10) GOTOF FEHLER
```

### 3.1.4.6 Zeichen/String in String suchen (INDEX, RINDEX, MINDEX, MATCH)

Diese Funktionalität erlaubt es, einzelne Zeichen bzw. einen String in einem weiteren String zu suchen. Die Funktionsergebnisse geben an, an welcher Position des Strings das Zeichen/der String im zu untersuchenden String gefunden wurde.

#### Syntax

`INT_ERG=INDEX(STRING,CHAR)` ; **Ergebnistyp: INT**

`INT_ERG=RINDEX(STRING,CHAR)` ; **Ergebnistyp: INT**

`INT_ERG=MINDEX(STRING,STRING)` ; **Ergebnistyp: INT**

`INT_ERG=MATCH(STRING,STRING)` ; **Ergebnistyp: INT**

### Semantik

Suchfunktionen: Sie liefern die Position im String (erster Parameter) zurück, wo die Suche erfolgreich war. Kann das Zeichen/der String nicht gefunden werden, wird der Wert -1 zurückgegeben. Das erste Zeichen hat dabei die Position 0.

## Bedeutung

| | |
|---|---|
| INDEX: | sucht das als zweiten Parameter angegebene Zeichen (von vorne) im ersten Parameter. |
| RINDEX: | sucht das als zweiten Parameter angegebene Zeichen (von hinten) im ersten Parameter. |
| MINDEX: | entspricht der Funktion INDEX, außer, dass eine Liste von Zeichen (als String) übergeben wird, von denen der Index des ersten gefundenen Zeichens zurückgegeben wird. |
| MATCH: | sucht einen String in einem String. |

So lassen sich Strings nach bestimmten Kriterien zerlegen, etwa an Positionen mit Leerzeichen oder Pfadtrennzeichen ("/").

## Beispiel

### Zerlegen einer Eingabe in Pfad- und Bausteinnamen

```
Programmcode                              Kommentar
DEF INT PFADIDX, PROGIDX
DEF STRING[26] EINGABE
DEF INT LISTIDX
EINGABE = "/_N_MPF_DIR/_N_EXECU-
TE_MPF"
LISTIDX = MINDEX (EINGABE,                ; Als Wert in LISTIDX wird 3 zurückgeliefert;
"M,N,O,P") + 1                            da "N" das erste Zeichen im Parameter EINGABE,
                                          aus der Auswahlliste von vorne, ist.
PFADIDX = INDEX (EINGABE, "/") +1         ; damit gilt: PFADIDX = 1
PROGIDX = RINDEX (EINGABE, "/") +1        ; damit gilt: PROGIDX = 12
                                          ; mit Hilfe der im nächsten Abschnitt eingeführ-
                                          ten Funktion SUBSTR läßt sich
                                          die Variable EINGABE in die Komponen-
                                          ten ;"Pfad";und "Baustein" zerlegen:
VARIABLE = SUBSTR (EINGABE,               ; liefert dann "_N_MPF_DIR"
PFADIDX, PROGIDX-PFADIDX-1)
VARIABLE = SUBSTR (EINGABE, PROGIDX)      ; liefert dann "_N_EXECUTE_MPF"
```

## 3.1.4.7 Auswahl eines Teilstrings (SUBSTR)

Mit der Funktion SUBSTRING können beliebige Teile innerhalb eines Strings gelesen werden.

## Syntax

```
<STRING_ERG>=SUBSTR(<String>,<Index>,<Länge>)
```

```
<STRING_ERG>=SUBSTR(<String>,<Index>)
```

## Bedeutung

| | |
|---|---|
| SUBSTR: | Die Funktion liefert aus <String> einen Teilstring, ausgehend von <Index> mit der angegebenen <Länge>.<br>Ist der Parameter <Länge> nicht angegeben, liefert die Funktion einen Teilstring ausgehend von <Index> bis zum Stringende. |
| <Index>: | Anfangsposition des Teilstrings innerhalb des Strings. Liegt die Anfangsposition hinter dem Stringende, wird ein Leerstring ( "") zurückgegeben. Erstes Zeichen des Strings: Index = 0<br>Wertebereich: 0 ... (Stringlänge - 1) |
| <Länge>: | Länge des Teilstrings. Wird eine zu große Länge angegeben, wird nur der Teilstring bis Stringende zurückgegeben.<br>Wertebereich: 1 ... (Stringlänge - 1) |

## Beispiel

```
Programmcode                                              Kommentar
DEF STRING[29] ERG
;                         1
;              01234567890123456789
ERG = SUBSTR("QUITTUNG: 10 bis 99", 10, 2)           ; ERG == "10"
ERG = SUBSTR("QUITTUNG: 10 bis 99", 10)              ; ERG == "10 bis 99"
```

### 3.1.4.8 Lesen und Schreiben von einzelnen Zeichen

Innerhalb eines Strings können einzelne Zeichen gelesen und geschrieben werden.

Dabei sind folgende Randbedingungen zu beachten:

- nur bei anwenderdefinierten Variablen möglich, nicht bei Systemvariablen
- einzelne Zeichen eines String werden bei Unterprogrammaufrufen nur "call by value" übergeben

## Syntax

```
<Zeichen>=<String>[<Index>]
<Zeichen>=<String_Array>[<Array_Index>,<Index>]
<String>[<Index>]=<Zeichen>
<String_Array>[<Array_Index>,<Index>]=<Zeichen>
```

## Bedeutung

| | |
|---|---|
| <String>: | Beliebiger String |
| <Zeichen>: | Variable vom Typ CHAR |
| <Index>: | Position des Zeichens innerhalb des Strings.<br>Erstes Zeichen des Strings: Index = 0<br>Wertebereich: 0 ... (Stringlänge - 1) |

## Beispiele

### Beispiel 1: Variable Meldung

```
Programmcode                                Kommentar
;                          0123456789
DEF STRING[50] MELDUNG = "Achse n hat Position erreicht"
MELDUNG[6] = "X"
MSG(MELDUNG)                               ; "Achse X hat Position erreicht"
```

### Beispiel 2: Auswerten einer Systemvariablen

```
Programmcode                                Kommentar
DEF STRING[50] STRG                        ; Zwischenspeicher für Systemvariable
...
STRG = $P_MMCA                             ; Systemvariable laden
IF STRG[0] == "E" GOTO ...                 ; Auswerten der Systemvariable
```

### Beispiel 3: Parameterübergabe "call by value" und "call by reference"

```
Programmcode                                Kommentar
;                        0123456
DEF STRING[50] STRG = "Achse X"
DEF CHAR CHR
...
EXTERN UP_VAL(ACHSE)                       ; Definition Unterprogramm mit "call
                                           by value" Parameter
EXTERN UP_REF(VAR ACHSE)                   ; Definition Unterprogramm mit "call
                                           by ref." Parameter
...
UP_VAL(STRG[6])                            ; Parameterübergabe "by value"
...
CHR = STRG[6]                              ; Zwischenspeichern
UP_REF(CHR)                                ; Parameterübergabe "by reference"
```

### 3.1.4.9 String formatieren (SPRINT)

Mit der vordefinierten Funktion SPRINT können Zeichenketten formatiert und z. B. für die Ausgabe an externe Geräte aufbereitet werden (siehe auch "Process DataShare - Ausgabe auf ein externes Gerät/Datei (EXTOPEN, WRITE, EXTCLOSE) (Seite 1014)").

## Syntax

```
"<Ergebnis_String>"=SPRINT("<Format_String>",<Wert_1>,<Wert_2>,...,
<Wert_n>)
```

## Bedeutung

| | |
|---|---|
| SPRINT: | Bezeichner für eine vordefinierte Funktion, die einen Wert vom Typ STRING liefert. |
| "<Format_String>": | Zeichenkette, die feste und variable Teile enthält. Die variablen Teile werden durch das Formatsteuerzeichen % und einer nachfolgenden Formatbeschreibung festgelegt. |
| < Wert_1>,< Wert_2>,…,< Wert_n>: | Wert in Form einer Konstanten oder NC-Variablen, der an der Stelle, an der das n-te Formatsteuerzeichen % steht, entsprechend der Formatbeschreibung in den <Format_String> eingefügt wird. |
| "<Ergebnis_String>": | Formatierte Zeichenkette (maximal 400 Bytes) |

## Verfügbare Formatbeschreibungen

| | |
|---|---|
| %B: | Wandlung in den String "TRUE", wenn der zu wandelnde Wert:<br>• ungleich 0 ist.<br>• kein Leerstring ist (bei String-Werten).<br>Wandlung in den String "FALSE", wenn der zu wandelnde Wert:<br>• gleich 0 ist.<br>• ein Leerstring ist.<br>**Beispiel:**<br>`N10 DEF BOOL BOOL_VAR=1`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF BOOL_VAR:%B", BOOL_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF BOOL_VAR:TRUE" beschrieben. |
| %C: | Wandlung in ein ASCII-Zeichen.<br>**Beispiel:**<br>`N10 DEF CHAR CHAR_VAR="X"`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF CHAR_VAR:%C",CHAR_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF CHAR_VAR:X" beschrieben. |

| | |
|---|---|
| %D: | Wandlung in einen String mit einem ganzzahligen Wert (INTEGER).<br>**Beispiel:**<br>`N10 DEF INT INT_VAR=123`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF INT_VAR:%D",INT_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF INT_VAR:123" beschrieben. |
| %<m>D: | Wandlung in einen String mit einem ganzzahligen Wert (INTEGER). Der String hat eine Mindestlänge von <m> Zeichen. Fehlende Stellen werden linksbündig mit Leerzeichen aufgefüllt.<br>**Beispiel:**<br>`N10 DEF INT INT_VAR=-123`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF INT_VAR:%6D",INT_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF INT_VAR:xx-123" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen). |
| %F: | Wandlung in einen String mit einer Dezimalzahl mit 6 Nachkommastellen. Die Nachkommstellen werden ggf. gerundet oder mit 0 aufgefüllt.<br>**Beispiel:**<br>`N10 DEF REAL REAL_VAR=-1.2341234EX+03`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%F",REAL_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR: -1234.123400" beschrieben. |
| %<m>F: | Wandlung in einen String mit einer Dezimalzahl mit 6 Nachkommastellen und einer Gesamtlänge von mindestens <m> Zeichen. Die Nachkommstellen werden ggf. gerundet oder mit 0 aufgefüllt. Fehlende Zeichen auf die Gesamtlänge <m> werden linksbündig mit Leerzeichen aufgefüllt.<br>**Beispiel:**<br>`N10 DEF REAL REAL_VAR=-1.23412345678EX+03`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15F",REAL_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR: xxx-1234.123457" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen). |
| %.<n>F: | Wandlung in einen String mit einer Dezimalzahl mit <n> Nachkommastellen. Die Nachkommastellen werden ggf. gerundet oder mit 0 aufgefüllt.<br>**Beispiel:**<br>`N10 DEF REAL REAL_VAR=-1.2345678EX+03`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3F",REAL_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:-1234.568" beschrieben. |

| | |
|---|---|
| %<m>.<n>F: | Wandlung in einen String mit einer Dezimalzahl mit <n> Nachkommastellen und einer Gesamtlänge von mindestens <m> Zeichen. Die Nachkommstellen werden ggf. gerundet oder mit 0 aufgefüllt. Fehlende Zeichen auf die Gesamtlänge <m> werden linksbündig mit Leerzeichen aufgefüllt.<br>**Beispiel:**<br>`N10 DEF REAL REAL_VAR=-1.2341234567890EX+03`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%10.2F",REAL_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:xx-1234.12" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen). |
| %E: | Wandlung in einen String mit einer Dezimalzahl in Exponentialdarstellung. Die Mantisse wird normalisiert mit einer Vorkommastelle und 6 Nachkommastellen abgelegt. Die Nachkommastellen werden ggf. gerundet oder mit 0 aufgefüllt. Der Exponent beginnt mit dem Schlüsselwort "EX". Es folgt das Vorzeichen ("+" oder "-") und eine zwei- oder dreistellige Zahl.<br>**Beispiel:**<br>`N10 DEF REAL REAL_VAR=-1234.567890`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%E",REAL_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:-1.234568EX+03" beschrieben. |
| %<m>E: | Wandlung in einen String mit einer Dezimalzahl in Exponentialdarstellung mit einer Gesamtlänge von mindestens <m> Zeichen. Fehlende Zeichen werden linksbündig mit Leerzeichen aufgefüllt. Die Mantisse wird normalisiert mit einer Vorkommastelle und 6 Nachkommastellen abgelegt. Die Nachkommastellen werden ggf. gerundet oder mit 0 aufgefüllt. Der Exponent beginnt mit dem Schlüsselwort "EX". Es folgt das Vorzeichen ("+" oder "-") und eine zwei- oder dreistellige Zahl.<br>**Beispiel:**<br>`N10 DEF REAL REAL_VAR=-1234.5`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%20E",REAL_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:xxxxxx-1.234500EX+03" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen) |
| %.<n>E: | Wandlung in einen String mit einer Dezimalzahl in Exponentialdarstellung. Die Mantisse wird normalisiert mit einer Vorkommastelle und <n> Nachkommastellen abgelegt. Die Nachkommastellen werden ggf. gerundet oder mit 0 aufgefüllt. Der Exponent beginnt mit dem Schlüsselwort "EX". Es folgt das Vorzeichen ("+" oder "-") und eine zwei- oder dreistellige Zahl.<br>**Beispiel:**<br>`N10 DEF REAL REAL_VAR=-1234.5678`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.2E",REAL_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:-1.23EX+03" beschrieben. |

| | |
|---|---|
| %<m>.<n>E: | Wandlung in einen String mit einer Dezimalzahl in Exponentialdarstellung mit einer Gesamtlänge von mindestens <m> Zeichen. Fehlende Zeichen werden linksbündig mit Leerzeichen aufgefüllt. Die Mantisse wird normalisiert mit einer Vorkommastelle und <n> Nachkommastellen abgelegt. Die Nachkommastellen werden ggf. gerundet oder mit 0 aufgefüllt. Der Exponent beginnt mit dem Schlüsselwort "EX". Es folgt das Vorzeichen ("+" oder "-") und eine zwei- oder dreistellige Zahl.<br>**Beispiel:**<br>`N10 DEF REAL REAL_VAR=-1234.5678`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.2E", REAL_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:xx-1.23EX+03" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen). |
| %G: | Wandlung in einen String mit einer Dezimalzahl je nach Wertebereich in Dezimal- oder Exponentialdarstellung: ist der darzustellende Wert betragsmäßig kleiner als 1.0EX-04 oder größer/gleich 1.0EX+06 wird die Exponentialdarstellung gewählt, sonst die Dezimaldarstellung. Es werden maximal sechs signifikante Stellen angezeigt, ggf. wird gerundet.<br>**Beispiel mit Dezimaldarstellung:**<br>`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:0.000123457" beschrieben.<br>**Beispiel mit Exponentialdarstellung:**<br>`N10 DEF REAL REAL_VAR=1.234567890123456EX+06`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:1.23457EX+06" beschrieben. |
| %<m>G: | Wandlung in einen String mit einer Dezimalzahl je nach Wertebereich in Dezimal- oder Exponentialdarstellung (wie %G). Der String hat eine Gesamtlänge von mindestens <m> Zeichen. Fehlende Zeichen werden linksbündig mit Leerzeichen aufgefüllt.<br>**Beispiel mit Dezimaldarstellung:**<br>`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:xxxx0.000123457" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen).<br>**Beispiel mit Exponentialdarstellung:**<br>`N10 DEF REAL REAL_VAR=1.234567890123456EX+06`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:xxx1.23457EX+06" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen). |

| | |
|---|---|
| %.<n>G: | Wandlung in einen String mit einer Dezimalzahl je nach Wertebereich in Dezimal- oder Exponentialdarstellung. Es werden maximal <n> signifikante Stellen angezeigt, ggf. wird gerundet. Ist der darzustellende Wert betragsmäßig kleiner als 1.0EX-04 oder größer/gleich 1.0EX(+<n>) wird die Exponentialdarstellung gewählt, sonst die Dezimaldarstellung.<br>**Beispiel mit Dezimaldarstellung:**<br>`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:0.000123" beschrieben.<br>**Beispiel mit Exponentialdarstellung:**<br>`N10 DEF REAL REAL_VAR=1.234567890123456EX+03`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT = SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:1.23EX+03" beschrieben. |
| %<m>.<n>G: | Wandlung in einen String mit einer Dezimalzahl je nach Wertebereich in Dezimal- oder Exponentialdarstellung (wie %.<n>G). Der String hat eine Gesamtlänge von mindestens <m> Zeichen. Fehlende Zeichen werden linksbündig mit Leerzeichen aufgefüllt.<br>**Beispiel mit Dezimaldarstellung:**<br>`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)`<br>Ergebnis: Die Stringvariable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:xxx0.0001235" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen).<br>**Beispiel mit Exponentialdarstellung:**<br>`N10 DEF REAL REAL_VAR=1.234567890123456EX+04`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF REAL_VAR:xx1.235EX+06" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen). |

| | |
|---|---|
| %.<n>P: | Wandlung eines REAL-Werts in einen INTEGER-Wert unter Berücksichtigung von <n> Nachkommastellen. Der INTEGER-Wert wird als 32-Bit-Binärzahl ausgegeben. Kann der zu wandelnde Wert nicht mit 32 Bit dargestellt werden, wird die Bearbeitung mit Alarm abgebrochen.<br>Da eine mit der Formatanweisung %.<n>P erzeugte Byte-Folge auch binäre Nullen enthalten kann, entspricht der so erzeugte Gesamt-String nicht mehr den Konventionen des NC-Datentyps STRING. Er kann daher weder in einer Variablen vom Typ STRING gespeichert werden, noch mit den String-Befehlen der NC-Sprache weiter bearbeitet werden. Die einzig mögliche Verwendung ist die Parameterübergabe an den WRITE-Befehl mit Ausgabe an ein entsprechendes externes Gerät (siehe folgendes Beispiel).<br>Sobald der <Format_String> eine Formatbeschreibung vom Typ %P enthält, wird der gesamte String, mit Ausnahme der mit %.<n>P generierten Binärzahl, entsprechend dem MD10750 $MN_SPRINT_FORMAT_P_CODE im Zeichen-Code ASCII, ISO (DIN6024) oder EIA (RS244) ausgegeben. Wird ein nicht umsetzbares Zeichen programmiert, wird die Bearbeitung mit Alarm abgebrochen.<br>**Beispiel:**<br>`N10 DEF REAL REAL_VAR=123.45`<br>`N20 DEF INT ERROR`<br>`N30 DEF STRING[20] EXT_DEVICE="/ext/dev/1"`<br>`...`<br>`N100 EXTOPEN(ERROR,EXT_DEVICE)`<br>`N110 IF ERROR <> 0`<br>`... ; Fehlerhandling`<br>`N200 WRITE(ERROR,EXT_DEVICE,SPRINT("INTEGER BINARY CODED:%.3P",REAL_VAR)`<br>`N210 IF ERROR <> 0`<br>`… ; Fehlerhandling`<br>Ergebnis: Der String "INTEGER BINARY CODED: 'H0001E23A'" wird an das Ausgabegerät /ext/dev/1 übertragen. Der hexadezimale Wert 0x0001E23A entspricht dem dezimalen Wert 123450. |

| | |
|---|---|
| %<m>.<n>P: | Wandlung eines REAL-Werts entsprechend der Einstellung im Maschinendatum MD10751 $MN_SPRINT_FORMAT_P_DECIMAL in einen String mit:<br>• einer Ganzzahl von <m> + <n> Stellen **oder**<br>• einer Dezimalzahl mit maximal <m> Vorkommastellen und exakt <n> Nachkommastellen.<br>Wie bei der Formatbeschreibung %.<n>P wird der gesamte String in dem durch MD10750 $MN_SPRINT_FORMAT_P_CODE festgelegten Zeichen-Code abgelegt.<br>**Wandlung bei MD10751 = 0:**<br>Der REAL-Wert wird in einen String mit einer Ganzzahl von <m> + <n> Stellen gewandelt. Ggf. werden Nachkommstellen auf <n> Stellen gerundet oder mit 0 aufgefüllt. Fehlende Vorkommastellen werden mit Leerzeichen aufgefüllt. Das Minus-Vorzeichen wird linksbündig angefügt, an Stelle des Plus-Vorzeichens wird ein Leerzeichen abgesetzt.<br>**Beispiel:**<br>`N10 DEF REAL REAL_VAR=-123.45`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("PUNCHED TAPE FORMAT:%5.3P",REAL_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "PUNCHED TAPE FORMAT:-xx123450" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen).<br>**Wandlung bei MD10751 = 1:**<br>Der REAL-Wert wird in einen String mit einer Dezimalzahl mit maximal <m> Vorkommastellen und exakt <n> Nachkommastellen gewandelt. Ggf. werden die Vorkommastellen abgeschnitten und die Nachkommastellen gerundet oder mit 0 aufgefüllt. Ist <n> gleich 0 entfällt auch der Dezimalpunkt.<br>**Beispiel:**<br>`N10 DEF REAL REAL_VAR1=-123.45`<br>`N20 DEF REAL REAL_VAR2=123.45`<br>`N30 DEF STRING[80] RESULT`<br>`N40 RESULT=SPRINT("PUNCHED TAPE FORMAT:%5.3P VAR2:%2.0P", REAL_VAR1,REAL_VAR2)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "PUNCHED TAPE FORMAT:-123.450 VAR2:23" beschrieben. |
| %S: | Einfügen eines Strings.<br>**Beispiel:**<br>`N10 DEF STRING[16] STRING_VAR="ABCDEFG"`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%S",STRING_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF STRING_VAR:ABCDEFG" beschrieben. |
| %<m>S: | Einfügen eines Strings mit mindestens <m> Zeichen. Fehlende Stellen werden mit Leerzeichen aufgefüllt.<br>**Beispiel:**<br>`N10 DEF STRING[16] STRING_VAR="ABCDEFG"`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10S",STRING_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF STRING_VAR:xxxABCDEFG" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen). |

| | |
|---|---|
| %.<n>S: | Einfügen von <n> Zeichen eines Strings (beginnend mit dem ersten Zeichen).<br>**Beispiel:**<br>`N10 DEF STRING[16] STRING_VAR="ABCDEFG"`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%.3S",STRING_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF STRING_VAR:ABC" beschrieben. |
| %<m>.<n>S: | Einfügen von <n> Zeichen eines Strings (beginnend mit dem ersten Zeichen). Die Gesamtlänge des erzeugten Strings hat mindestens < m> Zeichen. Fehlende Stellen werden mit Leerzeichen aufgefüllt.<br>**Beispiel:**<br>`N10 DEF STRING[16] STRING_VAR="ABCDEFG"`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10.5S", STRING_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "CONTENT OF STRING_VAR:xxxxxABCDE" beschrieben ("x" steht im Beispiel stellvertretend für Leerzeichen). |
| %X: | Wandlung eines INTEGER-Werts in einen String mit Hexadezimal-Darstellung.<br>**Beispiel:**<br>`N10 DEF INT INT_VAR='HA5B8'`<br>`N20 DEF STRING[80] RESULT`<br>`N30 RESULT=SPRINT("INTEGER HEXADECIMAL:%X",INT_VAR)`<br>Ergebnis: Die String-Variable RESULT wird mit der Zeichenkette "INTEGER HEXADECIMAL:A5B8" beschrieben. |

**Hinweis**

Die Eigenschaft der NC-Sprache, bezüglich Bezeichnern und Schlüsselworten nicht zwischen Groß- und Kleinbuchstaben zu unterscheiden, gilt auch für die Formatbeschreibungen. Sie können daher ohne funktionalen Unterschied mit Klein- oder Großbuchstaben programmiert werden.

## Kombinationsmöglichkeiten

Die folgende Tabelle gibt Auskunft darüber, welche NC-Datentypen mit welcher Formatbeschreibung kombiniert werden können. Es gelten die Regeln zur impliziten Datentypwandlung (siehe "Datentypen (Seite 428)").

| | NC-Datentypen | | | | | | |
|---|---|---|---|---|---|---|---|
| | **BOOL** | **CHAR** | **INT** | **REAL** | **STRING** | **AXIS** | **FRAME** |
| **%B** | + | + | + | + | + | - | - |
| **%C** | - | + | - | - | + | - | - |
| **%D** | + | + | + | + | - | - | - |
| **%F** | - | - | + | + | - | - | - |
| **%E** | - | - | + | + | - | - | - |
| **%G** | - | - | + | + | - | - | - |
| **%S** | - | + | - | - | + | - | - |

| | NC-Datentypen | | | | | | |
|---|---|---|---|---|---|---|---|
| | BOOL | CHAR | INT | REAL | STRING | AXIS | FRAME |
| %X | + | + | + | - | - | - | - |
| %P | - | - | + | + | - | - | - |

**Hinweis**

Die Tabelle zeigt, dass die NC-Datentypen AXIS und FRAME nicht direkt in der SPRINT-Funktion verwendet werden können. Es ist aber möglich:

- den Datentyp AXIS mit der Funktion AXSTRING in einen String zu wandeln, der dann mit SPRINT weiterverarbeitet werden kann.
- die einzelnen Werte des Datentyps FRAME per Frame-Komponentenzugriff zu lesen. Dadurch bekommt man ein Datum vom Typ REAL, das mit SPRINT weiterverarbeitet werden kann.

## 3.1.5 Programmsprünge und -verzweigungen

### 3.1.5.1 Rücksprung auf Programmanfang (GOTOS)

Mit dem Befehl `GOTOS` ist es möglich, zur Programmwiederholung an den Anfang eines Haupt- oder Unterprogramms zurückzuspringen.

Über Maschinendaten kann eingestellt werden, dass bei jedem Rücksprung auf den Programmanfang:

- die Programmlaufzeit auf "0" gesetzt wird.
- die Werkstückzählung um den Wert "1" erhöht wird.

**Syntax**

```
GOTOS
```

**Bedeutung**

| | | |
|---|---|---|
| `GOTOS`: | Sprunganweisung mit Sprungziel Programmanfang. | |
| | Die Ausführung wird gesteuert über das NC/PLC-Nahtstellensignal:<br>DB21, ... DBX384.0 (Programmverzweigung steuern) | |
| | Wert: | Bedeutung: |
| | 0 | Kein Rücksprung auf den Programmanfang. Die Programmbearbeitung wird mit dem nächsten Teileprogrammsatz nach `GOTOS` fortgeführt. |
| | 1 | Rücksprung auf den Programmanfang. Das Teileprogramm wird wiederholt. |

### Randbedingungen

- GOTOS löst intern ein STOPRE (Vorlaufstopp) aus.
- Bei einem Teileprogramm mit Datendefinitionen (LUD-Variablen) wird mit GOTOS auf den ersten Programmsatz nach dem Definitionsabschnitt gesprungen, d. h. die Datendefinitionen werden nicht erneut ausgeführt. Die definierten Variablen behalten daher den im GOTOS-Satz erreichten Wert und werden nicht auf die im Definitionsabschnitt programmierten Standardwerte zurückgesetzt.
- In Synchronaktionen und Technologie-Zyklen steht der Befehl GOTOS nicht zur Verfügung.

### Beispiel

| **Programmcode** | **Kommentar** |
|---|---|
| N10 ... | ; Programmanfang. |
| ... | |
| N90 GOTOS | ; Sprung an den Programmanfang. |
| ... | |

## 3.1.5.2 Programmsprünge auf Sprungmarken (GOTOB, GOTOF, GOTO, GOTOC)

In einem Programm können Sprungmarken (Labels) gesetzt werden, auf die von anderen Stellen innerhalb desselben Programms mit dem Befehlen GOTOF, GOTOB, GOTO bzw. GOTOC gesprungen werden kann. Die Programmbearbeitung wird dann mit der Anweisung fortgesetzt, die unmittelbar nach der Sprungmarke folgt. Dadurch sind Verzweigungen innerhalb des Programms realisierbar.

Neben den Sprungmarken sind als Sprungziele auch Haupt- und Nebensatznummern möglich.

Wenn vor der Sprunganweisung eine Sprungbedingung (IF ...) formuliert ist, dann erfolgt der Programmsprung nur dann, wenn die Sprungbedingung erfüllt ist.

### Syntax

```
GOTOB <Sprungziel>
IF <Sprungbedingung> == TRUE GOTOB <Sprungziel>

GOTOF <Sprungziel>
IF <Sprungbedingung> == TRUE GOTOF <Sprungziel>

GOTO <Sprungziel>
IF <Sprungbedingung> == TRUE GOTO <Sprungziel>

GOTOC <Sprungziel>
IF <Sprungbedingung> == TRUE GOTOC <Sprungziel>
```

### Bedeutung

| | |
|---|---|
| GOTOB: | Sprunganweisung mit Sprungziel in Richtung Programmanfang. |
| GOTOF: | Sprunganweisung mit Sprungziel in Richtung Programmende. |

| | | |
|---|---|---|
| GOTO: | Sprunganweisung mit Sprungzielsuche. Die Suche erfolgt erst in Richtung Programmende, dann in Richtung Programmanfang. | |
| GOTOC: | Wirkung wie GOTO mit dem Unterschied, dass der Alarm 14080 "Sprungziel nicht gefunden" unterdrückt wird.<br>Das bedeutet, dass die Programmbearbeitung im Falle einer ergebnislosen Sprungzielsuche nicht abgebrochen wird, sondern mit der auf den Befehl GOTOC folgenden Programmzeile fortgesetzt wird. | |
| <Sprungziel>: | Sprungzielparameter<br>Mögliche Angaben sind: | |
| | <Sprungmarke>: | Sprungziel ist die im Programm gesetzte Sprungmarke mit benutzerdefiniertem Namen:<br><Sprungmarke>: |
| | <Satznummer>: | Sprungziel ist eine Haupt- oder Nebensatznummer (z. B.: 200, N300) |
| | Variable vom Typ STRING: | Variables Sprungziel. Die Variable steht für eine Sprungmarke oder eine Satznummer. |
| IF: | Schlüsselwort zur Formulierung der Sprungbedingung.<br>Die Sprungbedingung lässt alle Vergleichs- und logischen Operationen zu (Ergebnis: TRUE oder FALSE). Der Programmsprung wird ausgeführt, wenn das Ergebnis dieser Operation TRUE ist. | |

**Hinweis**

**Sprungmarken (Labels)**

Sprungmarken stehen immer am Anfang eines Satzes. Wenn eine Programmnummer vorhanden ist, steht die Sprungmarke unmittelbar nach der Satznummer.

Für die Benennung von Sprungmarken gelten folgende Regeln:

- Anzahl an Zeichen:
  - mindestens 2
  - höchstens 32
- Erlaubte Zeichen sind:
  - Buchstaben
  - Ziffern
  - Unterstriche
- Die ersten beiden Zeichen müssen Buchstaben oder Unterstriche sein.
- Nach dem Namen der Sprungmarke folgt ein Doppelpunkt (":").

## Randbedingungen

- Sprungziel kann nur ein Satz mit Sprungmarke oder Satznummer sein, der **innerhalb** des Programms liegt.
- Eine Sprunganweisung ohne Sprungbedingung muss in einem separaten Satz programmiert werden. Bei Sprunganweisungen mit Sprungbedingungen gilt diese Einschränkung nicht. Hier können mehrere Sprunganweisungen in einem Satz formuliert werden.
- Bei Programmen mit Sprunganweisungen ohne Sprungbedingungen muss das Programmende `M2`/`M30` nicht zwangsläufig am Programmende stehen.

## Beispiele

### Beispiel 1: Sprünge auf Sprungmarken

| **Programmcode** | **Kommentar** |
|---|---|
| N10 … | |
| N20 GOTOF Label_1 | ; Sprung in Richtung Programmende zur |
| | ; Sprungmarke "Label_1". |
| N30 … | |
| N40 Label_0: R1=R2+R3 | ; Sprungmarke "Label_0" gesetzt. |
| N50 … | |
| N60 Label_1: | ; Sprungmarke "Label_1" gesetzt. |
| N70 … | |
| N80 GOTOB Label_0 | ; Sprung in Richtung Programmanfang |
| | ; zur Sprungmarke "Label_0". |
| N90 … | |

### Beispiel 2: Indirekter Sprung auf Satznummer

| **Programmcode** | **Kommentar** |
|---|---|
| IF <Bedingung> == TRUE | |
| R10=100 | ; Sprungziel zuweisen |
| ELSE | |
| R10=110 | ; Sprungziel zuweisen |
| ENDIF | |
| ; Sprung in Richtung Programmende auf den Satz, dessen Satznummer in R10 steht | |
| N10 GOTOF "N"<<R10 | |
| ... | |
| N90 ... | |
| N100 ... | ; Sprungziel |
| N110 ... | |
| ... | |

### Beispiel 3: Sprung auf variables Sprungziel

```
Programmcode                          Kommentar
DEF STRING[20] ZIEL
IF <Bedingung> == TRUE
   ZIEL = "Marke1"                    ; Sprungziel zuweisen
ELSE
   ZIEL = "Marke2"                    ; Sprungziel zuweisen
ENDIF
; Sprung in Richtung Programmende zum variablen Sprungziel "Inhalt von ZIEL"
GOTOF ZIEL
Marke1: T="Bohrer1"                   ; Sprungziel 1
...
Marke2: T="Bohrer2"                   ; Sprungziel 2
...
```

### Beispiel 4: Sprung mit Sprungbedingung

```
Programmcode                                          Kommentar
N40 R1=30 R2=60 R3=10 R4=11 R5=50 R6=20               ; Zuweisung der Anfangswerte
N41 LA1: G0 X=R2*COS(R1)+R5 Y=R2*SIN(R1)+R6           ; Sprungmarke LA1
N42 R1=R1+R3 R4=R4-1
; IF Sprungbedingung == TRUE
; THEN Sprung in Richtung Programmanfang zur Sprungmarke LA1
N43 IF R4>0 GOTOB LA1
N44 M30                                               ; Programmende
```

## 3.1.5.3 Programmverzweigung (CASE ... OF ... DEFAULT ...)

Die CASE-Funktion bietet die Möglichkeit, den aktuellen Wert (Typ: INT) einer Variablen oder einer Rechenfunktion zu überprüfen und abhängig vom Ergebnis an unterschiedliche Stellen im Programm zu springen.

### Syntax

```
CASE(<Ausdruck>) OF <Konstante_1> GOTOF <Sprungziel_1> <Konstante_2>
GOTOF <Sprungziel_2> ... DEFAULT GOTOF <Sprungziel_n>
```

### Bedeutung

| | | |
|---|---|---|
| CASE: | Sprunganweisung | |
| <Ausdruck>: | Variable oder Rechenfunktion | |
| OF: | Schlüsselwort zur Formulierung der bedingten Programmverzweigungen | |
| <Konstante_1>: | Erster angegebener konstanter Wert für die Variable oder Rechenfunktion | |
| | Typ: | INT |

| | | |
|---|---|---|
| <Konstante_2>: | Zweiter angegebener konstanter Wert für die Variable oder Rechenfunktion | |
| | Typ: | INT |
| DEFAULT: | Für die Fälle, in denen die Variable oder Rechenfunktion keinen der angegebenen konstanten Werte annimmt, kann mit der Anweisung DEFAULT ein Sprungziel bestimmt werden.<br>**Hinweis:**<br>Falls die DEFAULT-Anweisung nicht programmiert ist, wird in diesen Fällen der auf die CASE-Anweisung folgende Satz zum Sprungziel. | |
| GOTOF: | Sprunganweisung mit Sprungziel in Richtung Programmende.<br>Statt GOTOF sind auch alle anderen GOTO-Befehle programmierbar (siehe Thema "Programmsprünge auf Sprungmarken"). | |
| <Sprungziel_1>: | Auf dieses Sprungziel wird verzweigt, wenn der Wert der Variablen oder Rechenfunktion der ersten angegebenen Konstanten entspricht.<br>Das Sprungziel kann wie folgt angegeben werden: | |
| | <Sprungmarke>: | Sprungziel ist die im Programm gesetzte Sprungmarke mit benutzerdefiniertem Namen: <Sprungmarke>: |
| | <Satznummer>: | Sprungziel ist eine Haupt- oder Nebensatznummer (z. B.: 200, N300) |
| | Variable vom Typ STRING: | Variables Sprungziel. Die Variable steht für eine Sprungmarke oder eine Satznummer. |
| <Sprungziel_2>: | Auf dieses Sprungziel wird verzweigt, wenn der Wert der Variablen oder Rechenfunktion der zweiten angegebenen Konstanten entspricht. | |
| <Sprungziel_n>: | Auf dieses Sprungziel wird verzweigt, wenn der Wert der Variablen keinen der angegebenen konstanten Werte annimmt. | |

## Beispiel

**Programmcode**

```
...
N20 DEF INT VAR1 VAR2 VAR3
N30 CASE(VAR1+VAR2-VAR3) OF 7 GOTOF Label_1 9 GOTOF La-
bel_2 DEFAULT GOTOF Label_3
N40 Label_1: G0 X1 Y1
N50 Label_2: G0 X2 Y2
N60 Label_3: G0 X3 Y3
...
```

Die CASE-Anweisung aus N30 definiert folgende Programmverzeigungsmöglichkeiten:

1. Wenn der Wert der Rechenfunktion VAR1+VAR2-VAR3 = 7, dann springe zu dem Satz mit Sprungmarkendefinition "Label_1" (→ N40).
2. Wenn der Wert der Rechenfunktion VAR1+VAR2-VAR3 = 9, dann springe zu dem Satz mit Sprungmarkendefinition "Label_2" (→ N50).
3. Wenn der Wert der Rechenfunktion VAR1+VAR2-VAR3 weder 7 noch 9 beträgt, dann springe zu dem Satz mit Sprungmarkendefinition "Label_3" (→ N60).

## 3.1.6 Programmteilwiederholung (REPEAT, REPEATB, ENDLABEL, P)

Die Programmteilwiederholung ermöglicht die Wiederholung bereits geschriebener Programmteile innerhalb eines Programms in beliebiger Zusammensetzung.

Die zu wiederholenden Programmzeilen bzw. Programmbereiche werden durch Sprungmarken (Labels) gekennzeichnet.

**Hinweis**

**Sprungmarken (Labels)**

Sprungmarken stehen immer am Anfang eines Satzes. Wenn eine Programmnummer vorhanden ist, steht die Sprungmarke unmittelbar nach der Satznummer.

Für die Benennung von Sprungmarken gelten folgende Regeln:

- Anzahl an Zeichen:
  - mindestens 2
  - höchstens 32
- Erlaubte Zeichen sind:
  - Buchstaben
  - Ziffern
  - Unterstriche
- Die ersten beiden Zeichen müssen Buchstaben oder Unterstriche sein.
- Nach dem Namen der Sprungmarke folgt ein Doppelpunkt (":").

### Syntax

**1. Einzelne Programmzeile wiederholen:**

```
<Sprungmarke>: ...
...
REPEATB <Sprungmarke> P=<n>
...
```

**2. Programmbereich zwischen Sprungmarke und REPEAT-Anweisung wiederholen:**

```
<Sprungmarke>: ...
...
REPEAT <Sprungmarke> P=<n>
...
```

**3. Bereich zwischen zwei Sprungmarken wiederholen:**

```
<Start-Sprungmarke>: ...
...
<End-Sprungmarke>: ...
...
REPEAT <Start-Sprungmarke> <End-Sprungmarke> P=<n>
```

```
...
```

**Hinweis**

Die REPEAT-Anweisung mit den beiden Sprungmarken zu klammern, ist nicht möglich. Wird die <Start-Sprungmarke> vor der REPEAT-Anweisung gefunden und wird die <End-Sprungmarke> nicht vor der REPEAT-Anweisung erreicht, dann wird die Wiederholung zwischen <Start-Sprungmarke> und REPEAT-Anweisung durchgeführt.

**4. Bereich zwischen Sprungmarke und ENDLABEL wiederholen:**

```
<Sprungmarke>: ...
...
ENDLABEL: ...
...
REPEAT <Sprungmarke> P=<n>
...
```

**Hinweis**

Die REPEAT-Anweisung mit der <Sprungmarke> und dem ENDLABEL zu klammern, ist nicht möglich. Wird die <Sprungmarke> vor der REPEAT-Anweisung gefunden und wird ENDLABEL nicht vor der REPEAT-Anweisung erreicht, dann wird die Wiederholung zwischen <Sprungmarke> und REPEAT-Anweisung durchgeführt.

## Bedeutung

| | |
|---|---|
| REPEATB: | Befehl zum Wiederholen einer Programmzeile |
| REPEAT: | Befehl zum Wiederholen eines Programmbereichs |
| <Sprungmarke>: | Die <Sprungmarke> kennzeichnet:<br>• die zu wiederholende Programmzeile (bei REPEATB) bzw.<br>• den Beginn des zu wiederholenden Programmbereichs (bei REPEAT) |
| | Die mit der <Sprungmarke> gekennzeichnete Programmzeile kann vor oder nach der REPEAT-/REPEATB-Anweisung stehen. Gesucht wird zunächst in Richtung Programmanfang. Wird die Sprungmarke in dieser Richtung nicht gefunden, dann wird in Richtung Programmende gesucht.<br>**Ausnahme:**<br>Wenn der Programmbereich zwischen Sprungmarke und REPEAT-Anweisung wiederholt werden soll (siehe 2. unter Syntax), dann muss die mit der <Sprungmarke> gekennzeichnete Programmzeile **vor** der REPEAT-Anweisung stehen, da in diesem Fall **nur** in Richtung Programmanfang gesucht wird.<br>Enthält die Zeile mit der <Sprungmarke> weitere Anweisungen, so werden diese bei jeder Wiederholung erneut ausgeführt. |

| | |
|---|---|
| ENDLABEL: | Schlüsselwort, welches das Ende eines zu wiederholenden Programmbereichs markiert<br>Enthält die Zeile mit dem ENDLABEL weitere Anweisungen, so werden diese bei jeder Wiederholung erneut ausgeführt.<br>ENDLABEL kann mehrfach im Programm verwendet werden. |
| P: | Adresse zur Angabe der Wiederholungsanzahl |
| <n>: | Anzahl an Programmteilwiederholungen |
| | Typ: INT |
| | Der zu wiederholende Programmteil wird <n> mal wiederholt. Nach der letzten Wiederholung wird das Programm mit der auf die REPEAT-/REPEATB-Zeile folgenden Zeile fortgesetzt.<br>**Hinweis:**<br>Ist kein P=<n> angegeben, wird der zu wiederholende Programmteil genau einmal wiederholt. |

## Beispiele

### Beispiel 1: Einzelne Programmzeile wiederholen

```
Programmcode                          Kommentar
N10 POSITION1: X10 Y20
N20 POSITION2: CYCLE(0,,9,8)          ; Positionszyklus
N30 ...
N40 REPEATB POSITION1 P=5             ; Führe SATZ N10 fünfmal aus.
N50 REPEATB POSITION2                 ; Führe Satz N20 einmal aus.
N60 ...
N70 M30
```

### Beispiel 2: Programmbereich zwischen Sprungmarke und REPEAT-Anweisung wiederholen

```
Programmcode                  Kommentar
N5 R10=15
N10 Begin: R10=R10+1          ; Breite
N20 Z=10-R10
N30 G1 X=R10 F200
N40 Y=R10
N50 X=-R10
N60 Y=-R10
N70 Z=10+R10
N80 REPEAT BEGIN P=4          ; Führe Bereich N10 bis N70 viermal aus.
N90 Z10
N100 M30
```

### Beispiel 3: Bereich zwischen zwei Sprungmarken wiederholen

```
Programmcode                      Kommentar
N5 R10=15
```

| Programmcode | Kommentar |
|---|---|
| N10 Begin: R10=R10+1 | ; Breite |
| N20 Z=10-R10 | |
| N30 G1 X=R10 F200 | |
| N40 Y=R10 | |
| N50 X=-R10 | |
| N60 Y=-R10 | |
| N70 END: Z=10 | |
| N80 Z10 | |
| N90 CYCLE(10,20,30) | |
| N100 REPEAT BEGIN END P=3 | ; Führe Bereich N10 bis N70 dreimal aus. |
| N110 Z10 | |
| N120 M30 | |

### Beispiel 4: Bereich zwischen Sprungmarke und ENDLABEL wiederholen

| Programmcode | Kommentar |
|---|---|
| N10 G1 F300 Z-10 | |
| N20 BEGIN1: | |
| N30 X10 | |
| N40 Y10 | |
| N50 BEGIN2: | |
| N60 X20 | |
| N70 Y30 | |
| N80 ENDLABEL: Z10 | |
| N90 X0 Y0 Z0 | |
| N100 Z-10 | |
| N110 BEGIN3: X20 | |
| N120 Y30 | |
| N130 REPEAT BEGIN3 P=3 | ; Führe Bereich N110 bis N120 dreimal aus. |
| N140 REPEAT BEGIN2 P=2 | ; Führe Bereich N50 bis N80 zweimal aus. |
| N150 M100 | |
| N160 REPEAT BEGIN1 P=2 | ; Führe Bereich N20 bis N80 zweimal aus. |
| N170 Z10 | |
| N180 X0 Y0 | |
| N190 M30 | |

### Beispiel 5: Fräsbearbeitung, Bohrposition mit verschiedenen Technologien bearbeiten

| Programmcode | Kommentar |
|---|---|
| N10 ZENTRIERBOHRER() | ; Zentrierbohrer einwechseln. |
| N20 POS_1: | ; Bohrpositionen 1 |
| N30 X1 Y1 | |
| N40 X2 | |
| N50 Y2 | |
| N60 X3 Y3 | |

| Programmcode | Kommentar |
|---|---|
| N70 ENDLABEL: | |
| N80 POS_2: | ; Bohrpositionen 2 |
| N90 X10 Y5 | |
| N100 X9 Y-5 | |
| N110 X3 Y3 | |
| N120 ENDLABEL: | |
| N130 BOHRER() | ; Bohrer wechseln und Bohrzyklus. |
| N140 GEWINDE(6) | ; Gewindebohrer M6 einwechseln und Gewindezyklus. |
| N150 REPEAT POS_1 | ; Wiederhole Programmabschnitt ab POS_1 einmal bis ENDLABEL. |
| N160 BOHRER() | ; Bohrer wechseln und Bohrzyklus. |
| N170 GEWINDE(8) | ; Gewindebohrer M8 einwechseln und Gewindezyklus. |
| N180 REPEAT POS_2 | ; Wiederhole Programmabschnitt ab POS_2 einmal bis ENDLABEL. |
| N190 M30 | |

## Weitere Informationen

- Programmteilwiederholung kann geschachtelt aufgerufen werden. Jeder Aufruf belegt eine Unterprogrammebene.
- Ist während der Bearbeitung einer Programmteilwiederholung `M17` oder `RET` programmiert, so wird die Programmteilwiederholung abgebrochen. Das Programm wird mit dem auf die `REPEAT`-Zeile folgenden Satz fortgesetzt.
- In der aktuellen Programm-Anzeige wird die Programmteilwiederholung als eigene Unterprogrammebene angezeigt.
- Wird während der Programmteil-Bearbeitung Ebenenabbruch ausgelöst, so wird das Programm nach dem Aufruf der Programmteilbearbeitung fortgesetzt.
  Beispiel:

| Programmcode | Kommentar |
|---|---|
| N5 R10=15 | |
| N10 BEGIN: R10=R10+1 | ; Breite |
| N20 Z=10-R10 | |
| N30 G1 X=R10 F200 | |
| N40 Y=R10 | ; Ebenenabbruch |
| N50 X=-R10 | |
| N60 Y=-R10 | |
| N70 END: Z10 | |
| N80 Z10 | |
| N90 CYCLE(10,20,30) | |
| N100 REPEAT BEGIN END P=3 | |
| N120 Z10 | ; Programmbearbeitung fortsetzen. |
| N130 M30 | |

- Kontrollstrukturen und Programmteilwiederholung können kombiniert genutzt werden. Es sollte jedoch keine Überschneidungen geben. Eine Programmteilwiederholung sollte innerhalb eines Kontrollstruktur-Zweigs liegen bzw. eine Kontrollstruktur innerhalb einer Programmteilwiederholung.
- Bei der Mischung von Sprüngen und Programmteilwiederholung werden die Sätze rein sequentiell abgearbeitet. Erfolgt z. B. ein Sprung aus einer Programmteilwiederholung, so wird solange bearbeitet, bis das programmierte Programmteilende gefunden wird. Beispiel:

**Programmcode**

```
N10 G1 F300 Z-10
N20 BEGIN1:
N30 X=10
N40 Y=10
N50 GOTOF BEGIN2
N60 ENDLABEL:
N70 BEGIN2:
N80 X20
N90 Y30
N100 ENDLABEL: Z10
N110 X0 Y0 Z0
N120 Z-10
N130 REPEAT BEGIN1 P=2
N140 Z10
N150 X0 Y0
N160 M30
```

**Hinweis**

Die `REPEAT`-Anweisung sollte hinter den Verfahrsätzen stehen.

## 3.1.7 Kontrollstrukturen

Die Steuerung arbeitet die NC-Sätze standardmäßig in der programmierten Reihenfolge ab.

Diese Reihenfolge kann durch die Programmierung von alternativen Programmblöcken und Programmschleifen variiert werden. Die Programmierung dieser Kontrollstrukturen erfolgt mit den Schlüsselwörtern IF, ELSE, ENDIF, LOOP, FOR, WHILE und REPEAT.

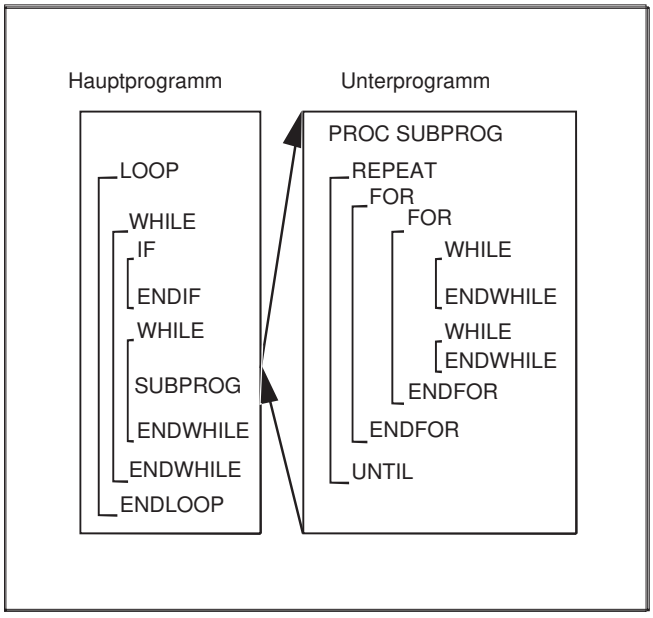
| ACHTUNG |
| --- |
| **Programmierfehler** |
| Kontrollstrukturen sind nur innerhalb des Anweisungsteils eines Programms möglich. Definitionen im Programmkopf können nicht bedingt oder wiederholt ausgeführt werden. |
| Schlüsselworte für Kontrollstrukturen dürfen ebenso wie Sprungziele nicht mit Makros überlagert werden. Eine Abprüfung bei der Makrodefinition findet nicht statt. |

## Wirksamkeit

Eine Kontrollstruktur kann nicht programmübergreifend verwendet werden.

## Schachtelungstiefe

Innerhalb jeder Unterprogrammebene ist eine Schachtelungstiefe von bis zu 16 Kontrollstrukturen möglich.




## Laufzeitverhalten

Im standardmäßig aktiven Interpreterbetrieb kann durch Verwendung von Programmsprüngen ein schnellerer Programmablauf als mit Kontrollstrukturen erreicht werden.

In vorkompilierten Zyklen ist kein Unterschied zwischen Programmsprüngen und Kontrollstrukturen vorhanden.

### Aktuelle Satzanzeige bei Programmschleifen

Werden innerhalb einer Programmschleife nur Vorlaufsätze ausgeführt, wird in der aktuellen Satzanzeige der letzte Hauptlaufsatz **vor** der Programmschleife angezeigt.

Damit z. B. zu Diagnosezwecken auch die abgearbeiteten Vorlaufsätze in der aktuellen Satzanzeige sichtbar werden, muss der Decodier-Einzelsatz SBL2 aktiviert werden.

### Schleifen ohne Hauptlaufsatz

Wird innerhalb einer Programmschleife kein Hauptlaufsatz programmiert, wird die Schleife so lange im Vorlauf bearbeitet, bis die Schleifenbedingung erfüllt ist.

Dadurch entsteht eine hohe Auslastung und die Anzeige kann beeinträchtigt werden.

Als Abhilfe bietet sich an, den Befehl STOPRE oder eine Verweilzeit G04 von 0 Sekunden **in** die Schleife einzufügen.

### Randbedingungen

- Sätze mit Kontrollstrukturelementen können nicht ausgeblendet werden.
- Sprungmarken (Labels) sind in Sätzen mit Kontrollstrukturelementen nicht erlaubt.
- Kontrollstrukturen werden interpretativ abgearbeitet. Bei Erkennen eines Schleifenendes wird unter Berücksichtigung der dabei gefundenen Kontrollstrukturen nach dem Schleifenanfang gesucht. Daher wird im Interpreterbetrieb die Blockstruktur eines Programms nicht komplett geprüft.
- Grundsätzlich empfiehlt sich, Kontrollstrukturen und Programmsprünge nicht gemischt zu verwenden.
- Bei Vorverarbeitung von Zyklen kann die korrekte Schachtelung von Kontrollstrukturen überprüft werden.

## 3.1.7.1 Bedingte Anweisung und Verzweigung (IF, ELSE, ENDIF)

#### Bedingten Anweisung: IF - Programmblock - ENDIF

Bei einer bedingten Anweisung wird der zwischen `IF` und `ENDIF` stehende Programmblock nur ausgeführt, wenn die Bedingung erfüllt ist.

#### Verzweigung: IF - Programmblock_1 - ELSE - Programmblock_2 - ENDIF

Bei einer Verzweigung wird immer einer von zwei Programmblöcken ausgeführt.

Ist die Bedingung erfüllt, wird der zwischen `IF` und `ELSE` stehende Programmblock_1 ausgeführt.

Ist die Bedingung **nicht** erfüllt, wird der zwischen `ELSE` und `ENDIF` stehende Programmblock_2 ausgeführt.

**Hinweis**

**ELSE in Synchronaktionen**

Das Schlüsselwort ELSE ist auch in Synchronaktionen programmierbar. Damit kann eine Synchronaktion um Aktionen erweitert werden, die bei nicht erfüllter Bedingung ausgeführt werden sollen.

## Syntax

### Bedingte Anweisung

```
IF <Bedingung>
   Programmblock                          ; Ausführung bei: <Bedingung> == TRUE
ENDIF
```

### Verzweigung

```
IF <Bedingung>
   Programmblock_1                        ; Ausführung bei: <Bedingung> == TRUE
ELSE
   Programmblock_2                        ; Ausführung bei: <Bedingung> == FALSE
ENDIF
```

## Bedeutung

| | |
|---|---|
| `IF`: | Leitet die bedingte Anweisung bzw. Verzweigung ein. |
| `ELSE`: | Leitet den alternativen Programmblock ein. |
| `ENDIF`: | Markiert das Ende der bedingte Anweisung bzw. Verzweigung. |
| `<Bedingung>`: | Logischer Ausdruck, dessen Auswertung TRUE oder FALSE ergibt. |

## Beispiel: Werkzeugwechselunterprogramm

| **Programmcode** | **Kommentar** |
|---|---|
| `PROC L6` | `Werkzeugwechselroutine` |
| `N500 DEF INT TNR_AKTUELL` | `Variable für aktive T-Nummer` |
| `N510 DEF INT TNR_VORWAHL` | `Variable für vorgewählte T-Nummer` |
| | `Aktuelles Werkzeug ermitteln` |
| `N520 STOPRE` | |
| `N530 IF $P_ISTEST` | `Im Programmtest-Betrieb wird ...` |
| `N540    TNR_AKTUELL = $P_TOOLNO` | `... aus dem Programmkontext das "aktuelle" Werkzeug gelesen.` |
| `N550 ELSE` | `Andernfalls wird ...` |

| **Programmcode** | **Kommentar** |
|---|---|
| N560 TNR_AKTUELL = $TC_MPP6[9998,1] | ... das Werkzeug der Spindel ausgelesen. |
| N570 ENDIF | |
| N580 GETSELT(TNR_VORWAHL) | T-Nummer des vorgewählten Werkzeugs auf der Spindel lesen. |
| N590 IF TNR_AKTUELL <> TNR_VORWAHL | Wenn das vorgewählte Werkzeug noch nicht das aktuelle Werkzeug ist, dann ... |
| N600 G0 G40 G60 G90 SUPA X450 Y300 Z300 D0 | ... Werkzeugwechselpunkt anfahren ... |
| N610 M206 | ... und Werkzeugwechsel ausführen. |
| N620 ENDIF | |
| N630 M17 | |

### 3.1.7.2 Endlos-Programmschleife (LOOP, ENDLOOP)

Die Endlos-Schleife findet Verwendung in Endlos-Programmen. Am Schleifenende findet immer wieder der Rücksprung zum Schleifenanfang statt.

## Syntax

```
LOOP
...
ENDLOOP
```

## Bedeutung

| | |
|---|---|
| LOOP: | Leitet die Endlosschleife ein. |
| ENDLOOP: | Markiert das Ende der Schleife und bewirkt Rücksprung auf den Schleifenanfang. |

## Beispiel

| **Programmcode** |
|---|
| ... |
| LOOP |
| MSG("keine Werkzeugschneide aktiv") |
| M0 |
| STOPRE |
| ENDLOOP |
| ... |

### 3.1.7.3 Zählschleife (FOR ... TO ..., ENDFOR)

Die Zählschleife wird verwendet, wenn ein Arbeitsablauf mit einer festen Anzahl von Durchläufen wiederholt werden soll.

## Syntax

```
FOR <Variable> = <Anfangswert> TO <Endwert>
...
ENDFOR
```

## Bedeutung

| | | |
|---|---|---|
| FOR: | Leitet die Zählschleife ein. | |
| ENDFOR: | Markiert das Ende der Schleife und bewirkt Rücksprung auf den Schleifenanfang, solange der Endwert der Zählung noch nicht erreicht ist. | |
| <Variable>: | Zählvariable, die vom Anfangs- bis zum Endwert hochgezählt wird und sich bei jedem Durchlauf um den Wert "1" erhöht. | |
| | Typ | INT oder REAL<br>**Hinweis:**<br>Der Typ REAL wird genommen, wenn z. B. R-Parameter für eine Zählschleife programmiert werden. Ist die Zählvariable vom Typ REAL, wird ihr Wert auf einen ganzzahligen Wert gerundet. |
| <Anfangswert>: | Anfangswert der Zählung<br>Bedingung: Der Anfangswert muss kleiner sein als der Endwert. | |
| <Endwert>: | Endwert der Zählung | |

## Beispiele

#### Beispiel 1: INTEGER-Variable oder R-Parameter als Zählvariable

INTEGER-Variable als Zählvariable:

```
Programmcode                          Kommentar
DEF INT iVARIABLE1
R10=R12-R20*R1 R11=6
FOR iVARIABLE1= R10 TO R11            ; Zählvariable = INTEGER-Variable
 R20=R21*R22+R33
ENDFOR
M30
```

R-Parameter als Zählvariable:

```
Programmcode                          Kommentar
R11=6
FOR R10=R12-R20*R1 TO R11             ; Zählvariable = R-Parameter (Realvariable)
```

| Programmcode | Kommentar |
|---|---|
| R20=R21*R22+R33 | |
| ENDFOR | |
| M30 | |

**Beispiel 2: Fertigung einer festen Teilestückzahl**

| Programmcode | Kommentar |
|---|---|
| DEF INT STUECKZAHL | ; Definiert Variable vom Typ INT mit Namen "STUECKZAHL". |
| FOR STUECKZAHL = 0 TO 100 | ; Leitet die Zählschleife ein. Die Variable "STUECKZAHL" wird vom Anfangswert "0" bis zum Endwert "100" hochgezählt. |
| G01 … | |
| ENDFOR | ; Ende der Zählschleife. |
| M30 | |

### 3.1.7.4 Programmschleife mit Bedingung am Schleifenanfang (WHILE, ENDWHILE)

Bei einer WHILE-Schleife steht die Bedingung am Schleifenanfang. Solange die Bedingung erfüllt ist, wird die WHILE-Schleife durchlaufen.

## Syntax

```
WHILE <Bedingung>
...
ENDWHILE
```

## Bedeutung

| | |
|---|---|
| WHILE: | Leitet die Programmschleife ein. |
| ENDWHILE: | Markiert das Ende der Schleife und bewirkt Rücksprung auf den Schleifenanfang. |
| <Bedingung>: | Bedingung, die erfüllt sein muss, damit die WHILE-Schleife durchlaufen wird. |

## Beispiel

| Programmcode | Kommentar |
|---|---|
| ... | |
| WHILE $AA_IW[BOHRACHSE] > -10 | ; Aufruf der WHILE-Schleife unter folgender Bedingung: der aktuelle WKS-Sollwert für die Bohrachse muss größer -10 sein. |
| G1 G91 F250 AX[BOHRACHSE] = -1 | |
| ENDWHILE | |

| Programmcode | Kommentar |
|---|---|
| ... | |

### 3.1.7.5 Programmschleife mit Bedingung am Schleifenende (REPEAT, UNTIL)

Bei einer REPEAT-Schleife steht die Bedingung am Schleifenende. Die REPEAT-Schleife wird einmal durchlaufen und solange wiederholt, bis die Bedingung erfüllt ist.

#### Syntax

```
REPEAT
...
UNTIL <Bedingung>
```

#### Bedeutung

| | |
|---|---|
| REPEAT: | Leitet die Programmschleife ein. |
| UNTIL: | Markiert das Ende der Schleife und bewirkt Rücksprung auf den Schleifenanfang. |
| <Bedingung>: | Bedingung, die erfüllt sein muss, damit die REPEAT-Schleife nicht mehr durchlaufen wird. |

#### Beispiel

| Programmcode | Kommentar |
|---|---|
| ... | |
| REPEAT | ; Aufruf der REPEAT-Schleife. |
| ... | |
| UNTIL ... | ; Prüfung, ob Bedingung erfüllt ist. |
| ... | |

### 3.1.7.6 Programmbeispiel mit verschachtelten Kontrollstrukturen

| Programmcode | Kommentar |
|---|---|
| LOOP | |
| IF NOT $P_SEARCH | ; IF kein Satzsuchlauf |
| G1 G90 X0 Z10 F1000 | |
| WHILE $AA_IM[X] <= 100 | ; WHILE (Sollwert X-Achse <= 100) |
| G1 G91 X10 F500 | ; Bohrbild |
| Z-5 F100 | |
| Z5 | |
| ENDWHILE | |
| ELSE | ; ELSE Satzsuchlauf |

| Programmcode | Kommentar |
|---|---|
| MSG("Im Suchlauf wird nicht gebohrt") | |
| ENDIF | ; ENDIF |
| $A_OUT[1] = 1 | ; nächste Bohrplatte |
| G4 F2 | |
| ENDLOOP | |
| M30 | |

## 3.1.8 Kanalübergreifende Programmkoordinierung (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM)

Ein Kanal der NC kann prinzipiell das in ihm gestartete Programm unabhängig von anderen Kanälen seiner Betriebsartengruppe (BAG) abarbeiten. Sind aber gleichzeitig mehrere Programme in mehreren Kanälen der BAG an der Fertigung eines Werkstücks beteiligt, müssen die Programmabläufe mit den nachfolgenden Koordinierungsbefehlen in den unterschiedlichen Kanälen koordiniert werden.

### Voraussetzung

Alle an der Programmkoordinierung beteiligten Kanäle müssen zur **selben** Betriebsartengruppe (BAG) gehören:

MD10010 $MC_ASSIGN_CHAN_TO_MODE_GROUP[<Kanal>] = <BAG-Nummer>

### Kanalname statt Kanalnummer

Als Parameter der vordefinierten Prozeduren der Programmkoordinierung können anstelle von Kanalnummern auch die in MD20000 $MC_CHAN_NAME[<Kanalindex>] eingetragenen Kanalnamen verwendet werden. Die Verwendung der Kanalnamen in NC-Programmen muss zuerst freigegeben werden:

MD10280 $MN_PROG_FUNCTION_MASK, Bit 1 = TRUE

### Hinweis

### Mindestabstand zwischen Befehlen

Zwischen den Befehlen `INIT`, `START`, `WAITE`, `WAITM`, `SETM`, `CLEARM` und dem Befehl `WAITMC` müssen mindestens zwei Verfahrsätze Abstand gehalten werden. WAITMC ist ein ausführbarer Satz, wird aber zur Optimierung in den vorherigen Satz verschoben und dann als Satz gelöscht. `SETM` als Beispiel ist kein ausführbarer Satz und wird in den nächsten Satz verschoben, so dass bei einem Satz Abstand beide Befehle im mittleren Satz stehen würden. Da nur ein Satz möglich ist, wird bei einem Satz Abstand für `WAITMC` die Optimierung nicht durchgeführt.

Damit bremst das Programm ab und die Bearbeitung wird kurz angehalten.

### Syntax

```
INIT(<ChanNr>, <Prog>, <AckMode>)
```

```
START(<ChanNr>, <ChanNr>, ...)
WAITM(<MarkNr>, <ChanNr>, <ChanNr>, ...)
WAITE(<ChanNr>, <ChanNr>, ...)
WAITMC(<MarkNr>, <ChanNr>, <ChanNr>, ...)
SETM(<MarkNr>, <MarkNr>, ...)
CLEARM(<MarkNr>, <MarkNr>, ...)
```

## Bedeutung

| | | |
|---|---|---|
| `INIT()`: | Vordefinierte Prozedur zur Anwahl des NC-Programms, das im angegebenen Kanal abgearbeitet werden soll | |
| `START()`: | Vordefinierte Prozedur zum Starten des im jeweiligen Kanal angewählten Programms | |
| `WAITM()`: | Vordefinierte Prozedur zum Warten auf das Erreichen einer Wartemarke in den angegebenen Kanälen<br>Im eigenen Kanal wird die angegebene Wartemarke durch `WAITM` gesetzt. Der vorhergehenden Satz wird mit Genauhalt beendet. Die Wartemarke wird nach Synchronisation gelöscht.<br>Gleichzeitig können max. 10 Marken pro Kanal gesetzt werden. | |
| `WAITE()`: | Vordefinierte Prozedur zum Warten auf das Programmende in einem oder mehreren anderen Kanälen | |
| `WAITMC()`: [1)] | Vordefinierte Prozedur zum Warten auf das Erreichen einer Wartemarke in den angegebenen Kanälen<br>Im Gegensatz zu `WAITM` wird das Abbremsen der Achsen auf Genauhalt nur eingeleitet, wenn die anderen Kanäle die Wartemarke noch nicht erreicht haben. | |
| `SETM()`: [1)] | Vordefinierte Prozedur zum Setzen einer oder mehrerer Wartemarken für die Kanalkoordinierung<br>Die Bearbeitung im eigenen Kanal wird davon nicht beeinflusst.<br>`SETM` behält seine Gültigkeit über Kanal-Reset und NC-Start hinweg. | |
| `CLEARM()`: [1)] | Vordefinierte Prozedur zum Löschen einer oder mehrerer Wartemarken der Kanalkoordinierung<br>Die Bearbeitung im eigenen Kanal wird davon nicht beeinflusst.<br>`CLEARM()` löscht alle Wartemarken im Kanal.<br>`CLEARM(0)` löscht nur die Wartemarke "0".<br>`CLEARM` behält seine Gültigkeit über Kanal-Reset und NC-Start hinweg. | |
| `<ChanNr>`: | Kanalnummer<br>Die Nummer des eigenen Kanals muss nicht angegeben werden. | |
| | Typ: | INT |
| `<Prog>`: | Absolute oder relative Pfadangabe (**optional**) + Programmname | |
| | Typ: | STRING |
| | Informationen zur Pfadangabe siehe Programmierhandbuch NC-Programmierung, Kapitel "Adressierung von Dateien des Programmspeichers" (Seite 563). | |

| | | | |
|---|---|---|---|
| <AckMode>: | Quittierungsmodus (**optional**) | | |
| | Typ: | CHAR | |
| | Werte: | "N" | Ohne Quittung<br>Die Programmbearbeitung wird nach Absenden des Kommandos fortgeführt. Der Absender wird nicht benachrichtigt, wenn das Kommando nicht erfolgreich ausgeführt werden kann. |
| | | "S" | Synchrone Quittung<br>Die Programmabarbeitung wird solange angehalten, bis die Empfängerkomponente das Kommando quittiert hat. Bei positiver Quittung wird der nächste Befehl abgearbeitet. Bei negativer Quittung wird eine Fehlermeldung ausgegeben. |
| <MarkNr>: | Nummer der Wartemarke<br>**Hinweis**<br>In einem mehrkanaligen System stehen maximal 100 Wartemarken zur Verfügung (Wartemarke 0 ... 99).<br>In einem einkanaligen System steht nur die Wartemarke 0 zur Verfügung. | | |
| 1) Zur anwenderspezifischen Kommunikation und / oder Koordination von Kanälen können Wartemarken mittels SETM / CLEARM auch ohne Verwendung des bedingten Wartebefehles WAITMC eingesetzt werden. Die Wartemarken behalten dabei über Kanal-Reset und NC-Start hinaus ihre Werte bei. | | | |

## Beispiele

### START mittels Kanalnamen aus MD20000

- Parametrierung
```
MD10280 $MN_PROG_FUNCTION_MASK, Bit 1 = TRUE
$MC_CHAN_NAME[ 0 ] = "BEARBEITUNG"  ; Name für Kanal 1
$MC_CHAN_NAME[ 1 ] = "ZUFUEHRUNG"   ; Name für Kanal 2
```
- Programmierung

| **Programmcode** | **Kommentar** |
|---|---|
| START(BEARBEITUNG) | ; Start von Kanal 1 |
| START(ZUFUEHRUNG) | ; Start von Kanal 2 |

### START über lokale "Kanalnamen" und Anwendervariable

| **Programmcode** | **Kommentar** |
|---|---|
| DEF INT MASCHINE = 1 | ; Definition Anwendervariable für Kanal 1 |
| DEF INT LADER = 2 | ; Definition Anwendervariable für Kanal 2 |
| ... | |
| START(MASCHINE) | ; Start von Kanal 1 |
| START(LADER) | ; Start von Kanal 2 |

### START über lokale "Kanalnamen", Anwendervariable und parametrierte Kanalnamen

| **Programmcode** | **Kommentar** |
|---|---|
| DEF INT chanNo1 | ; Definition Anwendervariable für Kanal 1 |
| DEF INT chanNo2 | ; Definition Anwendervariable für Kanal 2 |

| **Programmcode** | **Kommentar** |
|---|---|
| chanNo1 = CHAN_1 | ; Zuweisung parametrierter Kanalnamen Kanal 1 |
| chanNo2 = CHAN_2 | ; Zuweisung parametrierter Kanalnamen Kanal 2 |
| ... | |
| START(chanNo1) | ; Start von Kanal 1 |
| START(chanNo2) | ; Start von Kanal 2 |

### INIT-Befehl mit absoluter Pfadangabe

Anwahl von Programm /_N_MPF_DIR/_N_ABSPAN1_MPF in Kanal 2.

**Programmcode**

```
INIT(2,"/_N_WKS_DIR/_N_WELLE1_WPD/_N_ABSPAN1_MPF")
```

### INIT-Befehl mit Programmnamen

Anwahl des Programms mit dem Namen "MYPROG". Die Steuerung sucht das Programm anhand des Suchpfades.

**Programmcode**

```
INIT(2,"MYPROG")
```

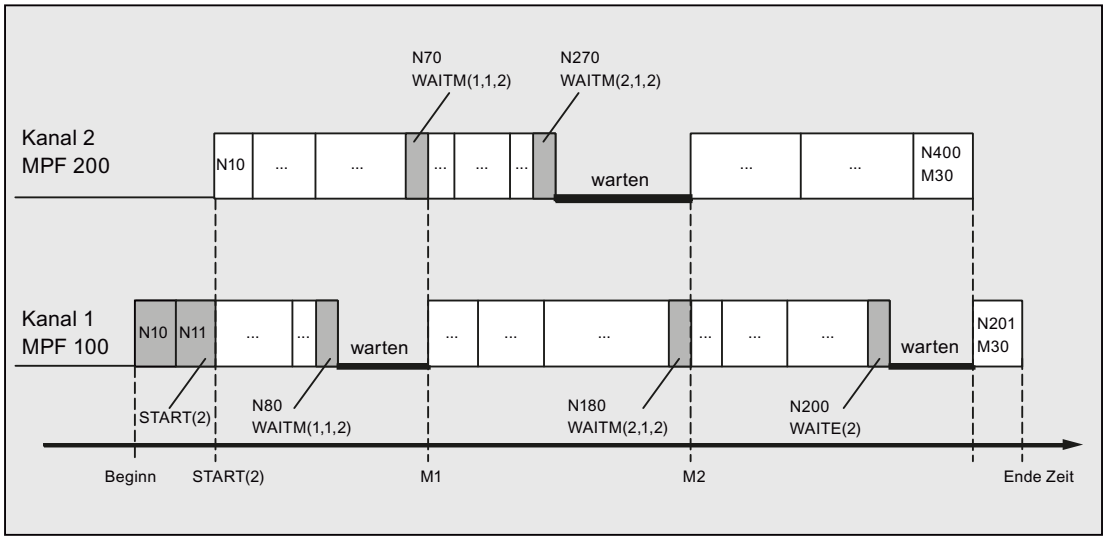
### Programmkoordinierung mit WAITM

- Kanal 1: Das Programm /_N_MPF_DIR/_N_MPF100_MPF ist bereits angewählt. und gestartet.

| **Programmcode** | **Kommentar** |
|---|---|
| | ; Programm MPF100 |
| N10 INIT(2,"MPF200","N") | ; Anwahl Programm MPF200, Kanal 2 |
| N11 START(2) | ; Start von Kanal 2 |
| ... | |
| N80 WAITM(1,1,2) | ; Warten auf WAIT-Marke 1 in Kanal 1 und 2 |
| N81 ... | ; Kanal 1, N81 und Kanal 2, N71 werden<br>; synchron begonnen |
| ... | |
| N180 WAITM(2,1,2) | ; Warten auf WAIT-Marke 2 in Kanal 1 und 2 |
| N181 ... | ; Kanal 1, N181 und Kanal 2, N271 werden<br>; synchron begonnen |
| ... | |
| N200 WAITE(2) | ; Warten auf Programmende in Kanal 2 |
| N201 ... | ; N201 wird erst nach dem Programmende von<br>; MPF200 in Kanal 2 begonnen |
| N201 M30 | ; Programmende Kanal 1 |

- Kanal 2: In Kanal 1 wird über Satz N10 und N20 das Programm MPF200_MPF für Kanal 2 angewählt und gestartet.

| **Programmcode** | **Kommentar** |
|---|---|
| ;$PATH=/_N_MPF_DIR | ; Programm MPF200 |
| ... | |
| N70 WAITM(1,1,2) | Warten auf WAIT-Marke 1 in Kanal 1 und 2 |
| N71 ... | ; Kanal 1, N81 und Kanal 2, N71 werden<br>; synchron begonnen |
| ... | |
| N270 WAITM(2,1,2) | Warten auf WAIT-Marke 2 in Kanal 1 und 2 |
| N271 ... | ; Kanal 1, N181 und Kanal 2, N271 werden<br>; synchron begonnen |
| ... | |
| N400 M30 | Programmende Kanal 2 |




## Randbedingungen

### Nicht synchroner Beginn des Abarbeitens von Folgesätzen nach WAIT-Marken

Bei einer Kanalkoordinierung mittels WAIT-Marken kann es zu einem nicht synchronen Beginn des Abarbeitens der Folgesätze kommen. Dieses Verhalten tritt auf, wenn unmmittelbar vor Erreichen der gemeinsamen WAIT-Marke in einem der zu synchronisierenden Kanäle eine Aktion ausgelöst wird, die in diesem Restweglöschen mit implizitem Repositionieren (REPOSA) zur Folge hat.

Annahme: Aktuelle Achszuordnung in den Kanälen 1 und 2

- Kanal 1: Achsen X1 und U
- Kanal 2: Achse X2

Tabelle 3-2 Zeitlicher Abblauf in den Kanälen 1 und 2

| Kanal 1 | Kanal 2 | Beschreibung |
|---|---|---|
| ... | ... | Beliebige Bearbeitung in Kanal 1 und 2 |
| N100<br>WAITM(20,1,2) | | Kanal 1: erreicht die WAIT-Marke und wartet auf Synchronisation mit Kanal 2 |
| *Beginn der GETD(U) Bearbeitung:*<br>• *Achstausch*<br>• *Restweglöschen*<br>• *REPOSA*<br>*Ende* | N200 GETD(U) | Kanal 2: fordert die Achse U aus Kanal 1 an<br>Kanal 1: Bearbeitung von GET(U) im Hintergrund |
| | N210<br>WAITM(20,1,2) | Kanal 2: erreicht die WAIT-Marke. ⇒ Die Synchronisation der Kanäle 1 und 2 ist damit abgeschlossen |
| | N220 G0 X2=100 | Kanal 2: Beginn des Abarbeitens von N220 |
| N110 G0 X1=100 | | Kanal 1: Zeitversetzter Beginn des Abarbeitens von N110 |

## 3.1.9 Makrotechnik (DEFINE ... AS)

| ACHTUNG |
| --- |
| **Makrotechnik erhöht die Komplexität der Programmierung** |
| Mit Makrotechnik kann die Programmiersprache der Steuerung stark verändert werden. Makrotechnik darf nur mit großer Sorgfalt eingesetzt werden. |

Als Makro bezeichnet man die Zusammenfassung von einzelnen Anweisungen zu einer neuen Gesamtanweisung mit eigenem Namen. Bei Aufruf des Makros im Programmablauf werden die unter dem Makronamen programmierten Anweisungen nacheinander abgearbeitet.

Entsprechend dem Gültigkeitsbereich, d. h. dem Bereich in dem die Makrodefinition wirksam ist, gibt es folgende Kategorien von Makros:

- Lokale Makros
  Lokale Makros sind Makros, die am Programmanfang eines NC-Programms definiert sind, das zum Zeitpunkt der Abarbeitung nicht das Hauptprogramm ist. Sie werden beim Aufruf des NC-Programms angelegt und mit Programmende-Reset bzw. dem nächsten Steuerungshochlauf gelöscht. Auf lokale Makros kann nur innerhalb des NC-Programms zugegriffen werden, in dem sie definiert sind.
- Programmglobale Makros
  Programmglobale Makros sind Makros, die am Programmanfang eines als Hauptprogramm verwendeten NC-Programms definiert sind. Sie werden beim Aufruf des NC-Programms angelegt und mit Programmende-Reset bzw. dem nächsten Steuerungshochlauf gelöscht. Auf programmglobale Makros kann im Hauptprogramm und in allen Unterprogrammen zugegriffen werden.

  **Hinweis**

  **Verfügbarkeit von programmglobalen Makros**

  Im Hauptprogramm definierte programmglobale Makros sind nur dann auch in den Unterprogrammen verfügbar, wenn folgendes Maschinendatum gesetzt ist:

  MD11120 $MN_LUD_EXTENDED_SCOPE = 1

  Mit MD11120 = 0 sind die im Hauptprogramm definierten programmglobalen Makros nur im Hauptprogramm verfügbar.

- Globale Makros
  Globale Makros sind NC- bzw. Kanal-globale Makros, die in einer Definitionsdatei (Makrodatei) definiert sind und auch nach Programmende-Reset bzw. dem nächsten Steuerungshochlauf erhalten bleiben. Globale Makros können in jedem beliebigen Haupt- oder Unterprogramm aufgerufen und abgearbeitet werden.

  **Hinweis**

  Um die Makros einer **externen** Makrodatei im NC-Programm verwenden zu können, muss die Makrodatei in die NC geladen werden.

Makros müssen vor ihrer Verwendung definiert worden sein. Folgende Regeln sind dabei zu beachten:

- Im Makro können beliebige Bezeichner, G-, M-, H-Funktionen und L-Unterprogrammnamen definiert werden.
- Die Makrodefinition kann am Programmanfang oder in einer eigenen Definitionsdatei (Makrodatei) erfolgen.
- Lokale und programmglobale Makros werden am Programmanfang definiert.
- Globale Makros müssen in einer Makrodatei, z. B. _N_DEF_DIR/_N_UMAC_DEF, definiert werden.
- G-Befehls-Makros können nur als globale Makros definiert werden.
- H- und L-Funktionen sind 2-stellig programmierbar.
- M- und G-Befehle können 3-stellig programmiert werden.

**Hinweis**

Schlüsselworte und reservierte Namen dürfen nicht mit Makros überschrieben werden. Das gilt auch für alle Sprungziele innerhalb eines GOTO Befehls und für die Schlüsselworte in Programmschleifen wie FOR, WHILE, LOOP, REPEAT.

## Syntax

Makro-Definition:
```
DEFINE <Macro_Name> AS <Operation_1> <Operation_2> ...
```

Aufruf im NC-Programm:
```
<Macro_Name>
```

## Bedeutung

| | |
|---|---|
| `DEFINE ... AS`: | Schlüsselwort-Kombination zur Definition eines Makros |
| `<Macro_Name>`: | Name des Makros<br>Als Makronamen sind nur Bezeichner zulässig.<br>Mit dem Makronamen wird das Makro aus dem NC-Programm heraus aufgerufen. |
| `<Operation_1>`: | Erste Programmieranweisung im Makro |
| `<Operation_2>`: | Zweite Programmieranweisung im Makro |

## Beispiele

### Beispiel 1: Makrodefinition am Programmanfang

| **Programmcode** | **Kommentar** |
|---|---|
| `DEFINE LINIE AS G1 G94 F300` | `; Makro-Definition` |
| `...` | |

| Programmcode | Kommentar |
|---|---|
| N70 LINIE X10 Y20 | ; Makro-Aufruf |
| ... | |

### Beispiel 2: Makrodefinitionen in einer Makrodatei

| Programmcode | Kommentar |
|---|---|
| DEFINE M6 AS L6 | ; Beim Werkzeugwechsel wird ein Unterprogramm aufgerufen, das den nötigen Datentransfer übernimmt. Im Unterprogramm wird die eigentliche Werkzeugwechsel-M-Funktion ausgegeben (z. B. M106). |
| DEFINE G81 AS DRILL(81) | ; Nachbildung des DIN-G-Befehls. |
| DEFINE G33 AS M333 G333 | ; Beim Gewindeschneiden wird Synchronisation mit der PLC angefordert. Der ursprüngliche G-Befehl G33 wurde per MD in G333 umbenannt, die Programmierung bleibt für den Anwender gleich. |

### Beispiel 3: Externe Makrodatei

Nach dem Einlesen der externen Makrodatei in die Steuerung muss die Makrodatei in die NC geladen werden. Erst dann können die Makros im NC-Programm verwendet werden.

| Programmcode | Kommentar |
|---|---|
| %_N_UMAC_DEF | |
| ;$PATH=/_N_DEF_DIR | ; Kundenspezifische Makros |
| DEFINE PI AS 3.14 | |
| DEFINE TC1 AS M3 S1000 | |
| DEFINE M13 AS M3 M7 | ; Spindel rechts, Kühlmittel ein |
| DEFINE M14 AS M4 M7 | ; Spindel links, Kühlmittel ein |
| DEFINE M15 AS M5 M9 | ; Spindel Halt, Kühlmittel aus |
| DEFINE M6 AS L6 | ; Aufruf des Werkzeugwechselprogramms |
| DEFINE G80 AS MCALL | ; Abwahl Bohrzyklus |
| M30 | |

# 3.2 Unterprogrammtechnik

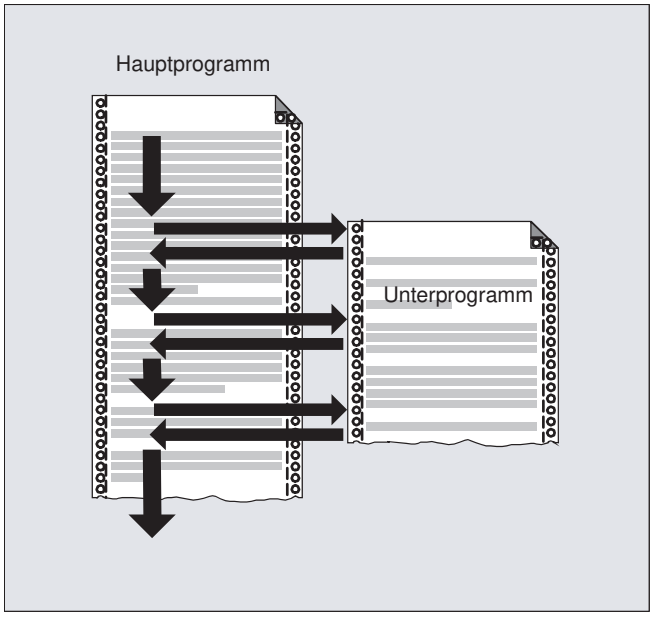
## 3.2.1 Allgemeines

### 3.2.1.1 Unterprogramm

Die Bezeichnung "Unterprogramm" stammt noch aus der Zeit, als Teileprogramme fest in Haupt- und Unterprogramme unterteilt waren. Hauptprogramme waren dabei die Teileprogramme, die an der Steuerung zum Abarbeiten angewählt und dann gestartet wurden. Unterprogramme waren die Teileprogramme, die vom Hauptprogramm aus aufgerufen wurden.

Diese feste Einteilung besteht mit der heutigen SINUMERIK NC-Sprache nicht mehr. Jedes Teileprogramm kann prinzipiell als Hauptprogramm angewählt und gestartet oder als Unterprogramm von einem anderen Teileprogramm aus aufgerufen werden.

Somit wird im weiteren Verlauf mit Unterprogramm ein Teileprogramm bezeichnet, das von einem anderen Teileprogramm aus aufgerufen wird.




### Anwendung

Wie in allen höheren Programmiersprachen werden auch in der NC-Sprache Unterprogramme dazu angewandt, um Programmteile, die mehrfach verwendet werden, in eigenständige, in sich abgeschlossene Programme auszulagern.

Unterprogrammen bieten folgende Vorteile:

- Erhöhen die Übersichtlichkeit und Lesbarkeit der Programme
- Erhöhen die Qualität durch Wiederverwendung getesteter Programmteile

- Bieten die Möglichkeit zur Schaffung spezifischer Bearbeitungsbibliotheken
- Sparen Speicherplatz

### 3.2.1.2 Unterprogrammnamen

#### Benennungregeln

Der Unterprogrammname kann unter Einhaltung folgender Regeln frei gewählt werden:

- Erlaubte Zeichen:
  - Buchstaben: A ... Z, a ... z
  - Ziffern: 0 ... 9
  - Unterstrich: _
- Die ersten beiden Zeichen sollten zwei Buchstaben oder ein Unterstrich gefolgt von einem Buchstaben sein.

  **Hinweis**

  Wenn diese Bedingung erfüllt ist, dann kann ein NC-Programm allein durch Angabe des Programmnamens als Unterprogramm aus einem anderen Programm heraus aufgerufen werden. Beginnt der Programmname hingegen mit Ziffern, dann ist der Unterprogramm-Aufruf nur über die CALL-Anweisung möglich.

- Maximale Länge: 24 Zeichen

**Hinweis**

**Groß- / Kleinbuchstaben**

In der SINUMERIK NC-Sprache wird **nicht** zwischen Groß- und Kleinbuchstaben unterschieden.

**Hinweis**

**Unzulässige Programmnamen**

Um Probleme mit Windows-Applikationen zu vermeiden, dürfen folgende Programmnamen **nicht** verwendet werden:

- CON, PRN, AUX, NUL
- COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9
- LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9

### Steuerungsinterne Erweiterungen

Der bei der Programmerstellung vergebene Unterprogrammname wird steuerungsintern mit einem Pre- und Postfix erweitert:

- Prefix: `_N_`
- Postfix: `_SPF`

#### Verwendung des Programmnamens

Bei der Verwendung des Programmnamens, z. B. bei einem Unterprogrammaufruf, sind alle Kombinationen von Prefix, Programmnamen und Postfix möglich.

Beispiel:

Das Unterprogramm mit dem Programmnamen `SUB_PROG` kann über folgende Bezeichner aufgerufen werden:

1. `SUB_PROG`
2. `_N_SUB_PROG`
3. `SUB_PROG_SPF`
4. `_N_SUB_PROG_SPF`

### Namensgleichheit bei Haupt- und Unterprogrammen

Existiert ein Hauptprogramme (.MPF) und ein Unterprogramme (.SPF) mit gleichem Programmnamen, muss bei der Verwendung des Programmnamens im NC-Programm die entsprechende Dateierweiterung zur eindeutigen Kennzeichnung mit angegeben werden. Ansonsten wird das Programm verwendet, das entlang des Suchpfades als erstes mit dem angegebenen Programmnamen gefunden wird.

## 3.2.1.3 Schachtelung von Unterprogrammen

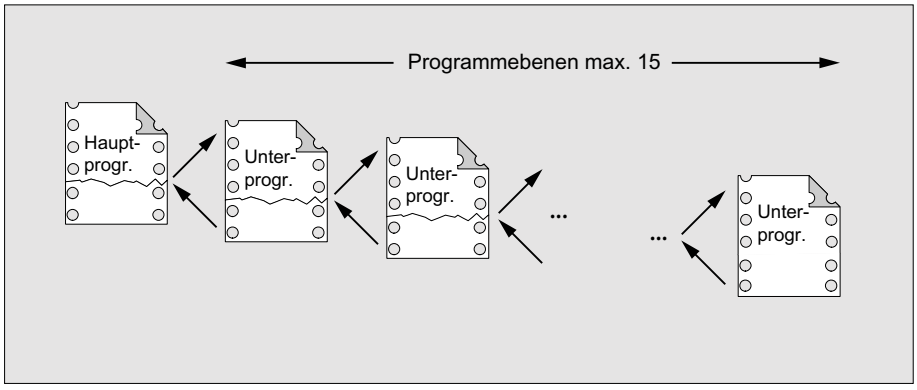
Ein Hauptprogramm kann Unterprogramme aufrufen, die wiederum Unterprogramme aufrufen. Die Abläufe der Programme sind somit ineinander geschachtelt. Jedes Programm läuft dabei in einer eigenen Programmebene.

### Schachtelungstiefe

Die NC-Sprache stellt aktuell 16 Programmebenen zur Verfügung. Das Hauptprogramm läuft immer in der obersten Programmebene 0. Ein Unterprogramm läuft immer in der dem Aufruf folgenden nächstniedrigeren Programmebene. Die Programmebene 1 ist somit die erste Unterprogrammebene.

Unterteilung der Programmebenen:

- Programmebene 0: Hauptprogrammebene
- Programmebene 1 - 15: Unterprogrammebene 1 - 15




### Interruptroutinen (ASUP)

Wird im Rahmen einer Interruptroutine ein Unterprogramm aufgerufen, wird dieses nicht in der aktuellen im Kanal aktiven Programmebene (n), sondern ebenfalls in der nächstniedrigeren Programmebene (n+1) abgearbeitet. Damit dies auch in der untersten Programmebene noch möglich ist, stehen im Zusammenhang mit Interruptroutinen 2 zusätzliche Programmebenen (16 und 17) zur Verfügung.

Werden mehr als 2 Programmebenen benötigt, muss dies explizit in der Strukturierung des im Kanal abgearbeiteten Teileprogramms berücksichtigt werden. D. h. es darf dann maximal nur so viele Programmebenen beanspruchen, dass noch ausreichend Programmebenen für die Interruptbearbeitung zur Verfügung stehen.

Benötigt die Interruptbearbeitung z. B. 4 Programmebenen, muss das Teileprogramm so strukturiert werden, dass es maximal Programmebene 13 belegt. Erfolgt dann ein Interrupt, stehen diesem die benötigten 4 Programmebenen (14 bis 17) zur Verfügung.

### Siemens-Zyklen

Siemens-Zyklen benötigen 3 Programmebenen. Der Aufruf eines Siemens-Zyklus muss daher spätestens erfolgen in:

- Teileprogrammbearbeitung: Programmebene 12
- Interruptroutine: Programmebene 14

## 3.2.1.4 Suchpfad

Beim Aufruf eines Unterprogramms ohne Pfadangabe durchsucht die Steuerung nach einer vordefinierten Suchreihenfolge (siehe "Suchpfad bei Unterprogrammaufruf (Seite 568)") die vorhandenen Programmspeicher.

### 3.2.1.5 Formal- und Aktualparameter

Von Formal- und Aktualparameter spricht man im Zusammenhang mit der Definition und dem Aufruf von Unterprogrammen mit Parameterübergabe.

#### Formalparameter

Bei der Definition eines Unterprogramms müssen die dem Unterprogramm zu übergebenden Parameter, die sogenannten Formalparameter, mit Typ und Parameternamen definiert werden.

Die Formalparameter definieren somit die Schnittstelle des Unterprogramms.

Beispiel:

| **Programmcode** | **Kommentar** |
|---|---|
| `PROC KONTUR (REAL X, REAL Y)` | `; Formalparameter: X und Y beide vom Typ REAL` |
| `N20 X1=X Y1=Y` | `; Verfahren der Achse X1 auf Position X und der Achse Y1 auf Position Y` |
| `...` | |
| `N100 RET` | |

#### Aktualparameter

Beim Aufruf eines Unterprogramms müssen dem Unterprogramm absolute Werte oder Variablen, die sogenannten Aktualparameter, übergeben werden.

Die Aktualparameter befüllen somit beim Aufruf die Schnittstelle des Unterprogramms mit aktuellen Werten.

Beispiel:

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 DEF REAL BREITE` | `; Variablendefinition` |
| `N20 BREITE=20.0` | `; Variablenzuweisung` |
| `N30 KONTUR(5.5, BREITE)` | `; Unterprogrammaufruf mit Aktualparametern: 5.5 und BREITE` |
| `...` | |
| `N100 M30` | |

### 3.2.1.6 Parameterübergabe

#### Definition eines Unterprogramms mit Parameterübergabe

Die Definition eines Unterprogramms mit Parameterübergabe erfolgt mit dem Schlüsselwort `PROC` und einer vollständigen Auflistung aller vom Unterprogramm erwarteten Parameter.

## Unvollständige Parameterübergabe

Beim Aufruf des Unterprogramms müssen nicht immer alle in der Unterprogrammschnittstelle definierten Parameter explizit übergeben werden. Wird ein Parameter weggelassen, wird für diesen Parameter der Standardwert "0" übergeben.

Zur eindeutigen Kennzeichnung der Reihenfolge der Parameter müssen allerdings die Kommas als Trennzeichen der Parameter immer mit angegeben werden. Eine Ausnahme bildet der letzte Parameter. Wird dieser beim Aufruf weggelassen, kann auch das letzte Komma entfallen.

**Beispiel:**

Unterprogramm:

| Programmcode | Kommentar |
|---|---|
| `PROC SUB_PROG (REAL X, REAL Y, REAL Z)` | `; Formalparameter: X, Y und Z` |
| `...` | |
| `N100 RET` | |

Hauptprogramm:

| Programmcode | Kommentar |
|---|---|
| `PROC MAIN_PROG` | |
| `...` | |
| `N30 SUB_PROG(1.0,2.0,3.0)` | `; Unterprogrammaufruf mit vollständiger Parameterübergabe:` `X=1.0, Y=2.0, Z=3.0` |
| … | |
| `N100 M30` | |

Beispiele für den Unterprogrammaufruf in `N30` mit unvollständiger Parameterübergabe:

```
N30 SUB_PROG( ,2.0,3.0)          ; X=0.0, Y=2.0, Z=3.0
N30 SUB_PROG(1.0, ,3.0)          ; X=1.0, Y=0.0, Z=3.0
N30 SUB_PROG(1.0,2.0)            ; X=1.0, Y=2.0, Z=0.0
N30 SUB_PROG( , ,3.0)            ; X=0.0, Y=0.0, Z=3.0
N30 SUB_PROG( , , )              ; X=0.0, Y=0.0, Z=0.0
```

**ACHTUNG**

**Parameterübergabe Call-by-Reference**

Parameter, die über Call-by-Reference übergeben werden, dürfen beim Unterprogramm-Aufruf nicht weggelassen werden.

**ACHTUNG**

**Datentyp AXIS**

Parameter vom Datentyp AXIS dürfen beim Unterprogramm-Aufruf nicht weggelassen werden.

### Überprüfung der Übergabeparameter

Über die Systemvariable $P_SUBPAR [ n ] mit n = 1, 2, ... kann im Unterprogramm überprüft werden, ob ein Parameter explizit übergeben oder weggelassen wurde. Der Index n bezieht sich auf die Reihenfolge der Formalparameter. Index n = 1 bezieht sich auf den 1. Formalparameter, Index n = 2 auf den 2. Formalparameter usw.

Der folgende Programmausschnitt zeigt beispielhaft für den 1. Formalparameter, wie eine Überprüfung realisiert werden kann:

| **Programmierung** | **Kommentar** |
|---|---|
| `PROC SUB_PROG (REAL X, REAL Y, REAL Z)` | `; Formalparameter: X, Y und Z` |
| `N20 IF $P_SUBPAR[1]==TRUE` | `; Überprüfung des 1.Formalparameters X.` |
| `...` | `; Diese Aktionen werden ausgeführt, wenn der Formalparameter X explizit übergeben wurde.` |
| `N40 ELSE` | |
| `...` | `; Diese Aktionen werden ausgeführt, wenn der Formalparameter X nicht übergeben wurde.` |
| `N60 ENDIF` | |
| `...` | `; Allgemeine Aktionen` |
| `N100 RET` | |

## 3.2.2 Definition eines Unterprogramms

### 3.2.2.1 Unterprogramm ohne Parameterübergabe

Bei der Definition von Unterprogrammen ohne Parameterübergabe kann die Definitionszeile am Programmanfang entfallen.

### Syntax

```
[PROC <Programmname>]
...
```

### Bedeutung

| | |
|---|---|
| `PROC`: | Definitionsanweisung am Anfang eines Programms |
| `<Programmname>`: | Name des Programms |

## Beispiel

Beispiel 1: Unterprogramm mit PROC-Anweisung

| Programmcode | Kommentar |
|---|---|
| PROC SUB_PROG | ; Definitionszeile |
| N10 G01 G90 G64 F1000 | |
| N20 X10 Y20 | |
| ... | |
| N100 RET | ; Unterprogrammrücksprung |

Beispiel 2: Unterprogramm ohne PROC-Anweisung

| Programmcode | Kommentar |
|---|---|
| N10 G01 G90 G64 F1000 | |
| N20 X10 Y20 | |
| ... | |
| N100 RET | ; Unterprogrammrücksprung |

## Siehe auch

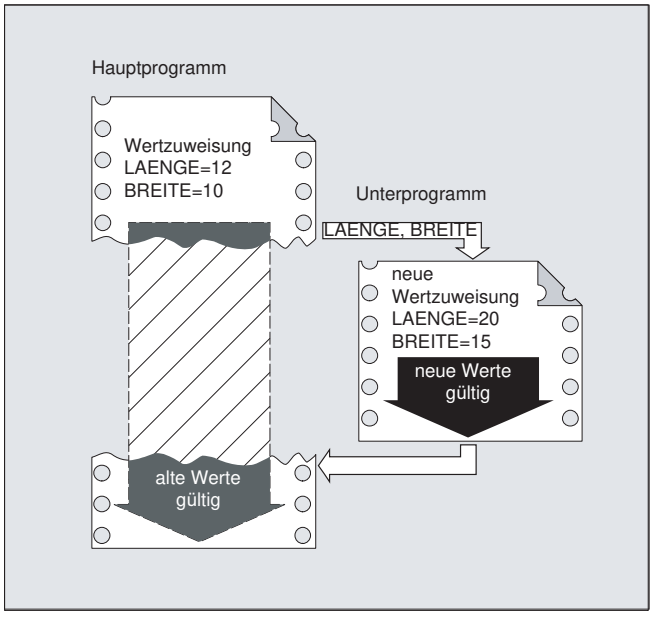
Unterprogrammaufruf ohne Parameterübergabe (Seite 530)

### 3.2.2.2 Unterprogramm mit Parameterübergabe Call-by-Value (PROC)

Die Definition eines Unterprogramms mit Parameterübergabe Call-by-Value erfolgt mit dem Schlüsselwort PROC, gefolgt vom Programmnamen und einer vollständigen Auflistung aller Parameter mit Typ und Namen. Die Definitionsanweisung muss in der ersten Programmzeile stehen.

#### Call-by-Value

Das aufrufende Programm übergibt bei einer Parameterübergabe Call-by-Value dem Unterprogramm nur den Wert einer Variablen Damit bekommt das Unterprogramm keinen direkten Zugriff auf die Variable. Dadurch wird bei einer Änderung des Parameterwerts nur der im Unterprogramm sichtbare Wert geändert. Der Wert der im aufrufenden Programm definierten Variablen bleibt unverändert. Die Parameterübergabe Call-by-Value hat somit keine Rückwirkungen auf das aufrufende Programm.




## Syntax

```
PROC <Programmname> (<Parametertyp> <Parametername>=<Init_Wert>, ...)
```

### Hinweis

Es können bis zu 127 Parameter übergeben werden.

## Bedeutung

| | |
|---|---|
| `PROC`: | Definitionsanweisung am Anfang eines Programms |
| `<Programmname>`: | Name des Programms |
| `<Parametertyp>`: | Datentyp des Parameters (z. B. REAL, INT, BOOL) |
| `<Parametername>`: | Name des Parameters |
| `<Init_Wert>`: | Optionaler Wert zur Initialisierung des Parameters (**optional**)<br>Wird beim Aufruf des Unterprogramms kein Parameter angegeben, erhält der Parameter den Initialisierungswert zugewiesen. |

## Beispiele

### Beispiel 1

Definition eines Unterprogramms SUB_PROG mit drei Parametern vom Typ REAL mit Defaultwerten:

**Programmcode**

```
PROC SUB_PROG(REAL LENGTH=10.0, REAL WIDTH=20.0, REAL HIGHT=30.0)
```

**Beispiel 2**

Verschiedene Aufrufvarianten

**Programmcode**

```
PROC MAIN_PROG
  REAL PAR_1 = 100
  REAL PAR_2 = 200
  REAL PAR_3 = 300
  ; Aufrufvarianten
  SUB_PROG
  SUB_PROG(PAR_1, PAR_2, PAR_3)
  SUB_PROG(PAR_1)
  SUB_PROG(PAR_1, , PAR_3)
  SUB_PROG( , , PAR_3)
N100 RET
```

## Siehe auch

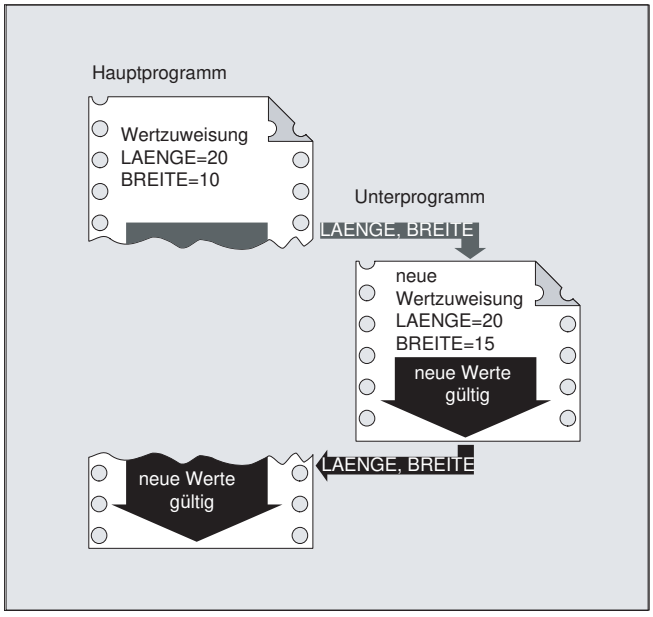
Unterprogrammaufruf mit Parameterübergabe (EXTERN) (Seite 532)

### 3.2.2.3 Unterprogramm mit Parameterübergabe Call-by-Reference (PROC, VAR)

Die Definition eines Unterprogramms mit Parameterübergabe Call-by-Reference erfolgt mit dem Schlüsselwort `PROC`, gefolgt vom Programmnamen und einer vollständigen Auflistung aller Parameter mit Schlüsselwort `VAR`, Typ und Namen. Die Definitionsanweisung muss in der ersten Programmzeile stehen. Als Parameter können auch Referenzen auf Felder übergeben werden.

**Call-by-Reference**

Das aufrufende Programm übergibt bei einer Parameterübergabe Call-by-Reference dem Unterprogramm nicht den Wert einer Variablen, sondern eine Referenz (Zeiger) auf die Variable. Damit bekommt das Unterprogramm direkten Zugriff auf die Variable. Dadurch wird bei einer Änderung des Parameterwertes nicht nur der im Unterprogramm sichtbare Wert geändert, sondern der Wert der im aufrufenden Programm definierten Variablen. Die Parameterübergabe Call-by-Reference hat somit auch nach Beendigung des Unterprogramms Rückwirkungen auf das aufrufende Programm.




**Hinweis**

Die Parameterübergabe Call-by-Reference ist nur dann erforderlich, wenn die übergebene Variable lokal in einem aufrufenden Programm definiert ist (LUD). Kanal-globale oder NC-globale Variablen müssen nicht übergeben werden, da auf diese auch direkt vom Unterprogramm aus zugegriffen werden kann.

## Syntax

```
PROC <Programmname> (VAR <Parametertyp> <Parametername>, ...)
PROC <Programmname> (VAR <Feldtyp> <Feldname> [<m>,<n>,<o>], ...)
```

**Hinweis**

Es können bis zu 127 Parameter übergeben werden.

## Bedeutung

| | |
|---|---|
| PROC: | Definitionsanweisung am Anfang eines Programms |
| VAR: | Schlüsselwort für die Parameterübergabe per Referenz |
| <Programmname>: | Name des Programms |
| <Parametertyp>: | Datentyp des Parameters (z. B. REAL, INT, BOOL) |
| <Parametername>: | Name des Parameters |
| <Feldtyp>: | Datentyp der Feldelemente (z. B. REAL, INT, BOOL) |
| <Feldname>: | Name des Feldes |

| | | |
|---|---|---|
| [<m>,<n>,<o>]: | Feldgröße<br>Aktuell sind maximal 3-dimensionale Felder möglich: | |
| | <m>: | Feldgröße für 1. Dimension |
| | <n>: | Feldgröße für 2. Dimension |
| | <o>: | Feldgröße für 3. Dimension |

**Hinweis**

- Der nach dem Schlüsselwort `PROC` angegebene Programmname muss mit dem an der Bedienoberfläche vergebenen Programmnamen übereinstimmen.
- Mit Feldern unbestimmter Feldlänge als Formalparameter können Unterprogramme Felder variabler Länge bearbeiten. Dazu wird bei der Definition z. B. eines zweidimensionalen Feldes als Formalparameter die Länge der 1. Dimension nicht angegeben. Das Komma aber muss geschrieben werden.
  Beispiel: `PROC <Programmname> (VAR REAL FELD[ ,5])`

## Beispiel

Definition eines Unterprogramms mit zwei Parameter als Referenz auf Typ REAL:

**Programmcode**

```
; Parameter 1: Referenz auf Typ: REAL, Name: LAENGE
; Parameter 2: Referenz auf Typ: REAL, Name: BREITE
PROC SUB_PROG(VAR REAL LAENGE, VAR REAL BREITE)
```

## Siehe auch

Unterprogrammaufruf mit Parameterübergabe (EXTERN) (Seite 532)

### 3.2.2.4 Modale G-Funktionen sichern (SAVE)

Das Attribut SAVE bewirkt, dass die vor dem Unterprogrammaufruf aktiven modalen G-Befehle gesichert und nach dem Unterprogrammende wieder reaktiviert werden.

| ACHTUNG |
|---|
| **Unterbrechung des Bahnsteuerbetriebs**<br>Wird bei aktivem Bahnsteuerbetrieb ein Unterprogramme mit Attribut SAVE aufgerufen, wird der Bahnsteuerbetrieb am Ende des Unterprogramms (Rücksprung) unterbrochen. |

## Syntax

```
PROC <Unterprogrammname> SAVE
```

## Bedeutung

| | |
|---|---|
| SAVE: | Sichern der modalen G-Befehle vor dem Unterprogrammaufruf und Wiederherstellen nach Unterprogrammende |

## Beispiel

Im Unterprogramm KONTUR wirkt der modale G-Befehl G91 (Kettenmaß). Im Hauptprogramm wirkt der modale G-Befehl G90 (Absolutmaß). Durch die Unterprogrammdefinition mit SAVE wirkt nach dem Unterprogrammende im Hauptprogramm wieder G90.

### Unterprogramm-Definition:

| Programmcode | Kommentar |
|---|---|
| PROC KONTUR (REAL WERT1) SAVE | ; Unterprogramm-Definition mit Parameter SAVE |
| N10 G91 ... | ; Modaler G-Befehl G91: Kettenmaß |
| N100 M17 | ; Unterprogrammende |

### Hauptprogramm:

| Programmcode | Kommentar |
|---|---|
| N10 G0 X... Y... G90 | ; Modaler G-Befehl G90: Absolutmaß |
| N20 ... | |
| ... | |
| N50 KONTUR (12.4) | ; Unterprogrammaufruf |
| N60 X... Y... | ; Modaler G-Befehl G90 durch SAVE reaktiviert |

## Randbedingungen

### Frames

Das Verhalten von Frames bezüglich Unterprogrammen mit dem Attribut SAVE ist abhängig vom Typ des Frames und kann über Maschinendaten eingestellt werden.

## 3.2.2.5 Einzelsatzbearbeitung unterdrücken (SBLOF, SBLON)

Auch bei aktiver Einzelsatzbearbeitung hat der Anwender die Möglichkeit, ein NC-Programm komplett oder teilweise ohne anzuhalten abzuarbeiten. Die Unterdrückung der Einzelsatzbearbeitung erfolgt über den Befehl SBLOF, das Wiedereinschalten über den Befehl SBLON.

### Einzelsatzbearbeitung für das gesamte NC-Programm unterdrücken

Ist das Ausschalten der Einzelsatzbearbeitung (SBLOF) in der ersten Zeile (PROC ...) eines **Hauptprogramms** programmiert, gilt dies bis zum Ende oder Abbruch des NC-Programms. Das NC-Programm wird dann bei Einzelsatz ohne anzuhalten abgearbeitet.

Ist das Ausschalten der Einzelsatzbearbeitung (SBLOF) in der ersten Zeile (PROC ...) eines **Unterprogramms** programmiert, gilt dies bis zum Ende oder Abbruch des NC-Programms. Mit

dem programmierten Rücksprung-Befehl wird entschieden, ob am Ende des Unterprogramms angehalten wird oder nicht:

- Rücksprung mit M17: Stopp am Ende des Unterprogramms
- Rücksprung mit RET: **Kein** Stopp am Ende des Unterprogramms

**Einzelsatzbearbeitung innerhalb des NC-Programms unterdrücken**

Ist das Ausschalten der Einzelsatzbearbeitung (SBLOF) in einem Satz innerhalb eines NC-Programms programmiert, wird die Einzelsatzbearbeitung ab diesem Satz bis zum nächsten programmierten Einschalten der Einzelsatzbearbeitung (SBLON) oder dem Ende der aktiven Unterprogrammebene ausgeschaltet.

## Syntax

**Einzelsatzbearbeitung für das gesamte NC-Programm unterdrücken:**

```
PROC ... SBLOF
...
```

**Einzelsatzbearbeitung innerhalb des NC-Programms unterdrücken:**

```
...
SBLOF
...
SBLON
...
```

## Bedeutung

| | | |
|---|---|---|
| PROC: | Erste Anweisung eines Programms | |
| SBLOF: | Vordefinierte Prozedur zum Ausschalten der Einzelsatzbearbeitung | |
| | Alleine im Satz: | ja, im PROC-Satz möglich |
| | Wirksamkeit: | modal |
| SBLON: | Vordefinierte Prozedur zum Einschalten der Einzelsatzbearbeitung | |
| | Alleine im Satz: | ja |
| | Wirksamkeit: | modal |

## Besonderheiten

- **Satzanzeige bei unterdrückter Einzelsatzbearbeitung**
  Die aktuelle Satzanzeige kann in Unterprogrammen mit DISPLOF unterdrückt werden. Wird DISPLOF zusammen mit SBLOF programmiert, wird bei Einzelsatz-Stopps innerhalb des Unterprogramms der Aufruf des Unterprogramms angezeigt.
- **Unterdrückung der Einzelsatzbearbeitung bei asynchronen Unterprogrammen (ASUP)**
  Um ein ASUP bei aktiver Einzelsatzbearbeitung in einem Schritt abzuarbeiten, muss im ASUP eine PROC-Anweisung mit SBLOF programmiert werden. Dies gilt auch für die Funktion "Editierbares System-ASUP" (MD11610 $MN_ASUP_EDITABLE).
  Wenn der Einzelsatz-Stopp im System- oder Anwender-ASUP über die Programmierung von SBLOF in der PROC-Zeile oder über die Einstellungen im Maschinendatum MD10702 $MN_IGNORE_SINGLEBLOCK_MASK unterdrückt wird (Bit0 = 1 bzw. Bit1 = 1), dann kann der Einzelsatz-Stopp durch Programmierung von SBLON im ASUP wieder aktiviert werden.
  Wird der Einzelsatz-Stopp im Anwender-ASUP über die Einstellung im Maschinendatum MD20117 $MC_IGNORE_SINGLEBLOCK_ASUP unterdrückt, dann kann der Einzelsatz-Stopp durch Programmierung von SBLON im ASUP **nicht** wieder aktiviert werden.
- **Besonderheiten bei den verschiedenen Einzelsatzbearbeitungstypen**
  - "SB2: Rechensatz" UND MD10702 $MN_IGNORE_SINGLEBLOCK_MASK, Bit 12 = 1:
    → Im SBLON-Satz wird nicht angehalten.
  - "SB3: Einzelsatz fein":
    → Der Befehl SBLOF wird unterdrückt.
- **Unterdrückung der Einzelsatzbearbeitung in geschachtelten Programmen**
  Wurde in einem Unterprogramm SBLOF in der PROC-Anweisung programmiert, so wird auf den Unterprogrammrücksprung mit M17 angehalten. Damit wird verhindert, dass im aufrufenden Programm bereits der nächste Satz ausgeführt wird. Wird in einem Unterprogramm mit SBLOF, ohne SBLOF in der PROC-Anweisung, eine Unterdrückung der Einzelsatzbearbeitung aktiviert, wird erst nach dem nächsten Maschinenfunktionssatz des aufrufenden Programms angehalten. Ist dies nicht erwünscht, muss im Unterprogramm noch vor dem Rücksprung (M17) wieder SBLON programmiert werden. Bei einem Rücksprung mit RET in ein übergeordnetes Programm wird nicht angehalten.

## Beispiele

### Beispiel 1: Einzelsatzbearbeitung innerhalb des NC-Programms unterdrücken

Ausgangssituation: Einzelsatzbearbeitung ist aktiv.

```
Programmcode                    Kommentar
N10 G1 X100 F1000
N20 SBLOF                       ; Einzelsatzbearbeitung ausschalten
N30 Y20
N40 M100
N50 R10=90
N60 SBLON                       ; Einzelsatzbearbeitung wieder einschalten
N70 M110
N80 ...
```

Der Bereich zwischen N20 und N60 wird im Einzelsatzbetrieb als ein Schritt bearbeitet.

**Beispiel 2: Zyklus soll für den Anwender wie ein Befehl wirken**

Ausgangssituation: Einzelsatzbearbeitung ist aktiv.

Hauptprogramm:

```
Programmcode
...
N100 G1 X10 G90 F200
N120 X-4 Y6
N130 CYCLE1
N140 G1 X0
N150 M30
```

Zyklus CYCLE1:

```
Programmcode                              Kommentar
N100 PROC CYCLE1 DISPLOF SBLOF            ; Einzelsatzbearbeitung für das gesamte Pro-
                                          gramm unterdrücken.
N110 R10=3*SIN(R20)+5
N120 IF (R11 <= 0)
N130  SETAL(61000)
N140 ENDIF
N150 G1 G91 Z=R10 F=R11
N160 M17
```

Der Zyklus CYCLE1 wird bei aktiver Einzelsatzbearbeitung abgearbeitet. D. h., es muss für die Bearbeitung von CYCLE1 einmal die Start-Taste gedrückt werden.

**Beispiel 3: Ein von der PLC gestartetes ASUP zum Aktivieren von geänderter Nullpunktverschiebung und Werkzeugkorrekturen soll nicht sichtbar sein**

```
Programmcode                              Kommentar
N100 PROC NV SBLOF DISPLOF                ; Einzelsatzbearbeitung und Satzanzeige unter-
                                          drücken.
N110 CASE $P_UIFRNUM OF
       0 GOTOF _G500
       1 GOTOF _G54
       2 GOTOF _G55
       3 GOTOF _G56
       4 GOTOF _G57
     DEFAULT GOTOF END
N120 _G54: G54 D=$P_TOOL T=$P_TOOLNO
N130 RET
N140 _G54: G55 D=$P_TOOL T=$P_TOOLNO
N150 RET
N160 _G56: G56 D=$P_TOOL T=$P_TOOLNO
N170 RET
```

| Programmcode | Kommentar |
|---|---|
| N180 _G57: G57 D=$P_TOOL T=$P_TOOLNO | |
| N190 RET | |
| N200 END: D=$P_TOOL T=$P_TOOLNO | |
| N210 RET | |

### Beispiel 4: Gezieltes Anhalten im Unterprogramm

Ausgangssituation:

- Einzelsatzbearbeitung ist aktiv.
- MD10702 $MN_IGNORE_SINGLEBLOCK_MASK, Bit12 = 1

Hauptprogramm:

| Programmcode | Kommentar |
|---|---|
| N10 G0 X0 | ; Einzelsatz-Stopp |
| N20 X10 | ; Einzelsatz-Stopp |
| N30 **CYCLE** | ; Vom Zyklus generierter Verfahrsatz. |
| N50 G90 X20 | ; Einzelsatz-Stopp |
| M30 | |

Zyklus CYCLE:

| Programmcode | Kommentar |
|---|---|
| PROC CYCLE **SBLOF** | ; Einzelsatzbearbeitung unterdrücken |
| N100 R0=1 | |
| N110 **SBLON** | ; Kein Einzelsatz-Stopp wegen MD10702 Bit12=1 |
| N120 X1 | ; Einzelsatz-Stopp |
| N140 **SBLOF** | |
| N150 R0=2 | |
| RET | |

### Beispiel 5: Unterdrückung der Einzelsatzbearbeitung bei Programmschachtelung

Ausgangssituation:

- Einzelsatzbearbeitung Typ 2 ist aktiv.
- Im SBLON-Satz soll nicht angehalten werden (MD10702 $MN_IGNORE_SINGLEBLOCK_MASK, Bit12 = 1)

| Programmcode | Kommentar |
|---|---|
| N10 X0 F1000 | ; Einzelsatz-Stopp |
| N20 UP1(0) | |
| **PROC** UP1(INT _NR) **SBLOF** | ; Einzelsatzbearbeitung für UP1 ausschalten |
| N100 X10 | |
| N110 UP2(0) | |
| PROC UP2(INT _NR) | |
| N200 X20 | |

```
Programmcode                                    Kommentar
            N210 SBLON                          ; Einzelsatzbearbeitung einschalten
            N220 X22                            ; Einzelsatz-Stopp
            N230 UP3(0)
                  PROC UP3(INT _NR)
                  N300 SBLOF                    ; Einzelsatzbearbeitung ausschalten
                  N305 X30
                  N310 SBLON                    ; Einzelsatzbearbeitung einschalten
                  N320 X32                      ; Einzelsatz-Stopp
                  N330 SBLOF                    ; Einzelsatzbearbeitung ausschalten
                  N340 X34
                  N350 M17                      ; Einzelsatz-Stopp (M17)
            N240 X24                            ; Einzelsatz-Stopp (N210)
            N250 M17                            ; Einzelsatz-Stopp (M17)
      N120 X12
      N130 M17                                  ; Einzelsatz-Stopp (M17)
N30 X0                                          ; Einzelsatz-Stopp
N40 M30                                         ; Einzelsatz-Stopp
```

### 3.2.2.6 Aktuelle Satzanzeige unterdrücken (DISPLOF, DISPLON, ACTBLOCNO)

In der Satzanzeige wird standardmäßig der aktuelle Programmsatz angezeigt. In Zyklen bzw. Unterprogrammen kann die Anzeige des aktuellen Satzes mit dem Befehl DISPLOF unterdrückt werden. Anstelle des aktuellen Satzes wird dann der Aufruf des Zyklus bzw. Unterprogramms angezeigt. Mit dem Befehl DISPLON kann die Unterdrückung der Satzanzeige wieder aufgehoben werden.

DISPLOF bzw. DISPLON wird in der Programmzeile mit der PROC-Anweisung programmiert und wirkt für das gesamte Unterprogramm und implizit für alle von diesem Unterprogramm aufgerufenen Unterprogramme, die keinen DISPLON- bzw. DISPLOF-Befehl enthalten. Dieses Verhalten gilt auch für ASUPs.

#### Syntax

```
PROC … DISPLOF
PROC … DISPLOF ACTBLOCNO
PROC … DISPLON
```

#### Bedeutung

| | | |
|---|---|---|
| DISPLOF: | Befehl zum Unterdrücken der aktuellen Satzanzeige. | |
| | Platzierung: | Am Ende der Programmzeile mit der PROC-Anweisung |
| | Wirksamkeit: | Bis zum Rücksprung aus dem Unterprogramm oder Programmende. |
| | **Hinweis:**<br>Wenn aus dem Unterprogramm mit dem DISPLOF-Befehl weitere Unterprogramme aufgerufen werden, dann wird auch in diesen Unterprogrammen die aktuelle Satzanzeige unterdrückt, sofern in diesen nicht explizit DISPLON programmiert ist. | |

| | | |
|---|---|---|
| DISPLON: | Befehl zum Aufheben der Unterdrückung der aktuellen Satzanzeige | |
| | Platzierung: | Am Ende der Programmzeile mit der PROC-Anweisung |
| | Wirksamkeit: | Bis zum Rücksprung aus dem Unterprogramm oder Programmende. |
| | **Hinweis:**<br>Wenn aus dem Unterprogramm mit dem DISPLON-Befehl weitere Unterprogramme aufgerufen werden, dann wird auch in diesen Unterprogrammen der aktuelle Programmsatz angezeigt, sofern in diesen nicht explizit DISPLOF programmiert ist. | |
| ACTBLOCNO: | DISPLOF zusammen mit dem Attribut ACTBLOCNO bewirkt, dass im Falle eines Alarms die Nummer des aktuellen Satzes ausgegeben wird, in dem der Alarm aufgetreten ist. Dies gilt auch dann, wenn in einer niedrigeren Programmebene nur DISPLOF programmiert ist.<br>Bei DISPLOF ohne ACTBLOCNO wird dagegen die Satznummer des Zyklen- bzw. Unterprogrammaufrufs aus der letzten nicht mit DISPLOF gekennzeichneten Programmebene angezeigt. | |

## Beispiele

### Beispiel 1: Aktuelle Satzanzeige im Zyklus unterdrücken

```
Programmcode                                Kommentar
PROC CYCLE(AXIS TOMOV, REAL POSITION)       ; Aktuelle Satzanzeige unterdrücken. Statt-
SAVE DISPLOF                                dessen soll der Zyklus-Aufruf angezeigt wer-
                                            den, z. B.: CYCLE(X,100.0)
DEF REAL DIFF                               ; Zyklen-Inhalt
G01 ...
...
RET                                         ; Unterprogramm-Rücksprung. In der Satzan-
                                            zeige wird der auf den Zyklus-Aufruf folgen-
                                            de Satz angezeigt.
```

### Beispiel 2: Satzanzeige bei der Alarmausgabe

Unterprogramm SUBPROG1 (mit ACTBLOCNO):

```
Programmcode                    Kommentar
PROC SUBPROG1 DISPLOF ACTBLOC-
NO
N8000 R10 = R33 + R44
...
N9040 R10 = 66 X100             ; Alarm 12080 auslösen
...
N10000 M17
```

Unterprogramm SUBPROG2 (ohne ACTBLOCNO):

| **Programmcode** | **Kommentar** |
|---|---|
| PROC SUBPROG2 DISPLOF | |
| N5000 R10 = R33 + R44 | |
| ... | |
| N6040 R10 = 66 X100 | ; Alarm 12080 auslösen |
| ... | |
| N7000 M17 | |

Hauptprogramm:

| **Programmcode** | **Kommentar** |
|---|---|
| N1000 G0 X0 Y0 Z0 | |
| N1010 ... | |
| ... | |
| N2050 SUBPROG1 | ; Alarmausgabe = "12080 Kanal K1 Satz N9040 Syntax-fehler bei Text R10=" |
| N2060 ... | |
| N2350 SUBPROG2 | ; Alarmausgabe = "12080 Kanal K1 Satz N2350 Syntax-fehler bei Text R10=" |
| ... | |
| N3000 M30 | |

### Beispiel 3: Unterdrückung der aktuellen Satzanzeige aufheben

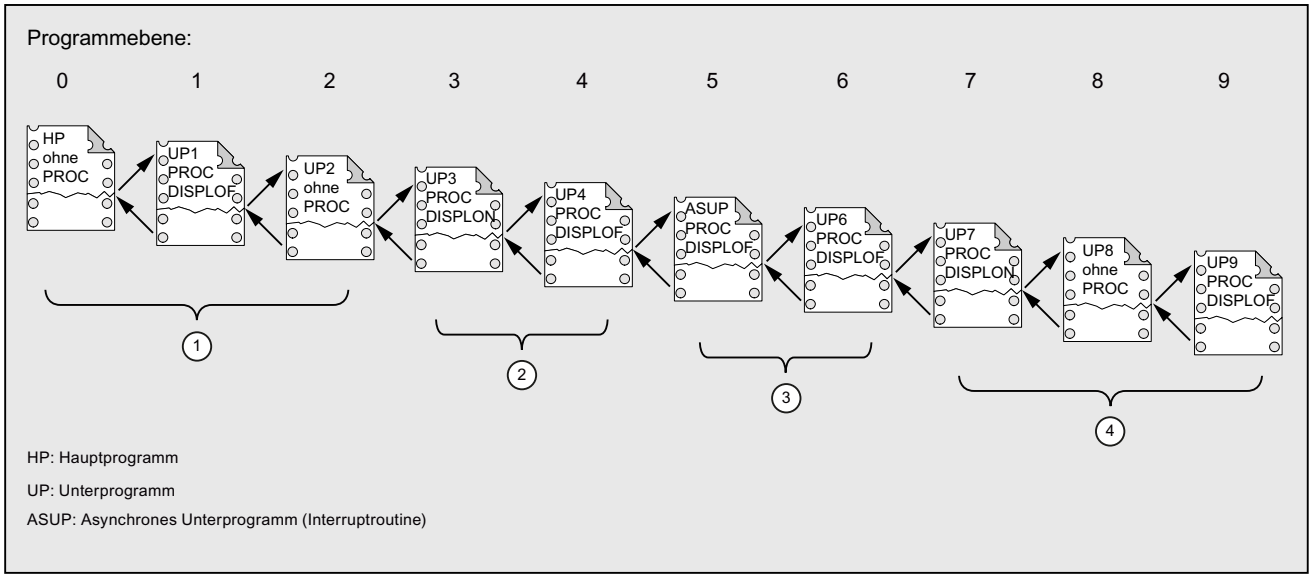
Unterprogramm SUB1 mit Unterdrückung:

| **Programmcode** | **Kommentar** |
|---|---|
| PROC SUB1 DISPLOF | ; Aktuelle Satzanzeige im Unterprogramm SUB1 unterdrücken. Stattdessen soll der Satz mit dem SUB1-Aufruf angezeigt werden. |
| ... | |
| N300 SUB2 | ; Unterprogramm SUB2 aufrufen. |
| ... | |
| N500 M17 | |

Unterprogramm SUB2 ohne Unterdrückung:

| **Programmcode** | **Kommentar** |
|---|---|
| PROC SUB2 DISPLON | ; Unterdrückung der aktuellen Satzanzeige im Unterprogramm SUB2 aufheben. |
| ... | |
| N200 M17 | ; Rücksprung ins Unterprogramm SUB1. In SUB1 wird die aktuelle Satzanzeige wieder unterdrückt. |

**Beispiel 4: Anzeigeverhalten bei unterschiedlichen DISPLON/DISPLOF-Kombinationen**




① In der aktuellen Satzanzeige werden die Teileprogrammzeilen aus Programmebene 0 angezeigt.
② In der aktuellen Satzanzeige werden die Teileprogrammzeilen aus Programmebene 3 angezeigt.
③ In der aktuellen Satzanzeige werden die Teileprogrammzeilen aus Programmebene 3 angezeigt.
④ In der aktuellen Satzanzeige werden die Teileprogrammzeilen aus Programmebene 7/8 angezeigt.

### 3.2.2.7 Unterprogramme mit Vorbereitung kennzeichnen (PREPRO)

Mit dem Schlüsselwort `PREPRO` können im Hochlauf am Ende der `PROC`-Anweisungszeile alle Dateien gekennzeichnet werden.

**Hinweis**

Diese Art der Programmvorbereitung ist von der Einstellung von Maschinendatum MD10700 $MN_PREPROCESSING_LEVEL abhängig. Bitte Angaben des Maschinenherstellers beachten.

## Syntax

```
PROC … PREPRO
```

## Bedeutung

| | |
|---|---|
| `PREPRO`: | Schlüsselwort für Kennzeichnung aller im Hochlauf vorbereiteten Dateien, der in Zyklenverzeichnissen abgelegten NC-Programme |

### Unterprogramme mit Vorbereitung einlesen und Unterprogrammaufruf

Sowohl im Hochlauf vorbereiteter Unterprogramme mit Parametern als auch beim Unterprogrammaufruf werden die Zyklenverzeichnissen in der gleichen Reihenfolge behandelt:

1. _N_CUS_DIR Anwenderzyklen
2. _N_CMA_DIR Herstellerzyklen
3. _N_CST_DIR Standardzyklen

Im Falle gleichnamiger NC-Programme mit unterschiedlicher Ausprägung wird die zuerst gefundene `PROC`-Anweisung aktiviert und die andere `PROC`-Anweisung wird ohne Alarmmeldung überlesen.

## 3.2.2.8 Unterprogrammrücksprung M17

Am Ende eines Unterprogramms steht der Rücksprung-Befehl `M17` (bzw. der Teileprogrammende-Befehl `M30`). Er bewirkt den Rücksprung in das aufrufende Programm auf den Teileprogrammsatz nach dem Unterprogrammaufruf.

**Hinweis**

`M17` und `M30` werden in der NC-Sprache gleichwertig behandelt.

### Syntax

```
PROC <Programmname>
...
M17/M30
```

### Randbedingungen

#### Auswirkung des Unterprogrammrücksprungs auf den Bahnsteuerbetrieb

Steht `M17` (bzw. `M30`) alleine im Teilprogrammsatz, wird dadurch ein im Kanal aktiver Bahnsteuerbetrieb unterbrochen.

Um zu vermeiden, dass der Bahnsteuerbetrieb unterbrochen wird, ist `M17` (bzw. `M30`) mit in den letzten Verfahrsatz zu schreiben. Zusätzlich muss folgendes Maschinendatum auf "0" gesetzt sein:

MD20800 $MC_SPF_END_TO_VDI = 0 (keine M30/M17-Ausgabe an die NC/PLC-Nahtstelle)

### Beispiel

1. Unterprogramm mit `M17` im eigenen Satz

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 G64 F2000 G91 X10 Y10` | |
| `N20 X10 Z10` | |

| Programmcode | Kommentar |
|---|---|
| N30 M17 | ; Rücksprung mit Unterbrechung des Bahnsteuerbetriebs. |

2. Unterprogramm mit M17 im letzten Verfahrsatz

| Programmcode | Kommentar |
|---|---|
| N10 G64 F2000 G91 X10 Y10 | |
| N20 X10 Z10 M17 | ; Rücksprung ohne Unterbrechung des Bahnsteuerbetriebs. |

### 3.2.2.9 Unterprogrammrücksprung RET

Als Ersatz für den Rücksprungsbefehl M17 kann im Unterprogramm auch der Befehl RET verwendet werden. RET muss in einem eigenen Teileprogrammsatz programmiert werden. Wie M17 bewirkt RET den Rücksprung in das aufrufende Programm auf den Teileprogrammsatz nach dem Unterprogrammaufruf.

**Hinweis**

Durch die Programmierung von Parametern kann das Rücksprungverhalten von RET geändert werden (siehe "Parametrierbarer Unterprogrammrücksprung (RET ...) (Seite 520)").

## Anwendung

Die RET-Anweisung ist dann zu benutzen, wenn ein G64-Bahnsteuerbetrieb (G641 ... G645) durch den Rücksprung nicht unterbrochen werden soll.

## Voraussetzung

Der Befehl RET kann nur in Unterprogrammen verwendet werden, die nicht mit dem Attribut SAVE definiert wurden.

## Syntax

```
PROC <Programmname>
...
RET
```

## Beispiel

Hauptprogramm:

| Programmcode | Kommentar |
|---|---|
| PROC MAIN_PROGRAM | ; Programmanfang |
| ... | |

| Programmcode | Kommentar |
|---|---|
| N50 SUB_PROG | ; Unterprogrammaufruf: SUB_PROG |
| N60 ... | |
| ... | |
| N100 M30 | ; Programmende |

Unterprogramm:

| Programmcode | Kommentar |
|---|---|
| PROC SUB_PROG | |
| ... | |
| N100 RET | ; Rücksprung erfolgt auf Satz N60 im Hauptprogramm. |

### 3.2.2.10 Parametrierbarer Unterprogrammrücksprung (RET ...)

Im Allgemeinen wird mit dem Befehl `RET` aus einem Unterprogramm in das aufrufende Programm zurückgesprungen. Die Bearbeitung wird dann mit der auf den Unterprogrammaufruf folgenden Programmzeile fortgesetzt. Soll die Programmbearbeitung an einer anderen Stelle fortgesetzt werden, stehen folgende Möglichkeiten zur Verfügung:

- Fortsetzung der Programmbearbeitung nach Aufruf der Abspanzyklen im ISO-Dialekt-Modus (nach der Konturbeschreibung).
- Rücksprung ins Hauptprogramm aus einer beliebigen Unterprogrammebene (auch nach ASUP) beim Fehlerhandling.
- Rücksprung über mehrere Programmebenen für spezielle Anwendungen in Compile-Zyklen und im ISO-Dialekt-Modus.

Dazu ist der Befehl `RET` mit weiteren Parametern zu programmieren.

#### Suchrichtung

Bei Angabe des Parameters `<Zielsatz>` erfolgt der Rücksprung zuerst auf den Satz nach dem Aufrufsatz. Anschließend wird das Ziel in Richtung **Ende** des Programms gesucht, in das zurückgesprungen wird. Ist die Suche nicht erfolgreich, wird als nächstes in Richtung Programmanfang gesucht.

## Syntax

```
RET("<Zielsatz>")
RET("<Zielsatz>",<Satz nach Zielsatz>)
RET("<Zielsatz>",<Satz nach Zielsatz>,<Anzahl der Rücksprungebenen>)
RET("<Zielsatz>", ,<Anzahl der Rücksprungebenen>)
RET("<Zielsatz>",<Satz nach Zielsatz>,<Anzahl der Rücksprungebenen>,
<Rücksprung auf Programmanfang>)
RET( , ,<Anzahl der Rücksprungebenen>,<Rücksprung auf Programmanfang
>)
```

## Bedeutung

| | | | |
|---|---|---|---|
| RET: | Unterprogrammende | | |
| <Zielsatz>: | Im Parameter wird als Sprungziel der Satz angegeben, an dem die Programmbearbeitung fortgesetzt werden soll.<br>Ist der Parameter <Anzahl der Rücksprungebenen> nicht programmiert, dann befindet sich das Sprungziel in dem Programm, aus dem das aktuelle Unterprogramm aufgerufen wurde.<br>Mögliche Angaben sind: | | |
| | <Satznummer> | Nummer des Zielsatzes.<br>Die Suche nach der Satznummer erfolgt im Programm, in das zurückgesprungen wird, zuerst in Richtung Programmende. | |
| | <Sprungmarke> | Sprungmarke, die im Programm, in das zurückgesprungen wird, vorhanden sein muss.<br>Die Suche nach der Sprungmarke erfolgt im Programm, in das zurückgesprungen wird, zuerst in Richtung Programmende. | |
| | <Zeichenkette> | Zeichenkette, die im Programm, in das zurückgesprungen wird, vorhanden sein muss (z. B. Programm- oder Variablenname).<br>Die Suche nach der Zeichenkette erfolgt im Programm, in das zurückgesprungen wird, zuerst in Richtung Programmende.<br>Für die Programmierung der Zeichenkette im Zielsatz gelten folgende Regeln:<br>• **Leerzeichen am Ende** (im Unterschied zur Sprungmarke, die durch einen ":" am Ende gekennzeichnet ist).<br>• **Vor der Zeichenkette** dürfen nur eine Satznummer und/oder eine Sprungmarke gesetzt sein, **keine Programmbefehle**. | |
| <Satz nach Zielsatz>: | Im Parameter wird angegeben, ob die Programmbearbeitung im unter Parameter <Zielsatz> angegebenen Satz oder im darauf folgenden Satz fortgeführt werden soll. | | |
| | Typ: | INT | |
| | Wert: | 0 | Der Rücksprung erfolgt auf den im Parameter <Zielsatz> angegebenen Satz. |
| | | > 0 | Der Rücksprung erfolgt auf den im Parameter <Zielsatz> angegebenen nachfolgenden Satz. |

| | | | |
|---|---|---|---|
| <Anzahl der Rücksprungebenen>: | Im Parameter wird die Anzahl der Programmebenen angegeben, die zurückzuspringen sind, um dort den Zielsatz zu suchen und die Programmbearbeitung fortzusetzen. | | |
| | Typ: | INT | |
| | Wert: | 1 | Das Programm wird in der "aktuellen Programmebene -1" fortgesetzt (wie RET ohne Parameter). |
| | | 2 | Das Programm wird in der "aktuellen Programmebene -2" fortgesetzt, d. h., es wird eine Ebene übersprungen. |
| | | 3 | Das Programm wird in der "aktuellen Programmebene -3" fortgesetzt, d. h., es werden zwei Ebenen übersprungen. |
| | | ... | |
| | Wertebereich: | 1 ... 15 | |
| <Rücksprung auf Programmanfang>: | Im Parameter wird angegeben, ob bei einem Rücksprung in das Hauptprogramm und dort aktivem **ISO-Dialekt-Modus** das Programm am Programmanfang fortgesetzt werden soll. | | |
| | Typ: | BOOL | |
| | Wert: | 1 | Wenn der Rücksprung ins Hauptprogramm erfolgt und dort ein **ISO-Dialekt-Modus** aktiv ist, wird auf den Programmanfang verzweigt. |

**Hinweis**

Bei einem Unterprogrammrücksprung mit einer Zeichenkette als Angabe für die Zielsatzsuche wird im aufrufenden Programm immer zuerst nach einer Sprungmarke gesucht.

Wenn ein Sprungziel durch eine Zeichenkette eindeutig definiert sein soll, darf die Zeichenkette daher nicht mit dem Namen einer Sprungmarke übereinstimmen, da sonst der Unterprogrammrücksprung immer auf die Sprungmarke und nicht auf die Zeichenkette ausgeführt wird (siehe Beispiel 2).

## Randbedingungen

Beim Rücksprung über mehrere Programmebenen werden die SAVE-Anweisungen der einzelnen Programmebenen ausgewertet.

Ist bei einem Rücksprung über mehrere Programmebenen ein modales Unterprogramm aktiv und ist in einem der übersprungenen Unterprogramme der Abwahlbefehl MCALL für das modale Unterprogramm programmiert, bleibt das modale Unterprogramm weiterhin aktiv.

**ACHTUNG**

**Programmierfehler**

Bei einem Rücksprung über mehrere Programmebenen liegt es in der alleinigen Verantwortung des Anwenders, dafür zu sorgen, dass mit den erforderlichen modalen Einstellungen fortgesetzt wird. Dies kann z. B. durch Programmierung eines entsprechenden Hauptsatzes erreicht.

## Beispiele

### Beispiel 1: Wiederaufsetzen im Hauptprogramm nach ASUP-Bearbeitung

| **Programmierung** | **Kommentar** |
|---|---|
| N10010 CALL "UP1" | ; Programmebene 0 (Hauptprogramm) |
| N11000 PROC UP1 | ; Programmebene 1 |
| N11010 CALL "UP2" | |
| N12000 PROC UP2 | ; Programmebene 2 |
| ... | |
| N19000 PROC ASUP | ; Programmebene 3 (ASUP-Bearbeitung) |
| ... | |
| N19100 RET("N10900", ,$P_STACK) | ; Unterprogrammrücksprung ins Hauptprogramm |
| | ; $P_STACK: aktuelle Programmebene |
| N10900 | ; Zielsatz im Hauptprogramm |
| N10910 MCALL | ; Modalen Unterprogrammaufruf ausschalten |
| N10920 G0 G60 G40 M5 | ; Weitere modale Einstellungen initialisieren |

### Beispiel 2: Zeichenkette (<String>) als Angabe für die Zielsatzsuche

Hauptprogramm:

| **Programmcode** | **Kommentar** |
|---|---|
| PROC MAIN_PROGRAM | |
| N1000 DEF INT iVar1=1, iVar2=4 | |
| N1010 ... | |
| N1200 subProg1 | ; Aufruf von Unterprogramm "subProg1" |
| N1210 M2 S1000 X10 F1000 | |
| N1220 ..... | |
| N1400 subProg2 | ; Aufruf von Unterprogramm "subProg2" |
| N1410 M3 S500 Y20 | |
| N1420 .. | |
| N1500 lab1: iVar1=R10*44 | |
| N1510 F500 X5 | |
| N1520 ... | |
| N1550 subprog1: G1 X30 | ; "subProg1" ist hier als Sprungmarke definiert. |
| N1560 ... | |
| N1600 subProg3 | ; Aufruf von Unterprogramm "subProg3" |
| N1610 ... | |
| N1900 M30 | |

Unterprogramm subProg1:

| **Programmcode** | **Kommentar** |
|---|---|
| PROC subProg1 | |

| Programmcode | Kommentar |
|---|---|
| N2000 R10=R20+100 | |
| N2010 ... | |
| N2200 RET("subProg2") | ; Rücksprung ins Hauptprogramm auf den Satz N1400 |

Unterprogramm subProg2:

| Programmcode | Kommentar |
|---|---|
| PROC subProg2 | |
| N2000 R10=R20+100 | |
| N2010 ... | |
| N2200 RET("iVar1") | ; Rücksprung ins Hauptprogramm auf den Satz N1500 |

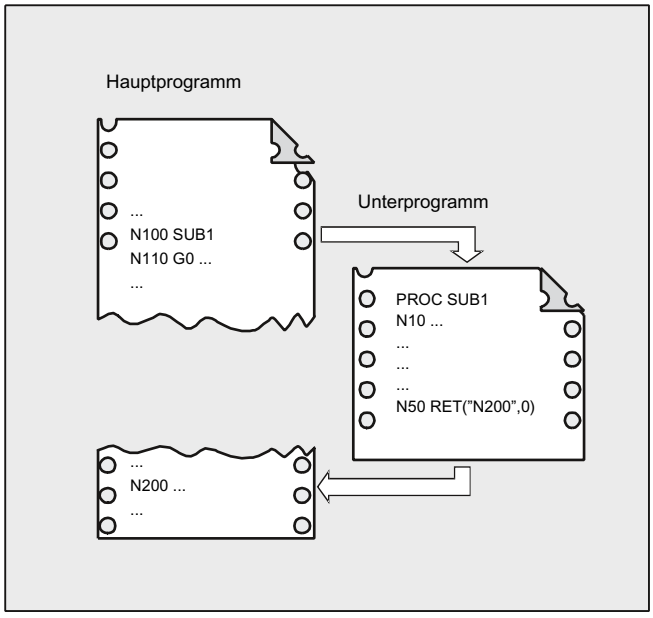
Unterprogramm subProg3:

| Programmcode | Kommentar |
|---|---|
| PROC subProg3 | |
| N2000 R10=R20+100 | |
| N2010 ... | |
| N2200 RET("subProg1") | ; Rücksprung ins Hauptprogramm auf den Satz N1550 |

## Weitere Informationen

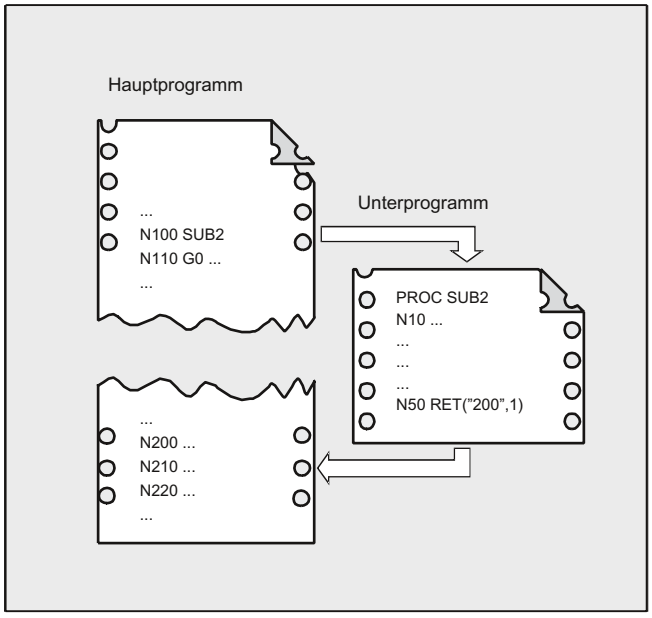
Die folgenden Grafiken zeigen die unterschiedlichen Wirkungen der Rücksprungsparameter

### 1. <Zielsatz> = "N200", <Satz nach Zielsatz> = 0



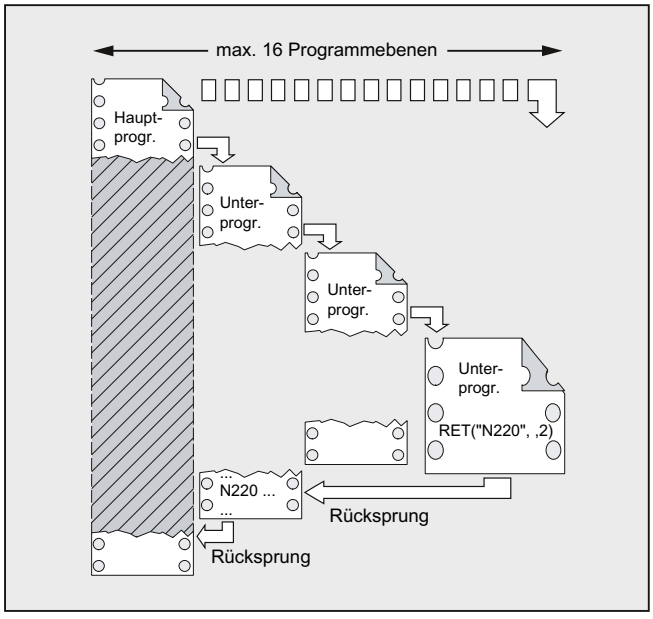
Nach dem RET-Befehl wird die Programmbearbeitung mit dem Satz N200 im Hauptprogramm fortgesetzt.

**2. <Zielsatz> = "N200", <Satz nach Zielsatz> = 1**




Nach dem RET-Befehl wird die Programmbearbeitung mit dem Satz (N210) fortgesetzt, der auf den Satz N200 im Hauptprogramm folgt.

**3. <Zielsatz> = "N220", <Anzahl der Rücksprungebenen> = 2**




Nach dem RET-Befehl wird zwei Programmebenen zurückgesprungen und die Programmbearbeitung wird mit dem Satz N220 fortgesetzt.

### 3.2.2.11 Parametrierbarer Unterprogrammrücksprung (RETB ...)

Im Allgemeinen wird mit dem Befehl `RETB` aus einem Unterprogramm in das aufrufende Programm zurückgesprungen. Die Bearbeitung wird dann mit der auf den Unterprogrammaufruf folgenden Programmzeile fortgesetzt. Soll die Programmbearbeitung an einer anderen Stelle fortgesetzt werden, stehen folgende Möglichkeiten zur Verfügung:

- Fortsetzung der Programmbearbeitung nach Aufruf der Abspanzyklen im ISO-Dialekt-Modus (nach der Konturbeschreibung).
- Rücksprung ins Hauptprogramm aus einer beliebigen Unterprogrammebene (auch nach ASUP) beim Fehlerhandling.
- Rücksprung über mehrere Programmebenen für spezielle Anwendungen in Compile-Zyklen und im ISO-Dialekt-Modus.

Dazu ist der Befehl `RETB` mit weiteren Parametern zu programmieren.

#### Suchrichtung

Bei Angabe des Parameters `<Zielsatz>` erfolgt der Rücksprung zuerst auf den Satz nach dem Aufrufsatz. Anschließend wird das Ziel in Richtung **Anfang** des Programms gesucht, in das zurückgesprungen wird. Ist die Suche nicht erfolgreich, wird als nächstes in Richtung Programmende gesucht.

## Syntax

```
RETB("<Zielsatz>")
RETB("<Zielsatz>",<Satz nach Zielsatz>)
RETB("<Zielsatz>",<Satz nach Zielsatz>,<Anzahl der Rücksprungebenen>)
RETB("<Zielsatz>", ,<Anzahl der Rücksprungebenen>)
RETB("<Zielsatz>",<Satz nach Zielsatz>,<Anzahl der Rücksprungebenen>,
<Rücksprung auf Programmanfang>)
RETB( , ,<Anzahl der Rücksprungebenen>,<Rücksprung auf Programmanfan
g>)
```

## Bedeutung

| | | | |
|---|---|---|---|
| RETB: | Unterprogrammende | | |
| <Zielsatz>: | Im Parameter wird als Sprungziel der Satz angegeben, an dem die Programmbearbeitung fortgesetzt werden soll.<br>Ist der Parameter <Anzahl der Rücksprungebenen> nicht programmiert, dann befindet sich das Sprungziel in dem Programm, aus dem das aktuelle Unterprogramm aufgerufen wurde.<br>Mögliche Angaben sind: | | |
| | <Satznummer> | Nummer des Zielsatzes.<br>Die Suche nach der Satznummer erfolgt im Programm, auf das zurückgesprungen wird, zuerst in Richtung Programmanfang. | |
| | <Sprungmarke> | Sprungmarke, die im Programm, in das zurückgesprungen wird, vorhanden sein muss.<br>Die Suche nach der Sprungmarke erfolgt im Programm, auf das zurückgesprungen wird, zuerst in Richtung Programmanfang. | |
| | <Zeichenkette> | Zeichenkette, die im Programm, in das zurückgesprungen wird, vorhanden sein muss (z. B. Programm- oder Variablenname).<br>Die Suche nach der Zeichenkette erfolgt im Programm, auf das zurückgesprungen wird, zuerst in Richtung Programmanfang.<br>Für die Programmierung der Zeichenkette im Zielsatz gelten folgende Regeln:<br>• **Leerzeichen am Ende** (im Unterschied zur Sprungmarke, die durch einen ":" am Ende gekennzeichnet ist).<br>• **Vor der Zeichenkette** dürfen nur eine Satznummer und/oder eine Sprungmarke gesetzt sein, **keine Programmbefehle**. | |
| <Satz nach Zielsatz>: | Im Parameter wird angegeben, ob die Programmbearbeitung im unter Parameter <Zielsatz> angegebenen Satz oder im darauf folgenden Satz fortgeführt werden soll. | | |
| | Typ: | INT | |
| | Wert: | 0 | Der Rücksprung erfolgt auf den im Parameter <Zielsatz> angegebenen Satz. |
| | | > 0 | Der Rücksprung erfolgt auf den im Parameter <Zielsatz> angegebenen nachfolgenden Satz. |

| | | | |
|---|---|---|---|
| <Anzahl der Rücksprungebenen>: | Im Parameter wird die Anzahl der Programmebenen angegeben, die zurückzuspringen sind, um dort den Zielsatz zu suchen und die Programmbearbeitung fortzusetzen. | | |
| | Typ: | INT | |
| | Wert: | 1 | Das Programm wird in der "aktuellen Programmebene -1" fortgesetzt (wie RET ohne Parameter). |
| | | 2 | Das Programm wird in der "aktuellen Programmebene -2" fortgesetzt, d. h., es wird eine Ebene übersprungen. |
| | | 3 | Das Programm wird in der "aktuellen Programmebene -3" fortgesetzt, d. h., es werden zwei Ebenen übersprungen. |
| | | ... | |
| | Wertebereich: | 1 ... 15 | |
| <Rücksprung auf Programmanfang>: | Im Parameter wird angegeben, ob bei einem Rücksprung in das Hauptprogramm und dort aktivem **ISO-Dialekt-Modus** das Programm am Programmanfang fortgesetzt werden soll. | | |
| | Typ: | BOOL | |
| | Wert: | 1 | Wenn der Rücksprung ins Hauptprogramm erfolgt und dort ein **ISO-Dialekt-Modus** aktiv ist, wird auf den Programmanfang verzweigt. |

**Hinweis**

Bei einem Unterprogrammrücksprung mit einer Zeichenkette als Angabe für die Zielsatzsuche wird im aufrufenden Programm immer zuerst nach einer Sprungmarke gesucht.

Wenn ein Sprungziel durch eine Zeichenkette eindeutig definiert sein soll, darf die Zeichenkette daher nicht mit dem Namen einer Sprungmarke übereinstimmen, da sonst der Unterprogrammrücksprung immer auf die Sprungmarke und nicht auf die Zeichenkette ausgeführt wird (siehe Beispiel 2).

## Randbedingungen

Beim Rücksprung über mehrere Programmebenen werden die SAVE-Anweisungen der einzelnen Programmebenen ausgewertet.

Ist bei einem Rücksprung über mehrere Programmebenen ein modales Unterprogramm aktiv und ist in einem der übersprungenen Unterprogramme der Abwahlbefehl MCALL für das modale Unterprogramm programmiert, bleibt das modale Unterprogramm weiterhin aktiv.

**ACHTUNG**

**Programmierfehler**

Bei einem Rücksprung über mehrere Programmebenen liegt es in der alleinigen Verantwortung des Anwenders, dafür zu sorgen, dass mit den erforderlichen modalen Einstellungen fortgesetzt wird. Dies kann z. B. durch Programmierung eines entsprechenden Hauptsatzes erreicht.

## Beispiel

```
BEISPIEL.MPF
    …
   N3000 START_CYC(parm1, param2, …)
   N3010 TECH_CYC1(param1, param2, …)
   N3020 TECH_CYC2(param1, param2, …)
   N3030 TECH_CYC3(param1, param2, …)
   N3040 END_CYC(param1, param2, …)
   N3040 END_CYC(param1, param2, …)
   N3050 …
   N4500 START_CYC(param11, param12, …)
   N4510 …
   N4590 END_CYC(param11, param12, ..)
   N5000 …
    …
N6000 M30
```

| Programmcode | Kommentar |
|---|---|
| PROC END_CYC(…) | ; Aufruf im Hauptprogramm Zeile N3040 |
| N10000 … | |
| N15000 if status == 1 | |
| N15010 RETB("START_CYC") | ; Rücksprung auf aufrufendes Programm BEISPIEL.MPF |
| | ; Suche nach Zeichenkette "START_CYC" |
| | ; Suchrichtung: rückwärts Richtung Progammanfang |
| | ; Fortsetzung der Programmbearbeitung mit Zeile N3000 |
| N15020 endif | |
| N15030 if status == 0 | |
| N15040 RET | ; Rücksprung auf aufrufendes Programm BEISPIEL.MPF |
| | ; Fortsetzung der Programmbearbeitung mit Zeile N3050 |
| N15050 endif | |
| N16000 RET("START_CYC") | ; Rücksprung auf aufrufendes Programm BEISPIEL.MPF |
| | ; Suche nach Zeichenkette "START_CYC" |
| | ; Suchrichtung: vorwärts Richtung Progammende |
| | ; Fortsetzung der Programmbearbeitung mit Zeile N4500 |
| N17060 RETB | ; Rücksprung auf aufrufendes Programm BEISPIEL.MPF |
| | ; Fortsetzung der Programmbearbeitung mit Zeile N3050 |
| | ; RETB ohne Parameter ist identisch mit RET |

## 3.2.3 Aufruf eines Unterprogramms

### 3.2.3.1 Unterprogrammaufruf ohne Parameterübergabe

Der Aufruf eines Unterprogramms erfolgt entweder mit Adresse L und Unterprogrammnummer oder durch Angabe des Programmnamens.

Auch ein Hauptprogramm kann als Unterprogramm aufgerufen werden. Das im Hauptprogramm gesetzte Programmende `M2` oder `M30` wird in diesem Fall wie `M17` (Programmende mit Rücksprung ins aufrufende Programm) gewertet.

**Hinweis**

Entsprechend kann ein Unterprogramm auch als Hauptprogramm gestartet werden.

Suchstrategie der Steuerung:

Gibt es *_MPF?

Gibt es *_SPF?

Daraus folgt: Falls der Name des aufzurufenden Unterprogramms mit dem Namen des Hauptprogramms identisch ist, dann wird das aufrufende Hauptprogramm wieder aufgerufen. Dieser in der Regel nicht gewünschte Effekt muss durch eindeutige Namenswahl über Unterprogramme und Hauptprogramme vermieden werden.

**Hinweis**

Unterprogramme, die keine Parameterübergabe erfordern, können auch aus einer Initialisierungsdatei aufgerufen werden.

### Syntax

```
L<Nummer>/<Programmname>
```

**Hinweis**

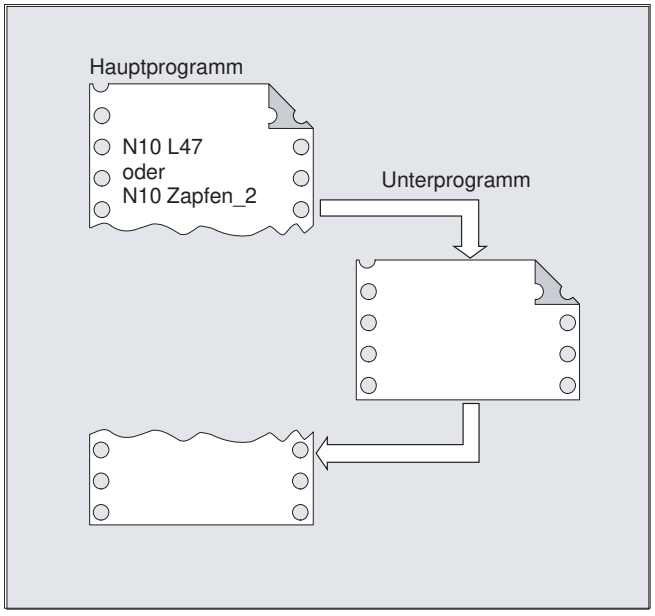
Der Aufruf eines Unterprogramms muss immer im eigenen NC-Satz programmiert werden.

### Bedeutung

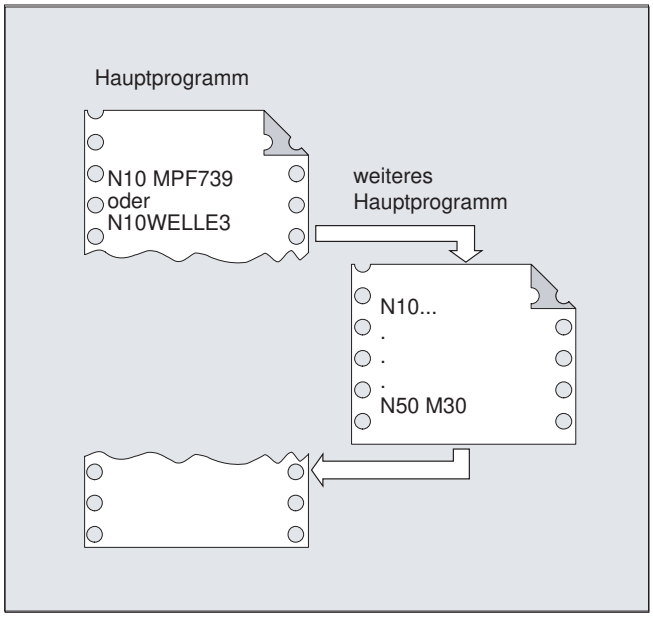
| | | |
|---|---|---|
| `L`: | Adresse für den Unterprogrammaufruf | |
| `<Nummer>`: | Nummer des Unterprogramms | |
| | Typ: | INT |
| | Wert: | Maximal 7 Dezimalstellen<br>**Achtung:**<br>Führende Nullen sind bei der Namensgebung von Bedeutung (⇒ L123, L0123 und L00123 sind drei verschiedene Unterprogramme). |
| `<Programmname>`: | Name des Unterprogramms (oder Hauptprogramms) | |

## Beispiele

### Beispiel 1: Aufruf eines Unterprogramms ohne Parameterübergabe




### Beispiel 2: Aufruf eines Hauptprogramms als Unterprogramm




## Siehe auch

Unterprogramm ohne Parameterübergabe (Seite 503)

### 3.2.3.2 Unterprogrammaufruf mit Parameterübergabe (EXTERN)

Beim Unterprogrammaufruf mit Parameterübergabe können Variablen oder Werte direkt übergeben werden (nicht bei VAR-Parametern).

Unterprogramme mit Parameterübergabe müssen vor dem Aufruf im Hauptprogramm mit EXTERN bekannt gemacht werden (z. B. am Programmanfang). Angegeben werden dabei der Name des Unterprogramms und die Variablentypen in der Reihenfolge der Übergabe.

| ACHTUNG |
| --- |
| **Verwechslungsgefahr**<br>Sowohl die Variablentypen als auch die Reihenfolge der Übergabe muss mit den Definitionen, die im Unterprogramm unter PROC vereinbart wurden, übereinstimmen. Die Parameternamen können in Haupt- und Unterprogramm unterschiedlich sein. |

## Syntax

```
EXTERN <Programmname>(<Typ_Par1>,<Typ_Par2>,<Typ_Par3>)
...
<Programmname>(<Wert_Par1>,<Wert_Par2>,<Wert_Par3>)
```

**Hinweis**

Der Unterprogrammaufruf muss immer im eigenen NC-Satz programmiert werden.

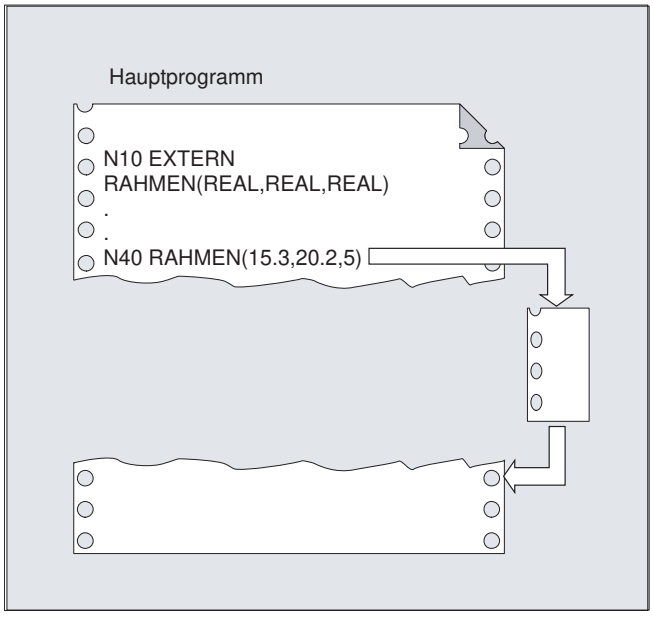
## Bedeutung

| | |
| --- | --- |
| <Programmname>: | Name des Unterprogramms |
| EXTERN: | Schlüsselwort für die Bekanntmachung eines Unterprogramms mit Parameterübergabe<br>**Hinweis:**<br>EXTERN muss nur dann angegeben werden, wenn das Unterprogramm im Werkstück- oder im globalen Unterprogrammverzeichnis steht. Zyklen müssen nicht als EXTERN erklärt werden. |
| <Typ_Par1>,<Typ_Par2>,<Typ_Par3>: | Variablentypen der zu übergebenden Parameter in der Reihenfolge der Übergabe |
| <Wert_Par1>,<Wert_Par2>,<Wert_Par3>: | Variablenwerte für die zu übergebenden Parameter |

## Beispiele

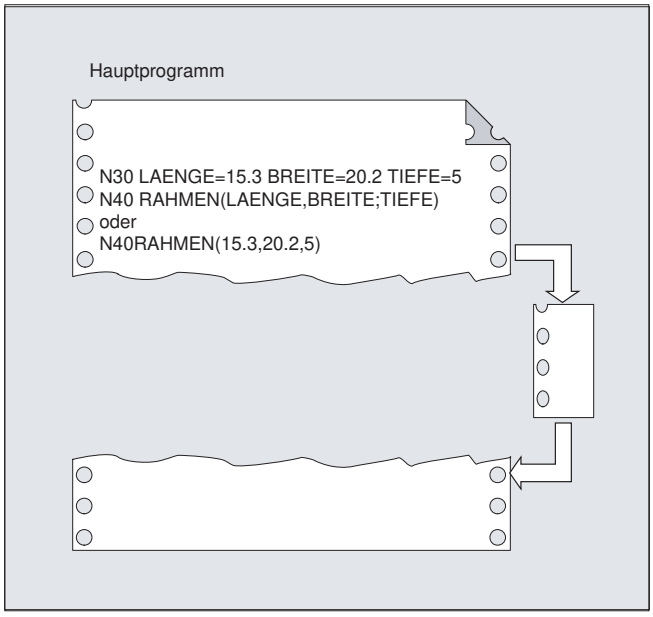
### Beispiel 1: Unterprogrammaufruf mit vorhergehender Bekanntmachung

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 EXTERN RAHMEN(REAL,REAL,REAL)` | `; Angabe des Unterprogramms.` |
| `...` | |
| `N40 RAHMEN(15.3,20.2,5)` | `; Aufruf des Unterprogramms mit Parameterübergabe.` |




### Beispiel 2: Unterprogrammaufruf ohne Bekanntmachung

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 DEF REAL LAENGE, BREITE, TIEFE` | |
| `N20 …` | |
| `N30 LAENGE=15.3 BREITE=20.2 TIEFE=5` | |
| `N40 RAHMEN(LAENGE,BREITE,TIEFE)` | `; oder: N40 RAHMEN(15.3,20.2,5)` |




## Siehe auch

Unterprogramm mit Parameterübergabe Call-by-Value (PROC) (Seite 504)

Unterprogramm mit Parameterübergabe Call-by-Reference (PROC, VAR) (Seite 506)

### 3.2.3.3 Anzahl der Programmwiederholungen (P)

Soll ein Unterprogramm mehrfach hintereinander abgearbeitet werden, kann im Satz mit dem Unterprogrammaufruf unter der Adresse `P` die gewünschte Anzahl der Programmwiederholungen programmiert werden.

**⚠ VORSICHT**

**Unterprogrammaufruf mit Programmwiederholung und Parameterübergabe**

Parameter werden nur beim Programmaufruf bzw. ersten Durchlauf übergeben. Für die weiteren Wiederholungen bleiben die Parameter unverändert. Falls Sie bei Programmwiederholungen die Parameter verändern wollen, müssen Sie im Unterprogramm entsprechende Vereinbarungen festlegen.

## Syntax

```
<Programmname> P<Wert>
```

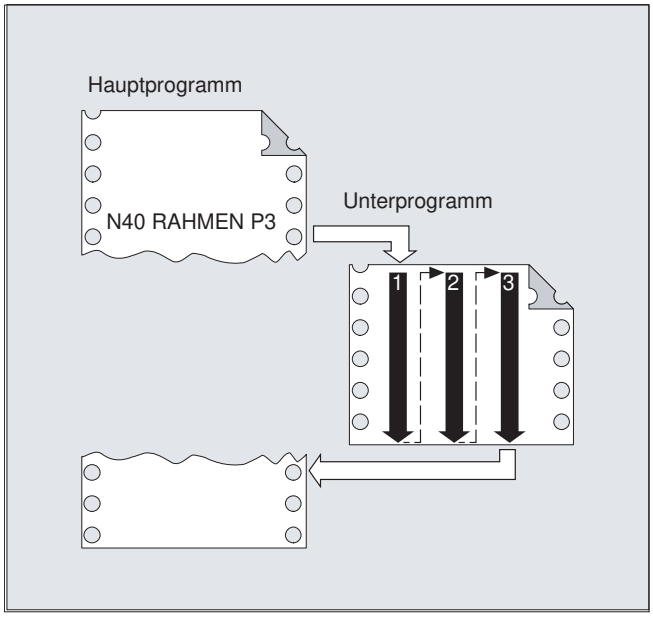
## Bedeutung

| | |
|---|---|
| `<Programmname>`: | Unterprogrammaufruf |
| `P`: | Adresse für die Programmierung von Programmwiederholungen |

| | | |
|---|---|---|
| <Wert>: | Anzahl der Programmwiederholungen | |
| | Typ: | INT |
| | Wertebereich: | 1 … 9999<br>(ohne Vorzeichen) |

## Beispiel

| Programmcode | Kommentar |
|---|---|
| ... | |
| N40 RAHMEN P3 | ; Das Unterprogramm RAHMEN soll dreimal hintereinander abgearbeitet werden. |
| ... | |




### 3.2.3.4 Modaler Unterprogrammaufruf (MCALL)

Durch den modalen Unterprogrammaufruf `MCALL <ProgName>` wird das angegebene Unterprogramm nicht sofort aufgerufen. Stattdessen erfolgt der Aufruf ab diesem Zeitpunkt im Teileprogramm automatisch nach jedem Verfahrsatz mit Bahnbewegungen. Auch über Programmebenen hinweg.

**Hinweis**

In einem Programmablauf wirkt immer nur der letzte modale Unterprogrammaufruf `MCALL <ProgName>`. Der aktuelle modale Unterprogrammaufruf ersetzt den bisher aktiven.

Werden Parameter an das Unterprogramm übergeben, erfolgt die Parameterübergabe nur beim Aufruf:

```
MCALL(<ProgName>(Par1, Par2, ...))
```

**Hinweis**

Der Datendefinitionsbereich am Anfang des Unterprogramms wird nur einmal bei der Ausführung des Satzes mit dem programmierten MCALL-Aufruf durchlaufen. Bei den folgenden Unterprogrammaufrufen nach den Verfahrsätzen wird der Datendefinitionsbereich nicht mehr durchlaufen. Dies hat zur Folge, dass ein bei der Definition einer lokalen Anwendervariablen (LUD) zugewiesener Wert nach dem ersten Aufruf nicht mehr zu Verfügung steht, sondern der Wert, der im Zyklus zuletzt geschrieben wurde. Die Variable wird nach dem ersten Aufruf nicht neu angelegt, sondern behält den letzten Wert aus dem vorherigen Aufruf. Um dieses Verhalten zu umgehen, wird empfohlen, die LUD-Definition und -Wertzuweisung voneinander zu trennen und die Wertzweisung in einem eigenen Satz nach dem Datendefinitionsbereich zu programmieren.

| ACHTUNG |
| --- |
| **Modale Unterprogrammaufrufe ohne Bahnbewegung**<br>Das modale Unterprogramm wird in folgenden Situationen auch ohne Programmierung einer Bahnbewegung aufgerufen:<br>• Programmierung der Adressen S oder F, wenn G0 oder G1 aktiv ist<br>• Wenn G0 oder G1 allein im Satz steht oder zusammen mit weiteren G-Befehlen programmiert wurde. |

## Syntax

```
MCALL <ProgName>
...
MCALL
```

## Bedeutung

| | |
| --- | --- |
| `MCALL <ProgName>`: | Funktion "Modaler Unterprogrammaufruf" einschalten |
| `<ProgName>`: | Name des Unterprogramms |
| `MCALL`: | Mit MCALL ohne Angabe eines Programmnamens wird die Funktion "Modaler Unterprogrammaufruf" ausgeschaltet. |

## Randbedingungen

### ASUP

Wird die Bearbeitung eines Teileprogramms durch ein ASUP (siehe Kapitel "Interruptroutine (ASUP) (Seite 547)") unterbrochen, werden in diesem ASUP keine modalen Unterprogrammaufrufe ausgeführt.

Wird ein ASUP im Kanalzustand "Reset" gestartet, verhält es sich bezüglich der modalen Unterprogrammaufrufe wie ein normales Teileprogramm.

### Werkzeugwechsel-Zyklus

Erfolgt im Werkzeugwechsel-Zyklus eine Abwahl der Funktion "Modaler Unterprogrammaufruf", so ist zu beachten, dass der Werkzeugwechsel-Zyklus ggf. auch nach Satzsuchlauf implizit über das Suchlauf-ASUP oder manuell per Überspeichern aufgerufen wird. In dieser Situation darf die Funktion "Modaler Unterprogrammaufruf" nicht abgewählt werden, weil sonst das Suchlauf-Ergebnis verfälscht wird. Es wird deshalb empfohlen, die Abwahl der Funktion "Modaler Unterprogrammaufruf" im Werkzeugwechsel-Zyklus wie folgt zu programmieren:

```
Programmcode                Kommentar
...
  IF $AC_ASUP == 0          ; Aufruf erfolgt nicht über Suchlauf-ASUP oder Überspei-
                              chern.
    MCALL                   ; Funktion "Modaler Unterprogrammaufruf" ausschalten.
  ENDIF
...
```

## Beispiele

### Beispiel 1

```
Programmcode                Kommentar
N10 G0 X0 Y0
N20 MCALL L70               ; Modalen Unterprogrammaufruf für L70 einschalten.
N30 X10 Y10                 ; X10 Y10 wird angefahren, anschließend wird L70 aufgerufen.
N40 X20 Y20                 ; X20 Y20 wird angefahren, anschließend wird L70 aufgerufen.
...
N100 MCALL                  ; Funktion "Modaler Unterprogrammaufruf" ausschalten.
N110 X0 Y0                  ; X0 Y0 wird angefahren, L70 wird nicht aufgerufen.
```

### Beispiel 2

```
Programmcode
N10 G0 X0 Y0
N20 MCALL L70
N30 L80
```

In diesem Beispiel stehen die nachfolgenden NC-Sätze mit programmierten Bahnachsen in Unterprogramm L80. L70 wird durch L80 aufgerufen.

### 3.2.3.5 Indirekter Unterprogrammaufruf (CALL)

In Abhängigkeit von den gegebenen Bedingungen können an einer Stelle unterschiedliche Unterprogramme aufgerufen werden. Hierzu wird der Name des Unterprogramms in einer Variablen vom Typ STRING hinterlegt. Der Unterprogrammaufruf erfolgt mit `CALL` und dem Variablennamen.

**Hinweis**

Der indirekte Unterprogrammaufruf ist nur für Unterprogramme ohne Parameterübergabe möglich. Für den direkten Aufruf eines Unterprogramms hinterlegen Sie den Namen in einer STRING-Konstanten.

## Syntax

```
CALL <Programmname>
```

## Bedeutung

| | | |
|---|---|---|
| `CALL`: | Befehl für den indirekten Unterprogrammaufruf | |
| `<Programmname>`: | Name des Unterprogramms (Variable oder Konstante) | |
| | Typ: | STRING |

## Beispiel

### Direkter Aufruf mit STRING-Konstante:

| **Programmcode** | **Kommentar** |
|---|---|
| … | |
| `CALL "/_N_WKS_DIR/_N_SUBPROG_WPD/_N_TEIL1_SPF"` | `; Unterprogramm TEIL1 mit CALL direkt aufrufen.` |
| … | |

### Indirekter Aufruf über Variable:

| **Programmcode** | **Kommentar** |
|---|---|
| … | |
| `DEF STRING[100] PROGNAME` | `; Variable definieren.` |
| `PROGNAME="/_N_WKS_DIR/_N_SUBPROG_WPD/_N_TEIL1_SPF"` | `; Unterprogramm TEIL1 der Variablen PROGNAME zuordnen.` |
| `CALL PROGNAME` | `; Unterprogramm TEIL1 über CALL und die Variable PROGNAME indirekt aufrufen.` |
| … | |

### 3.2.3.6 Indirekter Unterprogrammaufruf mit Angabe des auszuführenden Programmteils (CALL BLOCK ... TO ...)

Mit CALL und der Schlüsselwortkombination BLOCK ... TO wird ein Unterprogramm indirekt aufgerufen und der mit Start- und Endmarke gekennzeichnete Programmteil ausgeführt.

## Syntax

```
CALL <Programmname> BLOCK <Startmarke> TO <Endmarke>
CALL BLOCK <Startmarke> TO <Endmarke>
```

## Bedeutung

| | | |
|---|---|---|
| CALL: | Befehl für den indirekten Unterprogrammaufruf | |
| <Programmname>: | Name des Unterprogramms (Variable oder Konstante), das den zu bearbeitenden Programmteil enthält (**Angabe optional**). | |
| | Typ: | STRING |
| | **Hinweis:**<br>Ist kein <Programmname> programmiert, wird der mit <Startmarke> und <Endmarke> gekennzeichnete Programmteil im aktuellen Programm gesucht und ausgeführt. | |
| BLOCK ... TO ... : | Schlüsselwortkombination für indirekte Programmteilausführung | |
| <Startmarke>: | Variable, die auf den Beginn des zu bearbeitenden Programmteils verweist. | |
| | Typ: | STRING |
| <Endmarke>: | Variable, die auf das Ende des zu bearbeitenden Programmteils verweist. | |
| | Typ: | STRING |

## Beispiel

### Hauptprogramm:

| **Programmcode** | **Kommentar** |
|---|---|
| ... | |
| DEF STRING[20] STARTLABEL, ENDLABEL | ; Variablendefinition für die Start- und Endmarke. |
| STARTLABEL="LABEL_1" | |
| ENDLABEL="LABEL_2" | |
| ... | |
| CALL "CONTUR_1" BLOCK STARTLABEL TO ENDLABEL | ; Indirekter Unterprogrammaufruf und Kennzeichnung des auszuführenden Programmteils. |
| ... | |

**Unterprogramm:**

| Programmcode | Kommentar |
|---|---|
| PROC CONTUR_1 ... | |
| LABEL_1 | ; Startmarke: Beginn der Programmteilausführung |
| N1000 G1 ... | |
| ... | |
| LABEL_2 | ; Endmarke: Ende der Programmteilausführung |
| ... | |

### 3.2.3.7 Indirekter Aufruf eines in ISO-Sprache programmierten Programms (ISOCALL)

Mit dem indirekten Programmaufruf ISOCALL kann ein in einer ISO-Sprache programmiertes Programm aufgerufen werden. Dabei wird der in den Maschinendaten eingestellte ISO-Modus aktiviert. Am Programmende wird wieder der ursprüngliche Bearbeitungsmodus wirksam. Ist in den Maschinendaten kein ISO-Modus eingestellt, erfolgt der Aufruf des Unterprogramms im Siemens-Modus.

**Weitere Informationen** zum ISO-Modus: Funktionshandbuch ISO-Dialekte

## Syntax

```
ISOCALL <Programmname>
```

## Bedeutung

| | |
|---|---|
| ISOCALL: | Schlüsselwort für indirekten Unterprogrammaufruf, mit dem der in den Maschinendaten eingestellte ISO-Modus aktiviert wird |
| <Programmname>: | Name des in einer ISO-Sprache programmierten Programms (Variable oder Konstante vom Typ STRING) |

## Beispiel: Kontur mit Zyklenprogrammierung aus dem ISO-Modus heraus aufrufen

| Programmcode | Kommentar |
|---|---|
| 0122_SPF | ; Konturbeschreibung im ISO-Modus |
| N1010 G1 X10 Z20 | |
| N1020 X30 R5 | |
| N1030 Z50 C10 | |
| N1040 X50 | |
| N1050 M99 | |
| N0010 DEF STRING[5] PROGNAME = "0122" | ; Siemens-Teileprogramm (-Zyklus) |
| ... | |
| N2000 R11 = $AA_IW[X] | |
| N2010 ISOCALL PROGNAME | |
| N2020 R10 = R10+1 | ; Programm 0122.spf im ISO-Modus bearbeiten |
| ... | |

| Programmcode | Kommentar |
|---|---|
| N2400 M30 | |

### 3.2.3.8 Unterprogramm mit Pfadangabe und Parametern aufrufen (PCALL)

Mit PCALL können Unterprogramme mit absoluter Pfadangabe und Parameterübergabe aufgerufen werden.

#### Syntax

```
PCALL <Pfad/Programmname>(<Parameter 1>,…,<Parameter n>)
```

#### Bedeutung

| | |
|---|---|
| PCALL: | Schlüsselwort für Unterprogrammaufruf mit absoluter Pfadangabe. |
| <Pfad/Programmname>: | Absolute Pfadangabe einschließlich Unterprogrammnamen.<br>Regeln zur Pfadangabe siehe "Adressierung von Dateien des Programmspeichers (Seite 563)".<br>Wurde kein absoluter Pfad angegeben, verhält sich PCALL wie ein Standard-Unterprogrammaufruf mit Programmbezeichner.<br>Der Programmname wird ohne Präfix und ohne Datei-Kennung angegeben. Soll der Programmname mit Präfix und Datei-Kennung programmiert werden, so muss er explizit mit Präfix und Datei-Kennung mit dem Befehl EXTERN erklärt werden. |
| <Parameter 1>, ...: | Aktual-Parameter entsprechend der PROC-Anweisung des Unterprogramms. |

#### Beispiel

| Programmcode |
|---|
| PCALL/_N_WKS_DIR/_N_WELLE_WPD/WELLE(parameter1,parameter2,…) |

### 3.2.3.9 Suchpfad bei Unterprogrammaufrufen erweitern (CALLPATH)

Mit dem Befehl CALLPATH kann der Suchpfad für Unterprogrammaufrufe erweitert werden. Damit können auch Unterprogramme aus einem nicht ausgewählten Werkstückverzeichnis aufgerufen werden, ohne den vollständigen, absoluten Pfadnamen des Unterprogramms anzugeben.

Eine weitere Anwendungsmöglichkeit bietet sich im EES-Betriebsmodus "EES ohne GDIR", wenn ein Verzeichnis auf einem externen Programmspeicher zur Ablage globaler Unterprogramme genutzt wird. In diesem Fall kann mit CALLPATH der Suchpfad um dieses Unterprogrammverzeichnis erweitert werden.

Die Suchpfaderweiterung erfolgt vor dem Eintrag für Anwenderzyklen (_N_CUS_DIR).

Durch folgende Ereignisse wird die Suchpfaderweiterung wieder abgewählt:

- `CALLPATH` mit Leerzeichen
- `CALLPATH` ohne Parameter
- Teileprogrammende
- Reset

## Syntax

```
CALLPATH("<Pfadname>")
```

## Bedeutung

| | |
|---|---|
| `CALLPATH`: | Schlüsselwort für die programmierbare Suchpfaderweiterung.<br>Wird in einer eigenen Teileprogrammzeile programmiert. |
| `<Pfadname>`: | Konstante oder Variable vom Typ STRING.<br>Enthält die absolute Pfadangabe des Verzeichnisses, um das der Suchpfad erweitert werden soll.<br>Regeln zur Pfadangabe siehe "Adressierung von Dateien des Programmspeichers (Seite 563)". |

## Beispiel

Der Suchpfad soll um ein bestimmtes Werkstückverzeichnis erweitert werden:

**Programmcode**

```
...
CALLPATH("/_N_WKS_DIR/_N_MYWPD_WPD")
...
```

Damit wird folgender Suchpfad eingestellt (Position 5. ist neu):

1. Aktuelles Verzeichnis/*name*
2. Aktuelles Verzeichnis/*name_SPF*
3. Aktuelles Verzeichnis/*name_MPF*
4. //NC:/_N_SPF_DIR/*name_SPF*
5. **/_N_WKS_DIR/_N_MYWPD_WPD/name_SPF**
6. /N_CUS_DIR/*name_SPF*
7. /_N_CMA_DIR/*name_SPF*
8. /_N_CST_DIR/*name_SPF*

### Randbedingungen

- CALLPATH prüft, ob der programmierte Pfadname tatsächlich vorhanden ist. Im Fehlerfall wird die Teileprogrammbearbeitung mit Korrektursatz-Alarm 14009 abgebrochen.
- CALLPATH kann auch in INI-Dateien programmiert werden. Er wirkt dann für die Bearbeitungsdauer der INI-Datei (WPD-INI-Datei oder Initialisierungsprogramm für NC-aktive Daten, z. B. Frames im 1. Kanal _N_CH1_UFR_INI). Danach wird der Suchpfad wieder zurückgesetzt.

## 3.2.3.10 Externes Unterprogramm abarbeiten (EXTCALL)

Mit dem Befehl EXTCALL kann ein Teileprogramm von einem externen Speicher nachgeladen und abgearbeitet werden.

Als externer Speicher stehen zur Verfügung:

- Lokales Laufwerk
- Netzlaufwerk
- USB-Laufwerk

**Hinweis**

Als Schnittstelle zum Abarbeiten eines auf einem USB-Laufwerk befindlichen externen Programms dürfen nur die USB-Schnittstellen an der Bedientafelfront bzw. TCU verwendet werden.

**ACHTUNG**

**Werkzeug-/Werkstückbeschädigung durch USB-FlashDrive**

Es wird empfohlen zum Abarbeiten eines externen Unterprogramms kein USB-FlashDrive zu verwenden. Ein Kommunikationsabbruch zum USB-FlashDrive während der Abarbeitung des Teileprogramms durch Kontaktschwierigkeiten, Herausfallen, Abbrechen durch Anstoßen oder versehentliches Abziehen führt zum sofortigen Stop der Bearbeitung. Werkzeug und/oder Werkstück könnten dabei beschädigt werden.

#### Voreinstellung des externen Programmpfades

Der Pfad zum externen Programmverzeichnis kann voreingestellt mit dem Settingdatum:

SD42700 $SC_EXT_PROG_PATH

Zusammen mit dem beim EXTCALL-Aufruf angegebenen Programmpfad und -bezeichner ergibt sich daraus der Gesamtpfad des aufzurufenden Teileprogramms.

**Hinweis**

Soll der Programmpfad nur über den EXTCALL-Aufruf angegeben werden, muss SD42700 leer sein!

**Hinweis**

**Parameter**

Beim Aufruf eines externen Programms können diesem keine Parameter übergeben werden.

## Syntax

```
EXTCALL("<Pfad/><Programmname>")
```

## Bedeutung

| | | |
|---|---|---|
| EXTCALL: | Befehl zum Aufrufen eines externen Unterprogramms | |
| "<Pfad/><Programmname>": | Konstante/Variable vom Typ STRING | |
| | <Pfad/>: | Absolute oder relative Pfadangabe (**optional**) |
| | <Programmname>: | Der Programmname wird ohne Präfix "_N_" angegeben.<br>Die Dateierweiterung ("MPF", "SPF") kann mit dem Zeichen "_" oder "." am Programmnamen angefügt werden (**optional**).<br>Beispiel:<br>"WELLE"<br>"WELLE_SPF"<br>"WELLE.SPF" |

### Pfadangabe: Kurzbezeichnungen

Folgende Kurzbezeichnungen können bei der Pfadangabe verwendet werden:

- Lokales Laufwerk: "**LOCAL_DRIVE:**"
- CF-Karte: "**CF_CARD:**"
- USB-Laufwerk (Bedientafelfront): "**USB:**"

Die Kurzbezeichnungen "CF_CARD:" und "LOCAL_DRIVE:" sind alternativ verwendbar.

## Beispiel

### Abarbeiten von lokalem Laufwerk

Das Hauptprogramm "MAIN.MPF" befindet sich im NC-Speicher und ist zur Abarbeitung angewählt.

### Unterprogramm "SP_1"

Das externe Unterprogramm "SP_1.SPF" bzw. "SP_1.MPF" befindet sich auf dem lokalen Laufwerk in dem Verzeichnis "/user/sinumerik/data/prog/WKS.DIR/WST1.WPD".

Der Pfad zum externen Programmverzeichnis ist einzustellen mit:

SD42700 $SC_EXT_PROG_PATH = LOCAL_DRIVE:WKS.DIR/WST1.WPD

---

**Hinweis**

Pfadangabe für den Aufruf des externen Unterprogramms:

- Ohne Verwendung der Voreinstellung: "LOCAL_DRIVE:WKS.DIR/WST1.WPD/SP_1"
- Mit Verwendung der Voreinstellung: "SP_1"

---

#### Unterprogramm "SP_2"

Das externe Unterprogramm "SP_2.SPF" bzw. "SP_2.MPF" befindet sich im Verzeichnis WKS.DIR/WST1.WPD des USB-Laufwerks. Die Voreinstellung des Pfads zum externen Programmverzeichnis wird für den Pfad des Unterprogramms "SP_1" verwendet und wird im Hauptprogramm auch nicht umgeschrieben. Daher muss beim Aufruf des Unterprogramms "SP_2" der vollständige Pfad angegeben werden.

#### Hauptprogramm "MAIN"

```
Programmcode
N010 PROC MAIN
N020 ...
N030 EXTCALL("SP_1")
N030 EXTCALL("USB:WKS.DIR/WST1.WPD/SP_2")
N050 ...
N060 M30
```

## Weitere Informationen

#### EXTCALL-Aufruf mit absoluter Pfadangabe

Ist das Unterprogramm unter dem angegebenen Pfad vorhanden, wird es mit dem EXTCALL-Aufruf ausgeführt. Ist das Unterprogramm unter dem angegebenen Pfad nicht vorhanden, wird die Programmausführung mit dem EXTCALL-Aufruf abgebrochen.

#### EXTCALL-Aufruf mit relativer Pfadangabe / ohne Pfadangabe

Bei einem EXTCALL-Aufruf mit relativer Pfadangabe bzw. ohne Pfadangabe werden die vorhandenen Programmspeicher nach folgendem Muster durchsucht:

1. Ist in SD42700 $SC_EXT_PROG_PATH eine Pfadangabe voreingestellt, wird zuerst ausgehend von diesem Pfad nach der Angabe im EXTCALL-Aufruf (Programmname ggf. mit relativer Pfadangabe) gesucht. Der absolute Pfad ergibt sich dann durch Zeichenverkettung aus:
   - Voreingestellten Pfadangabe im SD42700 $SC_EXT_PROG_PATH
   - Trennzeichen "/"
   - Pfadangabe und Unterprogrammname im Befehl EXTCALL
2. Wurde das Unterprogramm unter 1. nicht gefunden, werden die Verzeichnisse des Anwenderspeichers durchsucht.

Die Suche endet, wenn das Unterprogramm erstmalig gefunden wurde. Wird das Unterprogramm nicht gefunden, wird die Programmausführung mit dem EXTCALL-Aufruf abgebrochen.

### Einstellbarer Nachladespeicher (FIFO-Puffer)

Für das Abarbeiten eines externen Unterprogramms wird ein Nachladespeicher benötigt. Die Größe des Nachladespeichers ist mit 30 kByte voreingestellt und kann nur vom Maschinenhersteller verändert werden (über MD18360 MM_EXT_PROG_BUFFER_SIZE).

**Hinweis**

**Unterprogramme mit Sprunganweisungen**

Bei externen Unterprogrammen, die Spunganweisungen enthalten (`GOTOF`, `GOTOB`, `CASE`, `FOR`, `LOOP`, `WHILE`, `REPEAT`, `IF`, `ELSE`, `ENDIF` etc.), müssen die Sprungziele innerhalb des Nachladespeichers liegen.

**Hinweis**

**ShopMill-/ShopTurn-Programme**

ShopMill- und ShopTurn-Programme müssen wegen der am Dateiende angefügten Konturbeschreibungen vollständig im Nachladespeicher abgelegt sein.

Für parallel abgearbeitete externe Unterprogramme wird jeweils ein eigener Nachladespeicher benötigt.

### Reset / Programmende / POWER ON

Durch Reset und POWER ON werden externe Unterprogrammaufrufe abgebrochen und die jeweiligen Nachladespeicher gelöscht.

Ein für "Abarbeiten von Extern" selektiertes Programm bleibt über Reset, Programmende oder POWER ON hinaus weiter für "Abarbeiten von Extern" angewählt. Das Verhalten unterscheidet sich nicht zu intern angewählten Programmen, sofern der externe Programmspeicher weiterhin zur Verfügung steht.

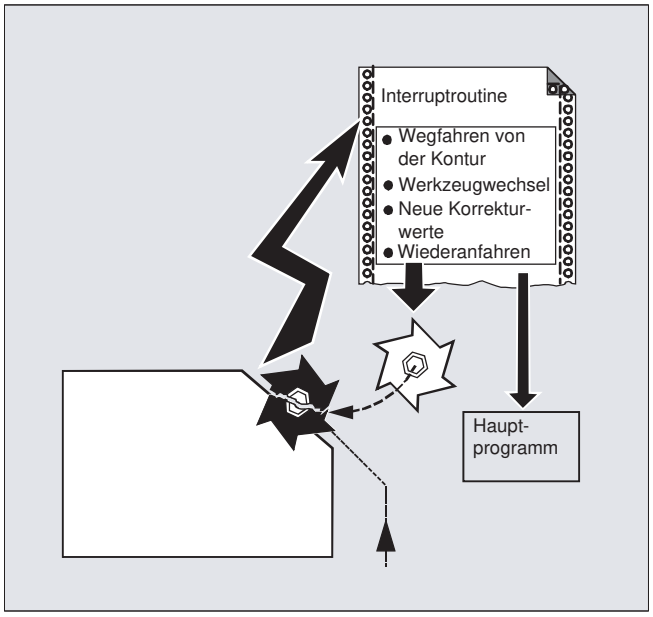
# 3.3 Interruptroutine (ASUP)

## 3.3.1 Funktion einer Interruptroutine

**Hinweis**

Die in der folgenden Beschreibung abwechselnd vorkommenden Begriffe "Asynchrones Unterprogramm (ASUP)" und "Interruptroutine" kennzeichnen die gleiche Funktionalität.

Die Funktion einer Interruptroutine soll anhand eines typischen Beispiels verdeutlicht werden:




Während der Bearbeitung bricht das Werkzeug. Hierdurch wird ein Signal ausgelöst, das den laufenden Bearbeitungsablauf stoppt und gleichzeitig ein Unterprogramm – die so genannte Interruptroutine – startet. In diesem Unterprogramm stehen alle Anweisungen, die in diesem Fall ausgeführt werden sollen.

Ist das Unterprogramm abgearbeitet (und hierdurch die Betriebsbereitschaft hergestellt), springt die Steuerung in das Hauptprogramm zurück und setzt die Bearbeitung – je nach REPOS-Befehl – an der Unterbrechungsstelle fort (siehe "Wiederanfahren an Kontur (Seite 869)").

**⚠ VORSICHT**

**Kollisionsgefahr**

Wenn im Unterprogramm kein REPOS-Befehl programmiert ist, dann wird auf den Endpunkt des Satzes positioniert, der auf den unterbrochenen Satz folgt.

## 3.3.2 Interruptroutine erstellen

### Interruptroutine als Unterprogramm erstellen

Die Interruptroutine wird bei der Definition wie ein Unterprogramm gekennzeichnet.

Beispiel:

```
Programmcode        Kommentar
PROC ABHEB_Z        ; Programmname "ABHEB_Z"
N10 ...             ; Danach folgen die NC-Sätze.
...
N50 M17             ; Zum Schluss Programmende und Rückkehr ins Hauptprogramm.
```

### Modale G-Befehle sichern (SAVE)

Die Interruptroutine kann bei der Definition mit SAVE gekennzeichnet werden.

Das Attribut SAVE bewirkt, dass die vor dem Aufruf der Interruptroutine aktiven modalen G-Befehle gesichert und nach dem Ende der Interruptroutine wieder reaktiviert werden (siehe " Unterprogramme mit SAVE-Mechanismus (SAVE) (Seite 508) ").

Dadurch ist es möglich, die Bearbeitung nach Ablauf der Interruptroutine an der Unterbrechungsstelle fortzusetzen.

Beispiel:

```
Programmcode
PROC ABHEB_Z SAVE
N10 ...
...
N50 M17
```

### Weitere Interruptroutinen zuordnen (SETINT)

Innerhalb der Interruptroutine können SETINT-Anweisungen (siehe "Interruptroutine zuordnen und starten (SETINT)" (Seite 548)) programmiert und hierdurch weitere Interruptroutinen scharf geschaltet werden. Das Auslösen erfolgt erst durch den Eingang.

## 3.3.3 Interruptroutine zuordnen und starten (SETINT, PRIO, BLSYNC)

Die Steuerung verfügt über mehrere schnelle Eingänge (Eingang 1 ... 8), die jeweils einen Interrupt (1 ... 8) auslösen. Jedem Interrupt kann über den Befehl `SETINT` eine Priorität und eine Interrupt-Routine zugeordnet werden. Wird der Interrupt durch Setzen des schnellen Eingangs ausgelöst, wird die aktuelle Bearbeitung im Kanal unterbrochen und die Interrupt-Routine gestartet.

### Interrupt-Priorität

Werden in einem Teileprogramm mehreren Eingängen Interrupts zugeordnet, müssen den Interrupts unterschiedliche Prioritäten zugewiesen werden.

Einem Interrupt kann ein Prioritätswerte von 1 ... 128 zugeordnet werden. Prioritätswert 1 entspricht der höchsten, 128 der niedrigsten Priorität.

## Syntax

```
SETINT(<n>) <NAME>
SETINT(<n>) PRIO=<Wert> <NAME>
SETINT(<n>) PRIO=<Wert> <NAME> BLSYNC
SETINT(<n>) PRIO=<Wert> <NAME> LIFTFAST
```

## Bedeutung

| | | |
|---|---|---|
| `SETINT(<n>):` | Dem Interruptsignal <n> wird NC-Programm (ASUP) <Name> zuordnen. Die zugeordnete Interruptroutine wird gestartet, sobald Interruptsignal <n> == 1 erkannt wird.<br>**Hinweis:**<br>Wird einem Interruptsignal <n> eine andere Interrupt-Routine zugeordnet, wird die vorhergende Zuordnung unwirksam. | |
| `<n>`: | Nummer des Interruptsignals | |
| | Typ: | INT |
| | Wertebereich: | 1 ... 32 |
| `PRIO=`: | Priorität des Interrupts<br>(optional) | |
| `<Wert>`: | (**optional**) Prioritätswert | |
| | Typ: | INT |
| | Wertebereich: | 1 ... 128 (1 ⇒ höchste Priorität) |
| `<NAME>`: | Name des NC-Programms (ASUP) | |
| `BLSYNC`: | (**optional**) `BLSYNC` bewirkt, dass nach dem Auslösen des Interrupts zuerst gewartet wird, bis der aktuelle Satz abgearbeitet ist. Erst danach wird die Interrupt-Routine ausgeführt. | |
| `LIFTFAST`: | (**optional**) `LIFTFAST` bewirkt, dass nach dem Auslösen des Interrupts zuerst ein Schnellabheben erfolgt ( siehe Kapitel "Schnellabheben von der Kontur (SETINT LIFTFAST, ALF) (Seite 552)"). Erst danach wird die Interrupt-Routine ausgeführt. | |

## Randbedingungen

### Interrupt-Regeln

1. Für jeden Interrupt, der nicht sofort abgearbeitet werden kann, oder aktuell schon in Bearbeitung ist, wird eine weitere Interrupt-Anforderung gespeichert. Darüber hinaus gehende Interrupt-Anforderungen für diesen Interrupt verloren.
2. Wird aktuell ein Interrupt bearbeitet und es wird ein weiterer Interrupt mit höherer Priorität ausgelöst, unterbricht dieser den niederprioren Interrupt. Nach Abschluss des höherprioren Interrupts, wird der niederpriore Interrupt fortgesetzt. Treffen während der höherpriore Interrupt bearbeitet wird, weiter Anforderungen für den niederprioren Interrupt ein, wird eine Anforderung gespeichert. Weitere gehen verloren.
3. Wird aktuell ein Interrupt bearbeitet und es wird ein weiterer Interrupt mit höherer Priorität ausgelöst, unterbricht dieser den niederprioren Interrupt. Der höherpriore Interrupt wird bearbeitet. Wird wiederum ein höherpriorer Interrupt ausgelöst, wird der aktuelle Interrupt unterbrochen und der höherpriore Interrupt bearbeitet. Maximal sind sechs aktive Interrupt-Ebenen möglich. Eine aktuell bearbeitete Interrupt-Ebene und fünf wartende Interrupt-Ebenen. Für jede aktive Interrupt-Ebene wird maximal eine weitere Interrupt-Anforderung gespeichert. Alle weiteren Interrupt-Anforderungen gehen verloren. Ebenso gehen Interrupt-Anforderung verloren, wenn diese für weitere Interrupt-Ebenen (Interrupt-Ebene ≥ 7) angefordert werden.

## Beispiele

### Beispiel 1: Interruptroutinen zuordnen und Priorität festlegen

| **Programmcode** | **Kommentar** |
|---|---|
| `N20 SETINT(3) PRIO=1 ABHEB_Z` | `; IF Eingang 3 == 1` |
| | `; THEN Interruptroutine "ABHEB_Z" starten` |
| `N30 SETINT(2) PRIO=2 ABHEB_X` | `; IF Eingang 2 == 1` |
| | `; THEN Interruptroutine "ABHEB_X" starten.` |

Die Interruptroutinen werden in der Reihenfolge der Prioritätswerte nacheinander abgearbeitet, wenn die Eingänge gleichzeitig anstehen: zuerst "ABHEB_Z", dann "ABHEB_X".

### Beispiel 2: Interruptroutine neu zuordnen

| **Programmcode** | **Kommentar** |
|---|---|
| `N20 SETINT(3) PRIO=2 ABHEB_Z` | `; IF Eingang 3 == 1` |
| | `; THEN Interruptroutine` **`"ABHEB_Z"`** `starten` |
| `...` | |
| `N80 SETINT(3) PRIO=1 ABHEB_X` | `; IF Eingang 3 == 1` |
| | `; THEN Interruptroutine` **`"ABHEB_X"`** `starten` |

## 3.3.4 Zuordnung einer Interruptroutine deaktivieren/reaktivieren (DISABLE, ENABLE)

Eine SETINT-Anweisung kann mit DISABLE deaktiviert und mit ENABLE wieder aktiviert werden, ohne dass die Zuordnung Eingang → Interruptroutine verloren geht.

### Syntax

```
DISABLE(<n>)
ENABLE(<n>)
```

### Bedeutung

| | | |
|---|---|---|
| DISABLE(<n>): | Befehl: **Deaktivieren** der Interruptroutinen-Zuordnung von Eingang <n> | |
| ENABLE(<n>): | Befehl: **Reaktivieren** der Interruptroutinen-Zuordnung von Eingang <n> | |
| <n>: | Parameter: Nummer des Interruptsignals | |
| | Typ: | INT |
| | Wertebereich: | 1 ... 32 |

### Beispiel

| **Programmcode** | **Kommentar** |
|---|---|
| N20 SETINT(3) PRIO=1 ABHEB_Z | ; Wenn Eingang 3 schaltet, dann soll die |
| | ; Interruptroutine "ABHEB_Z" starten. |
| ... | |
| N90 DISABLE(3) | ; Die SETINT-Anweisung aus N20 wird deaktiviert. |
| ... | |
| N130 ENABLE(3) | ; Die SETINT-Anweisung aus N20 wird wieder aktiviert. |
| ... | |

## 3.3.5 Zuordnung einer Interruptroutine löschen (CLRINT)

Eine mit SETINT definierte Zuordnung eines Interruptsignals zu einem NC-Programm (ASUP) kann mit CLRINT gelöscht werden.

### Syntax

```
CLRINT(<n>)
```

### Bedeutung

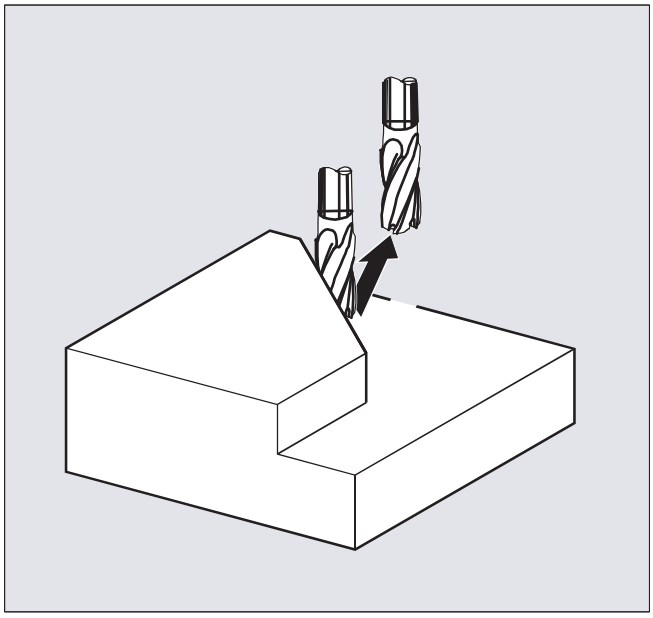
| | | |
|---|---|---|
| CLRINT(<n>): | Befehl: Löschen der Zuordnung des Interruptsignals <n> zum mit SETINT definierten NC-Programm (ASUP) <n> | |
| <n>: | Parameter: Nummer des Interruptsignals | |
| | Typ: | INT |
| | Wertebereich: | 1 ... 32 |

### Beispiel

| Programmcode | Kommentar |
|---|---|
| N20 SETINT(3) PRIO=2 ABHEB_Z | |
| ... | |
| N50 CLRINT(3) | ; Die Zuordnung zwischen Eingang "3" und der Interruptroutine "ABHEB_Z" ist gelöscht. |

## 3.3.6 Schnellabheben von der Kontur (SETINT LIFTFAST, ALF)

Bei einer SETINT-Anweisung mit LIFTFAST wird beim Schalten des Eingangs das Werkzeug durch schnelles Abheben von der Werkstückkontur weggefahren.



Der weitere Ablauf ist davon abhängig, ob die SETINT-Anweisung neben LIFTFAST eine Interruptroutine enthält:

| | |
|---|---|
| Mit Interruptroutine: | **Nach** dem Schnellabheben wird die Interruptroutine ausgeführt. |
| Ohne Interruptroutine: | Die Bearbeitung wird nach dem Schnellabheben mit Alarm gestoppt. |

## Syntax

```
SETINT(<n>) PRIO=1 LIFTFAST
SETINT(<n>) PRIO=1 <NAME> LIFTFAST
```

## Bedeutung

| | | |
|---|---|---|
| SETINT(<n>): | Befehl: Eingang <n> einer Interruptroutine zuordnen. Die zugeordnete Interruptroutine startet, wenn Eingang <n> schaltet. | |
| <n>: | Parameter: Nummer des Eingangs | |
| | Typ: | INT |
| | Wertebereich: | 1 ... 8 |
| PRIO=: | Festlegung der Priorität | |
| <Wert>: | Prioritätswert | |
| | Wertebereich: | 1 ... 128 |
| | Priorität 1 entspricht der höchsten Priorität. | |
| <NAME>: | Name des Unterprogramms (Interruptroutine), das abgearbeitet werden soll. | |
| LIFTFAST: | Befehl: Schnellabheben von der Kontur | |
| ALF=…: | Befehl: Programmierbare Verfahrrichtung (steht im Bewegungssatz)<br>Zu den Programmiermöglichkeiten mit ALF siehe Thema "Verfahrrichtung beim Schnellabheben von der Kontur (Seite 554)". | |

## Randbedingungen

### Verhalten bei aktivem Frame mit Spiegelung

Bei der Bestimmung der Abheberichtung wird geprüft, ob ein Frame mit Spiegelung aktiv ist. In diesem Fall werden bei der Abheberichtung bezogen auf die Tangentenrichtung rechts und links vertauscht. Die Richtungsanteile in Werkzeugrichtung werden nicht gespiegelt. Aktiviert wird dieses Verhalten durch die MD-Einstellung:

MD21202 $MC_LIFTFAST_WITH_MIRROR = TRUE

## Beispiel

Ein abgebrochenes Werkzeug soll automatisch durch ein Schwesterwerkzeug ersetzt werden. Die Bearbeitung wird dann mit dem neuen Werkzeug fortgesetzt.

### Hauptprogramm:

| **Hauptprogramm** | **Kommentar** |
|---|---|
| N10 SETINT(1) PRIO=1 W_WECHS LIFTFAST | ; Wenn Eingang 1 schaltet, wird sofort das Werkzeug mit Schnellabheben (Code Nr. 7 für Werkzeugradiuskorrektur G41) von der Kontur weggefahren. Dann wird die Interruptroutine "W_WECHS" abgearbeitet. |
| N20 G0 Z100 G17 T1 ALF=7 D1 | |
| N30 G0 X-5 Y-22 Z2 M3 S300 | |
| N40 Z-7 | |

| **Hauptprogramm** | **Kommentar** |
|---|---|
| N50 G41 G1 X16 Y16 F200 | |
| N60 Y35 | |
| N70 X53 Y65 | |
| N90 X71.5 Y16 | |
| N100 X16 | |
| N110 G40 G0 Z100 M30 | |

**Unterprogramm:**

| **Unterprogramm** | **Kommentar** |
|---|---|
| PROC W_WECHS SAVE | ; Unterprogramm mit Speicherung des aktuellen Betriebszustandes |
| N10 G0 Z100 M5 | ; Werkzeugwechselposition, Spindelstopp |
| N20 T11 M6 D1 G41 | ; Werkzeug wechseln |
| N30 REPOSL RMBBL M3 | ; Kontur wiederanfahren und Rücksprung ins Hauptprogramm (wird in einem Satz programmiert) |

## 3.3.7 Verfahrrichtung beim Schnellabheben von der Kontur

### Rückzugsbewegung

Die Ebene der Rückzugsbewegung wird durch folgende G-Befehle bestimmt:

- LFTXT
  Die Ebene der Rückzugsbewegung wird aus der Bahntangente und der Werkzeugrichtung bestimmt (Standardeinstellung).
- LFWP
  Die Ebene der Rückzugsbewegung ist die aktive Arbeitsebene, die mit den G-Befehlen G17, G18 oder G19 ausgewählt wird. Die Richtung der Rückzugsbewegung ist unabhängig von der Bahntangente. Damit ist ein achsparalleles Schnellabheben programmierbar.
- LFPOS
  Rückzug der mit POLFMASK / POLFMLIN bekannt gemachten Achse auf die mit POLF programmierte absolute Achsposition.
  ALF hat keinen Einfluss auf die Abheberichtung für mehrere Achsen sowie für mehrere Achsen im linearen Zusammenhang.

### Programmierbare Verfahrrichtung (ALF=…)

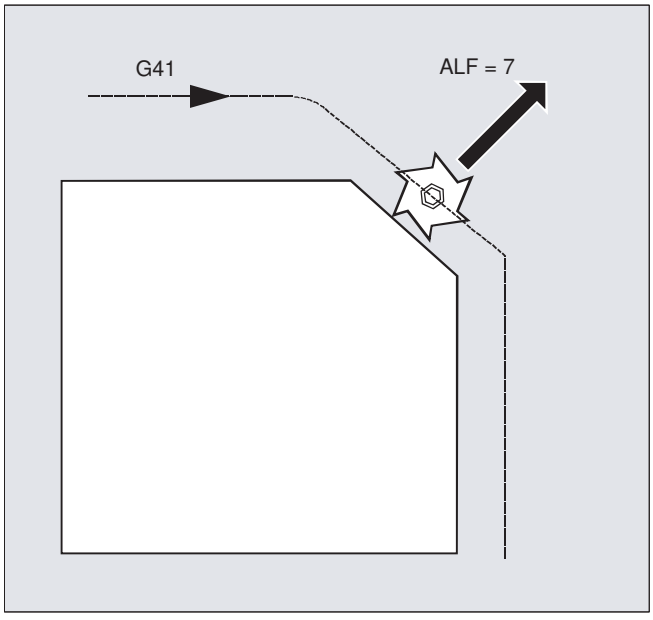
In der Ebene der Rückzugsbewegung wird mit ALF die Richtung in diskreten Schritten von 45 Grad programmiert.

Die möglichen Verfahrrichtungen sind in der Steuerung unter speziellen Code-Nummern gespeichert und unter dieser Nummer abrufbar.

Beispiel:

```
Programmcode
N10 SETINT(2) PRIO=1 ABHEB_Z LIFTFAST
ALF=7
```

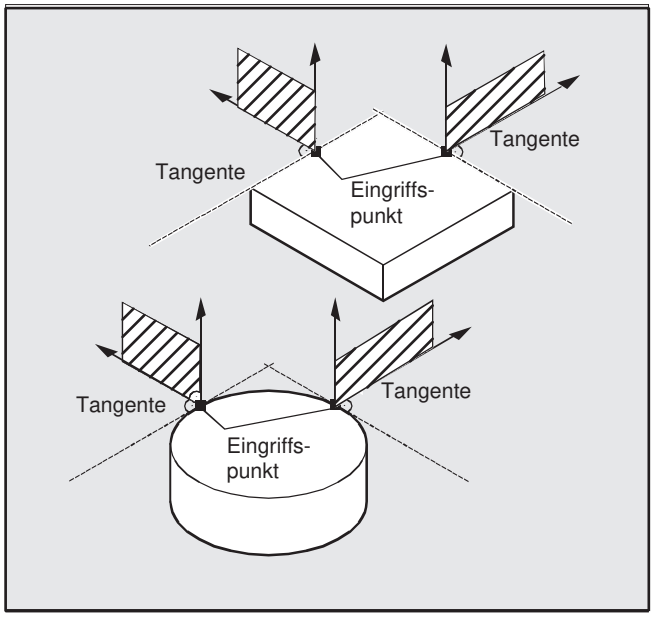
Das Werkzeug fährt bei eingeschaltetem `G41` (Bearbeitungsrichtung links von der Kontur) senkrecht von der Kontur weg.




### Bezugsebene für die Beschreibung der Verfahrrichtungen bei LFTXT

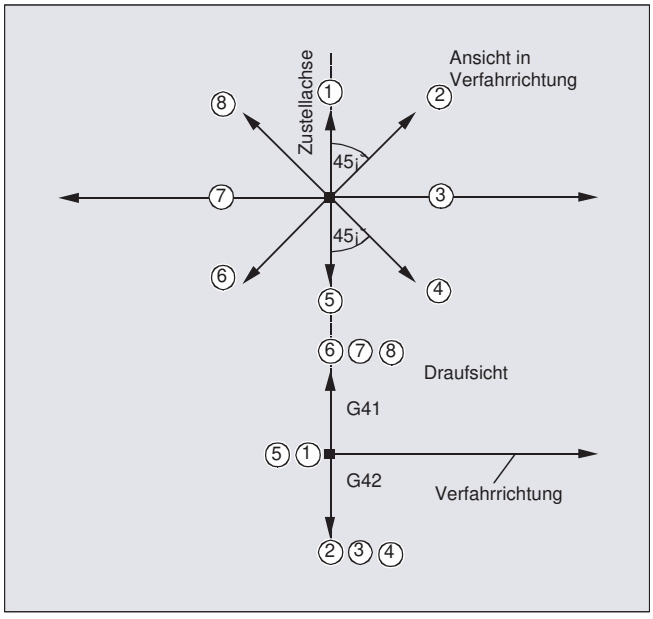
Im Eingriffspunkt des Werkzeugs an der programmierten Kontur wird eine Ebene aufgespannt, die als Bezug für die Angabe der Abhebebewegung mit der entsprechenden Code-Nummer dient.

Die Bezugsebene wird aufgespannt aus der Werkzeuglängsachse (Zustellrichtung) und einem Vektor, der zu dieser und senkrecht zur Tangente im Eingriffspunkt des Werkzeugs an der Kontur steht.




### Code-Nummern mit Verfahrrichtungen bei LFTXT

Ausgehend von der Bezugsebene finden Sie in folgender Abbildung die Code-Nummern mit Verfahrrichtungen.




Für `ALF=1` ist der Rückzug in Werkzeugrichtung festgelegt.

Mit `ALF=0` ist die Funktion "Schnellabheben" ausgeschaltet.

**⚠ VORSICHT**

**Kollisionsgefahr**

Bei eingeschalteter Werkzeugradiuskorrektur sollten:

- bei `G41` die Codierungen 2, 3, 4
- bei `G42` die Codierungen 6, 7, 8

**nicht** verwendet werden, da in diesen Fällen das Werkzeug zur Kontur hinfahren und mit dem Werkstück kollidieren würde.

**Code-Nummern mit Verfahrrichtungen bei LFWP**

Bei `LFWP` ergibt sich die Richtung in der Arbeitsebene nach folgender Zuordnung:

- `G17`: X/Y-Ebene
  `ALF=1`: Rückzug in X-Richtung
  `ALF=3`: Rückzug in Y-Richtung
- `G18`: Z/X-Ebene
  `ALF=1`: Rückzug in Z-Richtung
  `ALF=3`: Rückzug in X-Richtung
- `G19`: Y/Z-Ebene
  `ALF=1`: Rückzug in Y-Richtung
  `ALF=3`: Rückzug in Z-Richtung

### Siehe auch

Schnellrückzug während Gewindeschneiden (LFON, LFOF, DILF, ALF, LFTXT, LFWP, LFPOS, POLF, POLFMASK, POLFMLIN) (Seite 234)

## 3.3.8 Bewegungsablauf bei Interruptroutinen

### Interruptroutine ohne LIFTFAST

Die Achsbewegungen werden auf der Bahn bis zum Stillstand abgebremst. Anschließend startet die Interruptroutine.

Die Stillstandsposition wird als Unterbrechungsposition abgespeichert und wird bei `REPOS` mit `RMIBL` am Ende der Interruptroutine angefahren.

### Interruptroutine mit LIFTFAST

Die Achsbewegungen werden auf der Bahn abgebremst. Gleichzeitig wird die `LIFTFAST`-Bewegung als überlagerte Bewegung ausgeführt. Wenn die Bahnbewegung und `LIFTFAST`-Bewegung zum Stillstand gekommen sind, wird die Interruptroutine gestartet.

Als Unterbrechungsposition wird die Position auf der Kontur abgespeichert, bei der die `LIFTFAST`-Bewegung gestartet und dadurch die Bahn verlassen wurde.

Die Interruptroutine verhält sich mit `LIFTFAST` und `ALF=0` identisch wie die Interruptroutine ohne `LIFTFAST`.

**Hinweis**

Der Betrag, um den die Geometrieachsen beim Schnellabheben von der Kontur wegfahren, ist über ein Maschinendatum einstellbar.

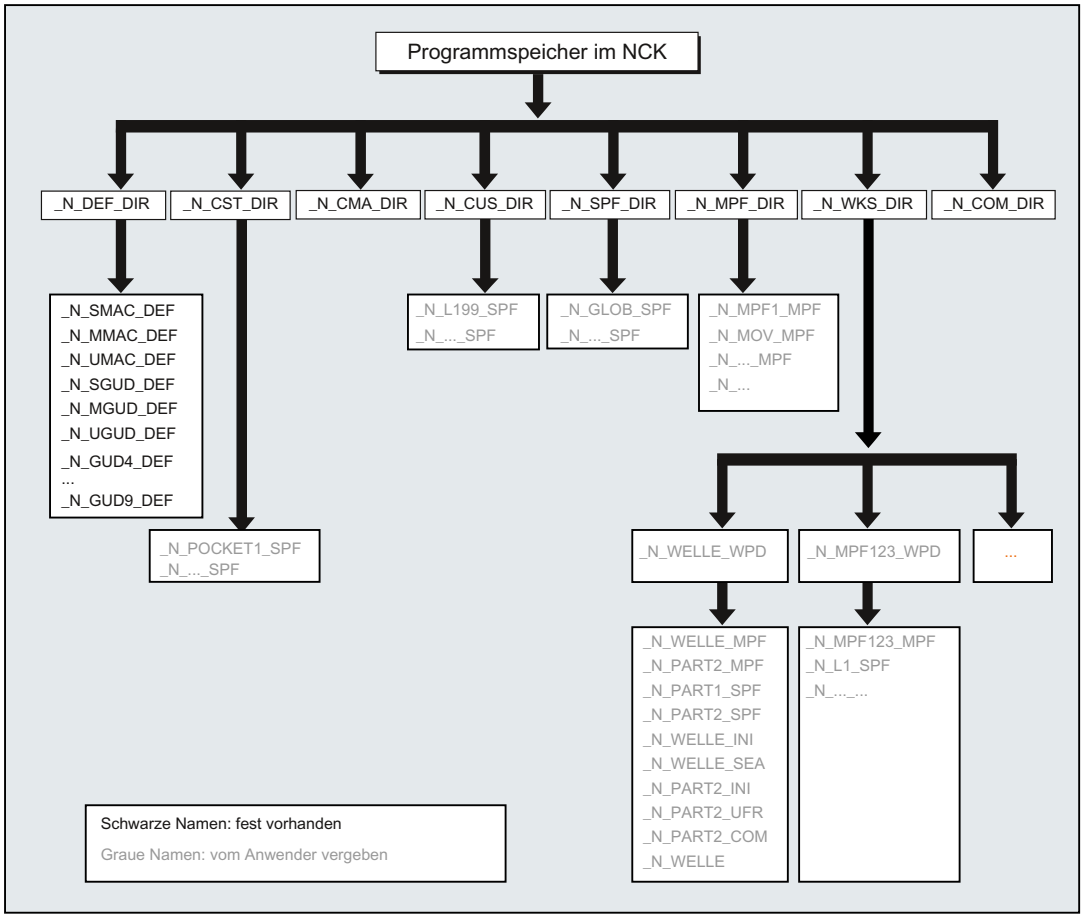
# 3.4 Datei- und Programmverwaltung

## 3.4.1 Programmspeicher

### 3.4.1.1 Programmspeicher im NCK

Im Programmspeicher werden Dateien und Programme (z. B. Haupt- und Unterprogramme, Makro-Definitionen) persistent gespeichert (→ Passives Filesystem).

Daneben gibt es eine Anzahl von Dateitypen, die hier zwischengespeichert werden können und bei Bedarf (z. B. bei Bearbeitung eines bestimmten Werkstücks) in den Arbeitsspeicher zu übertragen sind (z. B. für Initialisierungszwecke).



## Standard-Verzeichnisse

Folgende Verzeichnisse sind standardmäßig vorhanden:

| Verzeichnis | Inhalt |
|---|---|
| _N_DEF_DIR | Datenbausteine und Makrobausteine |
| _N_CST_DIR | Standard-Zyklen |
| _N_CMA_DIR | Hersteller-Zyklen |
| _N_CUS_DIR | Anwender-Zyklen |
| _N_WKS_DIR | Werkstücke |
| _N_SPF_DIR | Globale Unterprogramme |
| _N_MPF_DIR | Hauptprogramme |
| _N_COM_DIR | Kommentare |

## Dateitypen

Im Programmspeicher können folgende Dateitypen eingebracht werden:

| Dateityp | Beschreibung |
|---|---|
| <Name>_MPF | Hauptprogramm |
| <Name>_SPF | Unterprogramm |
| <Name>_TEA | Maschinendaten |
| <Name>_SEA | Settingdaten |
| <Name>_TOA | Werkzeugkorrekturen |
| <Name>_UFR | Nullpunktverschiebungen/Frame |
| <Name>_INI | Initialisierungsdatei |
| <Name>_GUD | Globale Anwenderdaten |
| <Name>_RPA | R-Parameter |
| <Name>_COM | Kommentar |
| <Name>_DEF | Definitionen für globale Anwenderdaten und Makros |

## Werkstück-Hauptverzeichnis (_N_WKS_DIR)

Das Werkstück-Hauptverzeichnis ist standardmäßig unter der Bezeichnung _N_WKS_DIR im Programmspeicher eingerichtet. Das Werkstück-Hauptverzeichnis enthält für alle Werkstücke, die Sie programmiert haben, die entsprechenden Werkstückverzeichnisse.

## Werkstückverzeichnisse ( ..._WPD)

Ein Werkstückverzeichnis enthält alle Dateien, die zum Bearbeiten eines Werkstücks notwendig sind. Dies können Hauptprogramme, Unterprogramme, beliebige Initialisierungs-Programme und Kommentar-Dateien sein.

Initialisierungsprogramme werden nach der Programmanwahl mit dem ersten Teileprogrammstart einmalig ausgeführt (entsprechend Maschinendatum MD11280 $MN_WPD_INI_MODE).

**Beispiel:**

Das Werkstückverzeichnis _N_WELLE_WPD, das für das Werkstück WELLE angelegt wurde, enthält folgende Dateien:

| Datei | Beschreibung |
|---|---|
| _N_WELLE_MPF | Hauptprogramm |
| _N_PART2_MPF | Hauptprogramm |
| _N_PART1_SPF | Unterprogramm |
| _N_PART2_SPF | Unterprogramm |
| _N_WELLE_INI | Allgemeines Initialisierungsprogramm der Daten für das Werkstück |
| _N_WELLE_SEA | Initialisierungsprogramm Settingdaten |
| _N_PART2_INI | Allgemeines Initialisierungsprogramm der Daten für Programm Part 2 |
| _N_PART2_UFR | Initialisierungsprogramm für Frame-Daten für Programm Part 2 |
| _N_WELLE_COM | Kommentardatei |

Zusätzlich können in einem Werkstückverzeichnis auch Daten abgelegt werden, welche nicht unmittelbar für die Bearbeitung durch den NC benötigt werden. Dies können neben ASCII-Dateien auch Binärdateien wie z. B. Bilder im JPG-Format oder Beschreibungen im PDF-Format sein. Damit diese vom NC als Binärdateien interpretiert werden können, müssen die Dateierweiterungen im NC bekannt sein (Einstellung bei der Inbetriebnahme über MD17000 $MN_ EXTENSIONS_OF_BIN_FILES; in der Grundeinstellung sind folgende Dateierweiterungen vorbelegt: JPG, GIF, PNG, BMP, PDF, ICO, HTM).

### Werkstück für die Bearbeitung anwählen

Ein Werkstückverzeichnis kann für die Abarbeitung in einem Kanal angewählt werden. Befindet sich in diesem Verzeichnis ein Hauptprogramm **gleichen Namens** oder nur ein einziges Hauptprogramm (_MPF), so wird dieses automatisch für die Abarbeitung angewählt.

## 3.4.1.2 Externe Programmspeicher

Neben dem passiven Filesystem im NC können an einer Maschine auch externe Programmspeicher verfügbar sein (z. B auf dem lokalen Laufwerk oder auf einem Netzlaufwerk).

Über die Funktionen "Abarbeiten von Extern" oder "EES (Execution from External Storage)" können Teilprogramme **direkt** von externen Programmspeichern abgearbeitet werden.

**Weitere Informationen:** Funktionshandbuch Basisfunktionen

### Globaler Teileprogrammspeicher (GDIR)

Bei der Vereinbarung der Laufwerke kann eines der Laufwerke als globaler Teileprogrammspeicher (GDIR) ausgezeichnet werden.

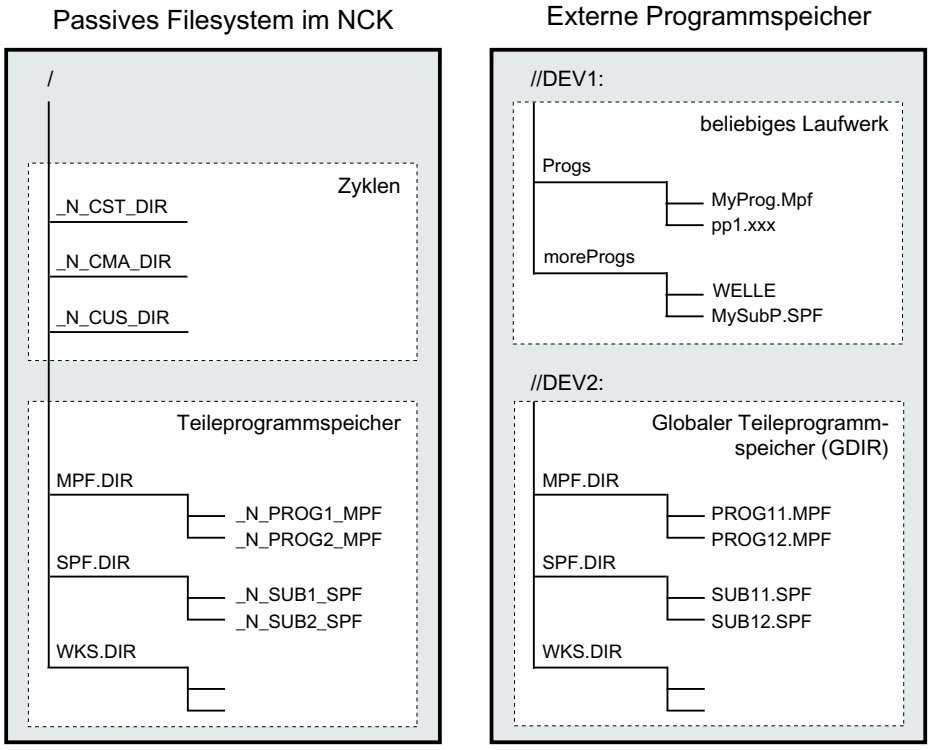
Vom System werden dann automatisch die Verzeichnisse MPF.DIR, SPF.DIR und WKS.DIR auf dem Laufwerk angelegt. Diese drei Verzeichnisse bilden das GDIR.

Das GDIR spielt ausschließlich für die Funktion EES eine Rolle. Je nach Laufwerkskonfiguration ersetzt oder erweitert das GDIR den NC-Teileprogrammspeicher. Das Einrichten eines GDIR ist für den EES-Betrieb allerdings nicht zwingend erforderlich.

Die Verzeichnisse und Dateien des GDIR können im Teileprogramm auf dieselbe Art wie im passiven Filesystem adressiert werden. Damit ist eine kompatible Verlagerung eines NC-Programms mit Pfadangaben aus dem passiven Filessystem ins GDIR möglich. Das Verzeichnis SPF.DIR des GDIR ist im Suchpfad für Unterprogramme enthalten.

## Programmorganisation

Die folgende Abbildung soll die Programmorganisation auf externen Programmspeichern veranschaulichen:




## Case-insensitive Dateisysteme

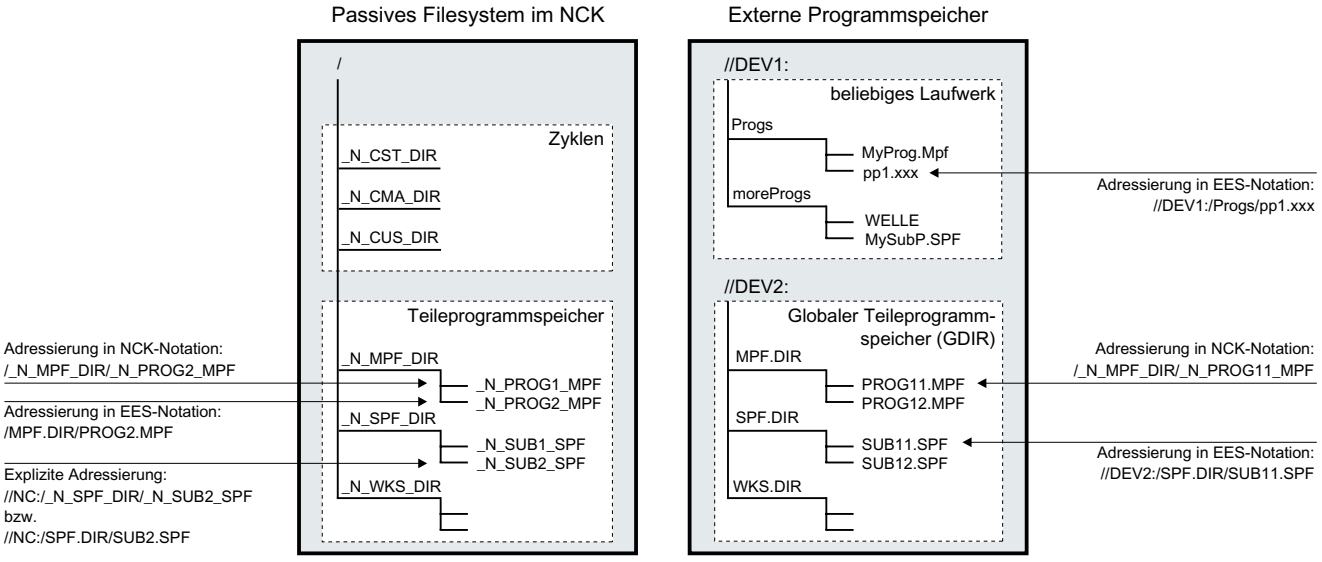
**Hinweis**

Um Probleme durch die Groß- und Kleinschreibung bei der Datei-Adressierung (siehe "Adressierung von Dateien des Programmspeichers (Seite 563)") zu vermeiden, sollten **case-insensitive** Dateisysteme als externe Programmspeicher verwendet werden.

### 3.4.1.3 Adressierung von Dateien des Programmspeichers

Eine Datei im Programmspeicher, die durch einen Dateihandlingsbefehl (z. B. WRITE, DELETE, READ, ISFILE, FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO) angesprochen wird, wird entweder durch einen absoluten Pfad plus Dateinamen oder nur durch den Dateinamen allein referenziert. Im zweiten Fall wird der Pfad des angewählten Programms als Dateipfad verwendet.

## Adressierung in NC-/EES-Notation




#### Adressierung von Dateien des passiven Filesystems

Die Adressierung von Dateien des passiven Filesystems erfolgt üblicherweise in **NC-Notation** (Verzeichnis- und Dateinamen beginnen mit der Domainkennung "_N_", Trennzeichen für die Verzeichnis-/Dateierweiterung ist "_") ohne Angabe des Laufwerknamens. Eine Adressierung in **EES-Notation** (ohne Domain-Kennung "_N_", Trennzeichen für die Verzeichnis-/Dateierweiterung ist ".") ist aber auch zulässig.

Beispiel:

- NC-Notation: "/_N_SPF_DIR/_N_SUB1_SPF"
- EES-Notation: "/SPF.DIR/SUB1.SPF"

#### Hinweis

Adressierungen von Dateien des passiven Filesystems in EES-Notation werden intern nach folgenden Regeln in NC-Notation umgewandelt:

- Verzeichnis- und Dateinamen werden mit der Domainkennung "_N_" erweitert.
- Ist das viertletzte Zeichen im Verzeichnis- bzw. Dateinamen ein Punkt ("."), wird es in einen Unterstrich ("_") umgewandelt.

Über den vordefinierten Laufwerksnamen "//NC:" kann das passive Filesystem auch gezielt adressiert werden.

Beispiel:

- NC-Notation: "//NC:/_N_SPF_DIR/_N_SUB1_SPF"
- EES-Notation: "//NC:/SPF.DIR/SUB1.SPF"

#### Adressierung von Dateien eines externen Programmspeichers

Die Adressierung von Dateien eines externen Programmspeichers, der nicht als GDIR ausgezeichnet ist, muss in EES-Notation erfolgen. Am Anfang des Adressierungspfades muss der Laufwerksname (z. B. "//DEV1:") angegeben werden. Es sind alle in /user/sinumerik/hmi/cfg/logdrive.ini projektierten symbolischen Gerätenamen zulässig.

Beispiel:

- EES-Notation: "//DEV1:/MyProgDir/pp1.xxx"
- NC-Notation: nicht zulässig

#### Adressierung von Dateien des globalen Teileprogrammspeichers (GDIR)

Bei der Adressierung von Dateien des GDIR ist neben der Pfadangabe in EES-Notation auch eine Pfadangabe in NC-Notation zulässig.

Beispiel:

- EES-Notation: "//DEV2:/MPF.DIR/PROG11.MPF"
- NC-Notation: "/_N_MPF_DIR/_N_PROG11_MPF"

---

**Hinweis**

Adressierungen von Dateien des GDIR in NC-Notation werden intern nach folgenden Regeln in EES-Notation umgewandelt:

- Die Domainkennung "_N_" in Verzeichnis- und Dateinamen wird entfernt.
- Ist das viertletzte Zeichen im Verzeichnis- bzw. Dateinamen ein Unterstrich ("_"), wird es in einen Punkt (".") umgewandelt.

---

## Regeln zur Pfadangabe

Eine vollständige Pfadangabe besteht aus Laufwerksnamen, Verzeichnispfad und Dateinamen.

#### Laufwerksname

Für die Angabe des Laufwerksnamens gelten folgende Regeln:

- Es sind alle in /user/sinumerik/hmi/cfg/logdrive.ini projektierten symbolischen Gerätenamen zulässig.
- Am Anfang steht das Zeichen "//", gefolgt von mindestens einem Buchstaben oder einer Ziffer.
- Die folgenden Zeichen können in beliebiger Kombination Buchstaben, Ziffern, "_" und Leerzeichen sein.

- Abgeschlossen wird der Name mit einem Buchstaben oder einer Ziffer, gefolgt von einem ":".
- Weitere Sonderzeichen sind nicht erlaubt.

**Hinweis**

Für das passive Filesystem ist der Laufwerksname "//NC:" vordefiniert.

Beispiele:

- Externe Programmspeicher:
  - //Drive1:
  - //Drive_1:
  - //Drive 1:
  - //A B:
  - //1 B C 2:

**Verzeichnispfad**

Für die Angabe des Verzeichnispfades gelten folgende Regeln:

- Am Anfang und am Ende des Verzeichnispfades und als Trennzeichen für die einzelnen Pfadanteile steht "/".

  **Hinweis**

  Ein doppelter Schrägstrich ("//") innerhalb des Verzeichnispfades ist **nicht** zulässig!

- Verzeichnisnamen:
  - Verzeichnisnamen müssen mit einem Buchstaben oder einer Ziffer beginnen. Nur bei der Adressierung in NC-Notation beginnen die Verzeichnisnamen mit der Domainkennung "_N_".
  - Die folgenden Zeichen können in beliebiger Kombination Buchstaben, Ziffern und "_" sein.

    **Hinweis**

    Bei externen Programmspeichern sind auch Leerzeichen im Verzeichnisnamen zulässig. Dies gilt jedoch nicht, wenn der externe Programmspeicher als globaler Teileprogrammspeicher (GDIR) eingerichtet ist.

  - Weitere Sonderzeichen sind nicht erlaubt.
- Verzeichniserweiterungen:
  - Verzeichniserweiterungen müssen aus genau drei Buchstaben/Ziffern bestehen.
  - Sie werden mit "_" (NC-Notation) bzw. "." (EES-Notation) vom Verzeichnisnamen getrennt.

  **Hinweis**

  Im passiven Filesystem gibt es nur die Verzeichniserweiterungen _DIR und _WPD.

Beispiele:

- Passives Filesystem bzw. GDIR:
  - NC-Notation: _N_WKS_DIR/_N_MYNCPROGS_WPD/...
  - EES-Notation: WKS.DIR/MYPROGS.WPD/...
- Externe Programmspeicher:
  - /abc
  - /ab_c.def
  - /ab c1.def
  - /a b c .d11
  - /abc.def/ghi.klm

**Dateiname**

Für Dateinamen gelten folgende Regeln:

- Nur bei der Adressierung in NC-Notation beginnen die Dateinamen mit der Domainkennung "_N_".
- Die beiden folgenden Zeichen sollten zwei Buchstaben oder ein Unterstrich gefolgt von einem Buchstaben sein.

  **Hinweis**

  Wenn diese Bedingung erfüllt ist, dann kann ein NC-Programm allein durch Angabe des Programmnamens als Unterprogramm aus einem anderen Programm heraus aufgerufen werden. Beginnt der Programmname hingegen mit Ziffern, dann ist der Unterprogramm-Aufruf nur über die CALL-Anweisung möglich.

- Die folgenden Zeichen können in beliebiger Kombination Buchstaben, Ziffern und "_" sein.
- Dateierweiterung:
  - Die Dateierweiterung muss aus genau drei Buchstaben/Ziffern bestehen.

    **Hinweis**

    Zulässige Dateierweiterungen im passiven Filesystem siehe "Programmspeicher im NCK (Seite 559)".

  - Sie wird mit "_" (NC-Notation) bzw. "." (EES-Notation) vom Dateinamen getrennt.

Beispiele:

- Passives Filesystem bzw. GDIR:
  - NC-Notation: _N_SUB1_SPF
  - EES-Notation: SUB1.SPF
- Externe Programmspeicher:
  - Teil1
  - _Teil1
  - Teil_1.spf
  - Teil1.mpf

**DIN Unterprogrammname**

Für DIN Unterprogrammnamen gelten folgende Regeln:

- Das erste Zeichen ist der Buchstabe "L".
- Die folgenden Zeichen sind Ziffern (mindestens eine).
- Dateierweiterung:
  - Die Dateierweiterung muss aus genau drei Buchstaben bestehen.
  - Sie wird mit "_" (NC-Notation) bzw. "." (EES-Notation) vom Dateinamen getrennt.

Beispiele:

- L123
- L1_SPF (NC-Notation) bzw. L1.SPF (EES-Notation)

**Maximale Pfadlänge**

Für die Angabe von Laufwerksname und Verzeichnispfad stehen maximal 128 Bytes zur Verfügung, die Länge des Dateinamens darf maximal 31 Bytes betragen. Für den gesamten Pfad ergibt das eine maximale Länge von 159 Bytes.

### 3.4.1.4 Suchpfad bei Unterprogrammaufruf

Bei Unterprogrammaufrufen ohne Pfadangabe wird der absolute Pfad durch Abarbeiten eines festen Suchpfades ermittelt.

Die Programmspeicher werden dabei in der folgenden Reihenfolge durchsucht:

| | | Verzeichnis | Beschreibung |
|---|---|---|---|
| 1 | | aktuelles Verzeichnis / *name* | Das aktuelle Verzeichnis ist das Verzeichnis, in dem die Programmanwahl erfolgt ist.<br>Dies kann sein:<br>• ein Werkstückverzeichnis oder das Standard-Verzeichnis _N_MPF_DIR im NC-Teileprogrammspeicher bzw. globalen Teileprogrammspeicher<br>oder<br>• ein beliebiges Verzeichnis eines externen Programmspeichers |
| 2 | | aktuelles Verzeichnis / *name_SPF* | |
| 3 | | aktuelles Verzeichnis / *name_MPF* | |
| 4 | a | //NC:/_N_SPF_DIR / *name_SPF* | Unterprogramm-Verzeichnis im NC-Teileprogrammspeicher |
| | b | //DEV2:/_N_SPF_DIR / name_SPF [1] | Unterprogramm-Verzeichnis im globalen Teileprogrammspeicher<br>**Hinweis:**<br>Dieser Suchschritt entfällt, wenn kein globaler Teileprogrammspeicher eingerichtet ist oder die Programmanwahl im NC-Teileprogrammspeicher erfolgt ist. |
| 5 | | Mit `CALLPATH` programmierte Suchpfaderweiterung (siehe "Suchpfad bei Unterprogrammaufrufen erweitern (CALLPATH) (Seite 541)").<br>**Hinweis:**<br>Dieser Suchschritt entfällt, wenn kein `CALLPATH` programmiert ist. | |
| 6 | | /_N_CUS_DIR / *name_SPF* | Anwenderzyklen-Verzeichnis |
| 7 | | /_N_CMA_DIR / *name_SPF* | Herstellerzyklen-Verzeichnis |
| 8 | | /_N_CST_DIR / *name_SPF* | Standardzyklen-Verzeichnis |

[1] //DEV2:" steht beispielhaft für das Laufwerk, auf dem der globale Teileprogrammspeicher eingerichtet ist.

Für die Suche gelten folgende Regeln:

- Der Suchpfad wird für jeden einzelnen Unterprogrammaufruf durchlaufen, d. h. es ist irrelevant, wo sich das übergeordnete Programm befindet.
- Je nach Verzeichnis werden unterschiedliche Dateitypen berücksichtigt.
- Grundsätzlich wird in einem Verzeichnis nicht in unterlagerten, d. h. geschachtelten Verzeichnissen gesucht.

### 3.4.1.5 Abfrage von Pfad und Dateiname

Zur Abfrage des Pfades und des Dateinamens eines NC-Programms stehen folgende im Teileprogramm lesbare Systemvariablen zur Verfügung:

| Systemvariable | Typ | Bedeutung |
|---|---|---|
| $P_STACK | INT | Liefert die Programmebene, in der das aktuelle NC-Programm bearbeitet wird. |
| $P_PATH[ <n>] | STRING | Liefert den Pfad des NC-Programms, welches in der durch den Feldindex <n> selektierten Programmebene bearbeitet wird.<br>Beispiele:<br>$P_PATH[0] liefert den Pfad für das Hauptprogramm, z. B. "/_N_WKS_DIR/_N_WELLE_WPD/".<br>$P_PATH[$P_STACK - 1] liefert den Pfad des aufrufenden Programms.<br>Bezieht sich der Pfad auf ein NC-Programm, das im passiven Filesystem des NC oder im globalen Teileprogrammspeicher (GDIR) abgelegt ist, wird der Pfad in NC-Notation geliefert.<br>Bezieht sich der Pfad auf ein NC-Programm, das von einem anderen externen Programmspeicher als dem globalen Teileprogrammspeicher abgearbeitet wird, liefert $P_PATH den Pfad in EES-Notation. |
| $P_PROG[ <n>] | STRING | Liefert den Namen des NC-Programms, welches in der durch den Feldindex <n> selektierten Programmebene bearbeitet wird.<br>Ist das NC-Programm im passiven Filesystem des NC oder im globalen Teileprogrammspeicher abgelegt, wird der Programmname in NC-Notation geliefert.<br>Wird das NC-Programm von einem anderen externen Laufwerk als dem globalen Teileprogrammspeicher abgearbeitet, liefert $P_PROG den Namen in EES-Notation. |
| $P_PROGPATH | STRING | Liefert den Pfad des NC-Programms, das gerade bearbeitet wird.<br>Der Aufruf von $P_PROGPATH ist identisch zu $P_PATH[$P_STACK]. |

| Systemvariable | Typ | Bedeutung | |
|---|---|---|---|
| $P_IS_EES_PATH[ <n>] | BOOL | Abfrage, ob der von $P_PATH[<n>] gelieferte Pfad bzw. der von $P_PROG[<n>] gelieferte Programmname der NC-Notation oder der EES-Notation entspricht. | |
| | | = FALSE | $P_PATH[<n>] und $P_PROG[<n>] liefern NC-Notation. D. h. jedem Bezeichner ist eine Präfix "_N_" vorangestellt. Das Trennzeichen für die Dateikennung ist "_".<br>Beispiele:<br>• Pfad in NC-Notation: "/_N_WKS_DIR/_N_MYWPD_WPD/"<br>• Programmname in NC-Notation:"_N_MYPROG_MPF"<br>Ein Pfad in NC-Notation kann sich sowohl auf das passive Filesystem im NC als auch auf den globalen Teileprogrammspeicher beziehen. |
| | | = TRUE | $P_PATH[<n>] und $P_PROG[<n>] liefern EES-Notation. D. h. den Bezeichnern ist kein Präfix "_N_" vorangestellt. Das Trennzeichen für die Dateikennung ist ".".<br>Beispiele:<br>• Pfad in EES-Notation: "//DEV1:/WKS.DIR/MYWPD.WPD/"<br>• Programmname in EES-Notation: "MYPROG.MPF" |

<n>: Der Index <n> definiert die Programmebene, aus der die Pfadinformation gelesen werden soll (Wertebereich: 0 ... 17)

**Hinweis**

Im EES-Betrieb werden außerhalb des globalen Teileprogrammspeichers (GDIR) von den Systemvariablen $P_PROG, $P_PATH, und $P_PROGPATH Pfadnamen in EES-Notation geliefert. Anwenderprogramme, die diese Pfadnamen auswerten und weiterverarbeiten, müssen daher für den EES-Betrieb so erweitert werden, dass sie auch Pfadnamen in EES-Notation verarbeiten können.

## 3.4.2 Arbeitsspeicher (CHANDATA, COMPLETE, INITIAL)

### Funktion

Der Arbeitsspeicher enthält die aktuellen System- und Anwenderdaten, mit denen die Steuerung betrieben wird (aktives Filesystem), z. B.:

- Aktive Maschinendaten
- Werkzeugkorrekturdaten

- Nullpunktverschiebungen
- ...

## Initialisierungsprogramme

Hierbei handelt es sich um Programme, mit denen die Daten des Arbeitsspeichers vorbesetzt (initialisiert) werden. Hierfür können folgende Dateitypen verwendet werden:

| Dateityp | Beschreibung |
|---|---|
| name_TEA | Maschinendaten |
| name_SEA | Settingdaten |
| name_TOA | Werkzeugkorrekturen |
| name_UFR | Nullpunktverschiebungen/Frame |
| name_INI | Initialisierungsdatei |
| name_GUD | Globale Anwenderdaten |
| name_RPA | R-Parameter |

## Datenbereiche

Die Daten können in unterschiedliche Bereiche eingegliedert werden, in denen sie gelten sollen. Beispielsweise kann eine Steuerung über mehrere Kanäle verfügen oder gewöhnlich auch über mehrere Achsen.

Es gibt:

| Kennung | Datenbereiche |
|---|---|
| NC | NC-spezifische Daten |
| CH<n> | Kanalspezifische Daten (<n> gibt die Kanalnummer an) |
| AX<n> | Achsspezifische Daten (<n> gibt die Nummer der Maschinenachse an) |
| TO | Werkzeugdaten |
| COMPLETE | Alle Daten |

## Initialisierungsprogramm am externen PC erzeugen

Mit Hilfe von Datenbereichskennung und Datentypenkennung können die Bereiche bestimmt werden, die bei der Datensicherung als Einheit betrachtet werden:

| | |
|---|---|
| _N_AX5_TEA_INI | Maschinendaten für Achse 5 |
| _N_CH2_UFR_INI | Frames des Kanals 2 |
| _N_COMPLETE_TEA_INI | Alle Maschinendaten |

Nach Inbetriebnahme der Steuerung ist ein Datensatz im Arbeitsspeicher vorhanden, der den ordnungsgemäßen Betrieb der Steuerung gewährleistet.

## Vorgehensweise bei mehrkanaligen Steuerungen (CHANDATA)

`CHANDATA(<Kanalnummer>)` für mehrere Kanäle ist nur in der Datei _N_INITIAL_INI zulässig. Das ist die Inbetriebnahmedatei, mit der alle Daten der Steuerung initialisiert werden.

| **Programmcode** | **Kommentar** |
|---|---|
| `%_N_INITIAL_INI` | |
| `CHANDATA(1)` | |
| | `; Maschinenachszuordnung Kanal 1:` |
| `$MC_AXCONF_MACHAX_USED[0]=1` | |
| `$MC_AXCONF_MACHAX_USED[1]=2` | |
| `$MC_AXCONF_MACHAX_USED[2]=3` | |
| `CHANDATA(2)` | |
| | `; Maschinenachszuordnung Kanal 2:` |
| `$MC_AXCONF_MACHAX_USED[0]=4` | |
| `$MC_AXCONF_MACHAX_USED[1]=5` | |
| `CHANDATA(1)` | |
| | `; Axiale Maschinendaten:` |
| | `; Genauhaltfenster grob:` |
| `$MA_STOP_LIMIT_COARSE[AX1]=0.2` | `; Achse 1` |
| `$MA_STOP_LIMIT_COARSE[AX2]=0.2` | `; Achse 2` |
| | `; Genauhaltfenster fein:` |
| `$MA_STOP_LIMIT_FINE[AX1]=0.01` | `; Achse 1` |
| `$MA_STOP_LIMIT_FINE[AX1]=0.01` | `; Achse 2` |

**ACHTUNG**

**CHANDATA-Anweisung**

Im Teileprogramm darf die `CHANDATA`-Anweisung nur für den Kanal gesetzt werden, auf dem das NC-Programm abgearbeitet wird. D. h. die Anweisung kann dazu benutzt werden, NC-Programme davor zu schützen, dass sie auf einem nicht vorgesehenen Kanal abgearbeitet werden.

Im Fehlerfall wird die Programmabarbeitung abgebrochen.

**Hinweis**

INI-Dateien in Joblisten enthalten keine `CHANDATA`-Anweisungen.

## Initialisierungsprogramme sichern (COMPLETE, INITIAL)

Die Dateien des Arbeitsspeichers können auf einem externen PC gesichert und von dort wieder eingelesen werden.

- Die Dateien werden mit `COMPLETE` gesichert.
- Mit `INITIAL` wird über alle Bereiche eine INI-Datei (_N_INITIAL_INI) erzeugt.

## Initialisierungsprogramme einlesen

| ACHTUNG |
| --- |
| **Datenverlust**<br>Wird die Datei mit dem Namen "INITIAL_INI" eingelesen, so werden alle Daten, die in der Datei nicht versorgt werden, mit Standarddaten initialisiert. Ausgenommen davon sind nur die Maschinendaten. Es werden also **Settingdaten, Werkzeugdaten, NPV, GUD-Werte, ...** mit Standarddaten (normalerweise "NULL") versorgt. |

Zum Einlesen von einzelnen Maschinendaten eignet sich z. B. die Datei COMPLETE_TEA_INI. In dieser Datei erwartet die Steuerung nur Maschinendaten. Damit bleiben die anderen Datenbereiche in diesem Fall unberührt.

## Initialisierungsprogramme laden

Die INI-Programme können auch als Teileprogramme angewählt und aufgerufen werden, wenn sie nur Daten eines Kanals verwenden. So ist es auch möglich, programmgesteuerte Daten zu initialisieren.

# 3.5 Dateihandling

## 3.5.1 Datei schreiben (WRITE)

Mit dem `WRITE`-Befehl können Sätze/Daten aus dem NC-Programm an das Ende einer sich im passiven Filesystem oder auf einem externen Programmspeicher befindlichen Datei (Protokolldatei) geschrieben werden. Dies kann auch das gerade in Abarbeitung befindliche Programm sein.

**Hinweis**

Eine per `WRITE`-Befehl zu beschreibende Datei wird neu angelegt, wenn sie nicht im Programmspeicher existiert.

### Voraussetzung

Die aktuell eingestellte Schutzstufe muss gleich oder größer dem WRITE-Recht der Datei sein. Ist dies nicht der Fall, wird der Zugriff mit Fehlermeldung (Rückgabewert der Fehlervariablen = 13) abgelehnt.

### Syntax

```
DEF INT <Fehler>
...
WRITE(<Fehler>,"<Dateiname>"/"<ExtG>","<Satz/Daten>")
```

## Bedeutung

| | | | |
|---|---|---|---|
| WRITE: | Befehl zum Anfügen eines Satzes bzw. von Daten an das Ende der angegebenen Datei | | |
| <Fehler>: | **Parameter 1:** Variable für die Rückgabe des Fehlerwerts | | |
| | Typ: | INT | |
| | Wert: | 0 | kein Fehler |
| | | 1 | Pfad nicht erlaubt |
| | | 2 | Pfad nicht gefunden |
| | | 3 | Datei nicht gefunden |
| | | 4 | falscher Dateityp |
| | | 10 | Datei ist voll |
| | | 11 | Datei wird benutzt |
| | | 12 | keine Ressourcen frei |
| | | 13 | keine Zugriffsrechte |
| | | 14 | fehlendes oder fehlgeschlagenes EXTOPEN für das Ausgabegerät |
| | | 15 | Fehler beim Schreiben auf externes Gerät |
| | | 16 | ungültiger externer Pfad programmiert |
| <Dateiname>: | **Parameter 2:** Name der Datei, in der der angegebene Satz bzw. die angegebenen Daten angefügt werden sollen | | |
| | Typ: | STRING | |
| | Vor dem eigentlichen Dateinamen kann der absolute Pfad angegeben werden. Ohne Pfadangabe wird die Datei im aktuellen Verzeichnis (= Verzeichnis des angewählten Programms) gesucht.<br>Regeln zur Pfadangabe siehe "Adressierung von Dateien des Programmspeichers (Seite 563)". | | |
| <ExtG>: | Sollen Daten über die Funktion "Process DataShare" auf ein externes Gerät/Datei ausgegeben werden, muss statt des Dateinamens der symbolische Bezeichner für das zu öffnende externe Gerät/Datei angegeben werden. | | |
| | Typ: | STRING | |
| | Weitere Informationen siehe "Process DataShare - Ausgabe auf ein externes Gerät/Datei (EXTOPEN, WRITE, EXTCLOSE) (Seite 1014)".<br>**Hinweis:**<br>Der Bezeichner muss identisch zu dem im EXTOPEN-Befehl angegebenen Bezeichner sein. | | |
| <Satz/Daten>: | **Parameter 3:** Satz bzw. Daten, die in der angegebenen Datei angefügt werden sollen. | | |
| | Typ: | STRING | |

**Hinweis**

Beim Schreiben ins passive Dateisystem oder auf einen externen Programmspeicher fügt der `WRITE`-Befehl implizit ein "LF"-Zeichen (LINE FEED = neue Zeile) an das Ende des Ausgabestrings an.

Für die Ausgabe auf ein externes Gerät/Datei über die Funktion "Process DataShare" gilt dieses Verhalten nicht. Soll ein "LF" mit ausgegeben werden, muss das explizit im Ausgabestring mit angegeben werden.

→ Siehe hierzu Beispiel 3: Implizites/explizites "LF"!

## Randbedingungen

- **Maximale Dateigröße (→ Maschinenhersteller!)**
  Die maximal mögliche Dateigröße von Protokolldateien im passiven Filesystem wird eingestellt mit dem Maschinendatum:
  MD11420 $MN_LEN_PROTOCOL_FILE
  Die maximale Dateigröße gilt für alle Dateien, die mit dem `WRITE`-Befehl im passiven Filesystem angelegt werden. Bei Überschreitung wird eine Fehlermeldung ausgegeben und der Satz bzw. die Daten werden nicht abgespeichert. Sofern der Speicher ausreicht, kann eine neue Datei angelegt werden.

## Beispiele

### Beispiel 1: WRITE-Befehl ins passive Filesystem ohne absolute Pfadangabe

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 DEF INT ERROR` | `; Definition der Fehlervariablen.` |
| `N20 WRITE(ERROR,"PROT","PROTOKOLL VOM 7.2.97")` | `; Schreibe den Text "PROTOKOLL VOM 7.2.97" in die Datei _N_PROT_MPF.` |
| `N30 IF ERROR` | `; Fehlerauswertung.` |
| `N40 MSG ("Fehler bei WRITE-Befehl:" << ERROR)` | |
| `N50 M0` | |
| `N60 ENDIF` | |
| `...` | |

### Beispiel 2: WRITE-Befehl ins passive Filesystem mit absoluter Pfadangabe

| **Programmcode** |
|---|
| `...` |
| `WRITE(ERROR,"/_N_WKS_DIR/_N_PROT_WPD/_N_PROT_MPF","PROTOKOLL VOM 7.2.97")` |
| `...` |

### Beispiel 3: Implizites/explizites "LF"

a, Schreiben ins passive Dateisystem mit implizit erzeugtem "LF"

**Programmcode**

```
...
N110 DEF INT ERROR
N120 WRITE(ERROR,"/_N_MPF_DIR/_N_MYPROTFILE_MPF","MY_STRING")
N130 WRITE(ERROR,"/_N_MPF_DIR/_N_MYPROTFILE_MPF","MY_STRING")
N140 M30
```

Ausgabeergebnis:

MY_STRING

MY_STRING

b, Schreiben in externe Datei ohne implizit erzeugtem "LF"

**Programmcode**

```
...
N200 DEF STRING[30] DEV_1
N210 DEF INT ERROR
N220 DEV_1="LOCAL_DRIVE/myprotfile.mpf"
N230 EXTOPEN(ERROR,DEV_1)
N240 WRITE(ERROR,DEV_1,"MY_STRING")
N250 WRITE(ERROR,DEV_1,"MY_STRING")
N260 EXTCLOSE(ERROR,DEV_1)
N270 M30
```

Ausgabeergebnis:

MY_STRINGMY_STRING

c, Schreiben in externe Datei mit explizit programmiertem "LF"

Um dasselbe Ergebnis wie bei a, zu erzielen, muss Folgendes programmiert werden:

**Programmcode**

```
...
N200 DEF STRING[30] DEV_1
N210 DEF INT ERROR
N220 DEV_1="LOCAL_DRIVE/myprotfile.mpf"
N230 EXTOPEN(ERROR,DEV_1)
N240 WRITE(ERROR,DEV_1,"MY_STRING'H0A'")
N250 WRITE(ERROR,DEV_1,"MY_STRING'H0A'")
N260 EXTCLOSE(ERROR,DEV_1)
N270 M30
```

Ausgabeergebnis:

MY_STRING

MY_STRING

## 3.5.2 Datei löschen (DELETE)

Mit dem DELETE-Befehl können alle Dateien gelöscht werden, egal, ob diese per WRITE-Befehl entstanden sind oder nicht. Auch Dateien, die unter höherer Zugriffsstufe erstellt wurden, können mit DELETE gelöscht werden.

### Syntax

```
DEF INT <Fehler>
DELETE(<Fehler>,"<Dateiname>")
```

### Bedeutung

| | | | |
|---|---|---|---|
| DELETE: | Befehl zum Löschen der angegebenen Datei | | |
| <Fehler>: | Variable für die Rückgabe des Fehlerwerts | | |
| | Typ. | INT | |
| | Wert: | 0 | kein Fehler |
| | | 1 | Pfad nicht erlaubt |
| | | 2 | Pfad nicht gefunden |
| | | 3 | Datei nicht gefunden |
| | | 4 | falscher Dateityp |
| | | 11 | Datei wird benutzt |
| | | 12 | keine Ressourcen frei |
| | | 20 | sonstiger Fehler |
| <Dateiname>: | Name der zu löschenden Datei | | |
| | Typ: | STRING | |
| | Vor dem eigentlichen Dateinamen kann der absolute Pfad angegeben werden. Ohne Pfadangabe wird die Datei im aktuellen Verzeichnis (= Verzeichnis des angewählten Programms) gesucht.<br>Regeln zur Pfadangabe siehe "Adressierung von Dateien des Programmspeichers (Seite 563)". | | |

### Beispiel

| **Programmcode** | **Kommentar** |
|---|---|
| N10 DEF INT ERROR | ; Definition der Fehlervariablen. |
| N15 STOPRE | ; Vorlaufstopp. |
| N20 DELETE(ERROR,"/_N_SPF_DIR/_N_TEST1_SPF") | ; Lösche die Datei TEST1 im Unterprogrammverzeichnis. |
| N30 IF ERROR | ; Fehlerauswertung. |
| N40 MSG("Fehler bei DELETE-Befehl:" <<ERROR) | |

| Programmcode | Kommentar |
|---|---|
| N50 M0 | |
| N60 ENDIF | |

## 3.5.3 Zeilen in Datei lesen (READ)

Der READ-Befehl liest in der angegebenen Datei eine oder mehrere Zeilen und legt die gelesenen Informationen in einem Feld vom Typ STRING ab. Jede gelesene Zeile belegt in diesem Feld ein Feldelement.

### Voraussetzung

Die aktuell eingestellte Schutzstufe muss gleich oder größer dem READ-Recht der Datei sein. Ist dies nicht der Fall, wird der Zugriff mit Fehlermeldung (Rückgabewert der Fehlervariablen = 13) abgelehnt.

### Syntax

```
DEF INT <Fehler>
DEF STRING[<Stringlänge>] <Ergebnis>[<n>,<m>]
READ(<Fehler>,"<Dateiname>",<Anfangszeile>,<Zeilenanzahl>,<Ergebnis>
)
```

### Bedeutung

| | | | |
|---|---|---|---|
| READ: | Befehl zum Lesen von Zeilen der angegebenen Datei und zur Ablage dieser Zeilen in einem Variablenfeld. | | |
| <Fehler>: | Variable für die Rückgabe des Fehlerwerts (Call-By-Reference-Parameter) | | |
| | Typ. | INT | |
| | Wert: | 0 | kein Fehler |
| | | 1 | Pfad nicht erlaubt |
| | | 2 | Pfad nicht gefunden |
| | | 3 | Datei nicht gefunden |
| | | 4 | falscher Dateityp |
| | | 11 | Datei wird benutzt |
| | | 13 | Zugriffsrechte nicht ausreichend |
| | | 21 | Zeile nicht vorhanden (Parameter <Anfangszeile> oder <Zeilenanzahl> größer als Anzahl der Zeilen in der angegebenen Datei). |
| | | 22 | Feldlänge der Ergebnisvariablen (<Ergebnis>) ist zu klein. |
| | | 23 | Zeilenbereich zu groß (Parameter <Zeilenanzahl> so groß gewählt, dass über das Dateiende hinausgelesen wird). |

| | | | |
|---|---|---|---|
| <Dateiname>: | Name der zu lesenden Datei (Call-By-Value-Parameter) | | |
| | Typ: | STRING | |
| | Vor dem eigentlichen Dateinamen kann der absolute Pfad angegeben werden. Ohne Pfadangabe wird die Datei im aktuellen Verzeichnis (= Verzeichnis des angewählten Programms) gesucht.<br>Regeln zur Pfadangabe siehe "Adressierung von Dateien des Programmspeichers (Seite 563)". | | |
| <Anfangszeile>: | Anfangszeile des zu lesenden Dateibereichs (Call-By-Value-Parameter) | | |
| | Typ: | INT | |
| | Wert: | 0 | Es werden die mit dem Parameter <Zeilenanzahl> angegebene Anzahl an Zeilen vor dem Dateiende gelesen. |
| | | 1 … n | Nummer der ersten zu lesenden Zeile. |
| <Zeilenanzahl>: | Anzahl der zu lesenden Zeilen (Call-By-Value-Parameter) | | |
| | Typ: | INT | |
| <Ergebnis>: | Ergebnisvariable (Call-By-Reference-Parameter)<br>Variablenfeld, in dem der gelesene Text abgelegt wird. | | |
| | Typ: | STRING (max. Länge: 255) | |
| | Wenn im Parameter <Zeilenanzahl> weniger Zeilen angegeben sind als die Feldgröße [<n>,<m>] der Ergebnisvariablen beträgt, dann werden die restlichen Feldelemente nicht verändert.<br>Der Abschluss einer Zeile durch die Steuerzeichen "LF" (Line Feed) oder "CR LF" (Carriage Return Line Feed) wird **nicht** in der Ergebnisvariablen abgelegt.<br>Gelesene Zeilen werden abgeschnitten, wenn die Zeile länger ist als die definierte Stringlänge. Es erfolgt keine Fehlermeldung. | | |

**Hinweis**

Binäre Files können nicht eingelesen werden. Es wird der Fehler "falscher Dateityp" (Rückgabewert der Fehlervariablen = 4) ausgegeben. Folgenden Dateitypen sind nicht lesbar: _BIN, _EXE, _OBJ, _LIB, _BOT, _TRC, _ACC, _CYC, _NCK.

## Beispiel

```
Programmcode                                          Kommentar
N10 DEF INT ERROR                                     ; Definition der Fehlervariablen.
N20 DEF STRING[255] RESULT[5]                         ; Definition der Ergebnisvariablen.
N30 READ(ERROR,"/_N_CST_DIR/_N_TESTFI-                ; Dateiname mit Domain-, Dateikennung
LE_MPF",1,5,RESULT)                                   und Pfadangabe.
N40 IF ERROR <>0                                      ; Fehlerauswertung.
N50 MSG("FEHLER"<<ERROR<<"BEI READ-BEFEHL")
N60 M0
N70 ENDIF
...
```

## 3.5.4 Vorhandensein einer Datei prüfen (ISFILE)

Mit dem `ISFILE`-Befehl kann geprüft werden, ob eine Datei im Programmspeicher existiert.

### Syntax

```
<Ergebnis>=ISFILE("<Dateiname>")
```

### Bedeutung

| | | | |
|---|---|---|---|
| `ISFILE`: | Befehl zum Prüfen des Vorhandenseins einer Datei | | |
| `<Dateiname>`: | Name der Datei, deren Vorhandensein geprüft werden soll. | | |
| | Typ: | STRING | |
| | Vor dem eigentlichen Dateinamen kann der absolute Pfad angegeben werden. Ohne Pfadangabe wird die Datei im aktuellen Verzeichnis (= Verzeichnis des angewählten Programms) gesucht.<br>Regeln zur Pfadangabe siehe "Adressierung von Dateien des Programmspeichers (Seite 563)". | | |
| `<Ergebnis>`: | Ergebnisvariable zur Aufnahme des Prüfergebnisses | | |
| | Typ. | BOOL | |
| | Wert: | TRUE | Datei vorhanden |
| | | FALSE | Datei nicht vorhanden |

### Beispiele

#### Beispiel 1

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 DEF BOOL RESULT` | `; Definition der Ergebnisvariablen.` |
| `N20 RESULT=ISFILE("TESTFILE")` | |
| `N30 IF(RESULT==FALSE)` | |
| `N40  MSG("DATEI NICHT VORHANDEN")` | |
| `N50  M0` | |
| `N60 ENDIF` | |
| `...` | |

#### Beispiel 2

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 DEF BOOL RESULT` | `; Definition der Ergebnisvariablen.` |
| `N20 RESULT=ISFILE("TESTFILE")` | |
| `N30 IF(NOT ISFILE("TESTFILE"))` | |
| `N40  MSG("DATEI NICHT VORHANDEN")` | |
| `N50  M0` | |
| `N60 ENDIF` | |
| `...` | |

## 3.5.5 Datei-Informationen auslesen (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO)

Über die Befehle FILEDATE, FILETIME, FILESIZE, FILESTAT und FILEINFO können bestimmte Datei-Informationen wie Datum/Uhrzeit des letzten schreibenden Zugriffs, aktuelle Dateigröße, Datei-Status oder die Summe dieser Informationen ausgelesen werden.

### Voraussetzung

Die aktuell eingestellte Schutzstufe muss gleich oder größer dem Show-Recht des übergeordneten Verzeichnisses sein. Ist dies nicht der Fall, wird der Zugriff mit Fehlermeldung (Rückgabewert der Fehlervariablen = 13) abgelehnt.

### Syntax

```
FILE....(<Fehler>,"<Dateiname>",<Ergebnis>)
```

### Bedeutung

| | | | |
|---|---|---|---|
| FILEDATE: | Liefert das **Datum** des letzten Schreibzugriffs auf eine Datei | | |
| FILETIME: | Liefert die **Uhrzeit** des letzten Schreibzugriffs auf eine Datei | | |
| FILESIZE: | Liefert die **aktuelle Größe** einer Datei | | |
| FILESTAT: | Liefert für den **Status** einer Datei bezüglich folgender **Rechte**:<br>• Lesen (r: read)<br>• Schreiben (w: write)<br>• Ausführen (x: execute)<br>• Anzeigen (s: show)<br>• Löschen (d: delete)<br>**Hinweis:**<br>Diese Schutzstufen sind spezielle Eigenschaften des passiven Filesystems. Beim Zugriff auf externe Programmspeicher liefert FILESTAT daher nur Default-Zugriffsrechte (77777). | | |
| FILEINFO: | Liefert für eine Datei die **Summe der Informationen**, die über FILEDATE, FILETIME, FILESIZE und FILESTAT auslesbar sind | | |
| <Fehler>: | Variable für die Rückgabe des Fehlerwerts (Call-By-Reference-Parameter) | | |
| | Typ. | VAR INT | |
| | Wert: | 0 | kein Fehler |
| | | 1 | Pfad nicht erlaubt |
| | | 2 | Pfad nicht gefunden |
| | | 3 | Datei nicht gefunden |
| | | 4 | falscher Dateityp |
| | | 13 | Zugriffsrechte nicht ausreichend |
| | | 22 | Stringlänge der Ergebnisvariablen (<Ergebnis>) ist zu klein. |

| | | | | |
|---|---|---|---|---|
| <Dateiname>: | Name der Datei, von der Datei-Information(en) ausgelesen werden soll(en) | | | |
| | Typ: | CHAR[160] | | |
| | Vor dem eigentlichen Dateinamen kann der absolute Pfad angegeben werden. Ohne Pfadangabe wird die Datei im aktuellen Verzeichnis (= Verzeichnis des angewählten Programms) gesucht.<br>Regeln zur Pfadangabe siehe "Adressierung von Dateien des Programmspeichers (Seite 563)". | | | |
| <Ergebnis>: | Ergebnisvariable (Call-By-Reference-Parameter)<br>Variable, in der die angeforderte Datei-Information abgelegt wird. | | | |
| | Typ: | VAR CHAR[8] | bei | FILEDATE<br>Format: "dd.mm.yy" |
| | | VAR CHAR[8] | bei | FILETIME<br>Format: "hh:mm.ss" |
| | | VAR INT | bei | FILESIZE<br>Die Dateigröße wird in Byte ausgegeben. |
| | | VAR CHAR[5] | bei | FILESTAT<br>Format: "rwxsd"<br>(r: **r**ead, w: **w**rite, x: e**x**ecute, s: **s**how, d: **d**elete) |
| | | VAR CHAR[32] | bei | FILEINFO<br>Format: "rwxsd nnnnnnnn dd.mm.yy hh:mm:ss" |

## Beispiel

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 DEF INT ERROR` | `; Definition der Fehlervariablen.` |
| `N20 STRING[32] RESULT` | `; Definition der Ergebnisvariablen.` |
| `N30 FILEINFO(ERROR,"/_N_MPF_DIR/_N_TESTFILE_MPF",RESULT)` | `; Dateiname mit Domain-, Dateikennung und Pfadangabe.` |
| `N40 IF ERROR <> 0` | `; Fehlerauswertung` |
| `N50 MSG("FEHLER"<<ERROR<<"BEI FILEINFO-BEFEHL")` | |
| `N60 M0` | |
| `N70 ENDIF` | |
| `...` | |

Das Beispiel könnte in der Ergebnisvariablen RESULT z. B. folgendes Ergebnis liefern:

"77777 12345678 26.05.00 13:51:30"

# 3.6 Schutzbereiche

## 3.6.1 Schutzbereiche definieren (CPROTDEF, NPROTDEF)

Schutzbereiche, die Maschinenelemente vor Kollisionen schützen sollen, werden im Teileprogramm jeweils in Blöcken definiert. Diese enthalten folgende Elemente:

1. Festlegung der Arbeitsebene
   Vor der eigentlichen Schutzbereichsdefinition muss die Arbeitsebene angewählt werden, auf die sich die Konturbeschreibung des Schutzbereichs beziehen soll.
2. Definitionsbeginn
   Je nach NC-Befehl wird entweder ein kanalspezifischer oder maschinenspezifischer Schutzbereich angelegt.
3. Konturbeschreibung des Schutzbereichs
   Die Kontur eines Schutzbereichs wird mittels Verfahrbewegungen beschrieben. Diese werden nicht ausgeführt und haben keine Verbindung zu vorhergehenden oder nachfolgenden Geometriebeschreibungen. Sie definieren ausschließlich den Schutzbereich.
4. Definitionsende

### Syntax

```
DEF INT <Var>
G17/G18/G19
CPROTDEF/NPROTDEF(<n>,<t>,<AppLim>,<AppPlus>,<AppMinus>)
G0/G1/... X/Y/Z...
...
EXECUTE(<Var>)
```

### Bedeutung

| | | |
|---|---|---|
| `DEF INT <Var>`: | Definition einer lokalen Hilfsvariablen vom Datentyp INTEGER | |
| `<Var>`: | Name der Hilfsvariablen | |
| `G17/G18/G19`: | Arbeitsebene<br>**Hinweis:**<br>Die Arbeitsebene darf vor dem Definitionsende nicht geändert werden. Eine Programmierung der Applikate zwischen Definitionsbeginn und -ende ist nicht zulässig. | |
| `CPROTDEF()`: | Vordefinierte Prozedur zur Definition eines **kanal**spezifischen Schutzbereichs | |
| `NPROTDEF()`: | Vordefinierte Prozedur zur Definition eines **maschinen**spezifischen Schutzbereichs | |
| `<n>`: | Nummer des definierten Schutzbereichs | |
| | Datentyp: | INT |

| | | | |
|---|---|---|---|
| <t>: | Typ des Schutzbereichs | | |
| | Datentyp: | BOOL | |
| | Wert: | TRUE | **Werkzeug**bezogener Schutzbereich |
| | | FALSE | **Werkstück**bezogener Schutzbereich |
| <AppLim>: | Art der Begrenzung in der 3. Dimension | | |
| | Datentyp: | INT | |
| | Wert: | 0 | keine Begrenzung |
| | | 1 | Begrenzung in Plus-Richtung |
| | | 2 | Begrenzung in Minus-Richtung |
| | | 3 | Begrenzung in Plus- und Minus-Richtung |
| <AppPlus>: | Wert der Begrenzung in Plus-Richtung der 3. Dimension | | |
| | Datentyp: | REAL | |
| <AppMinus>: | Wert der Begrenzung in Minus-Richtung der 3. Dimension | | |
| | Datentyp: | REAL | |
| G0/G1/... X/Y/Z... ...: | Die Kontur eines Schutzbereichs wird mit maximal 11 Verfahrbewegungen in der angewählten Arbeitsebene beschrieben. Dabei ist die erste Verfahrbewegung die Bewegung an die Kontur. Der letzte Punkt der Konturbeschreibung muss immer mit dem ersten Punkt der Konturbeschreibung zusammenfallen.<br>Als Schutzbereich gilt der Bereich links von der Kontur:<br>• Innenschutzbereich<br>Die Kontur für einen Innenschutzbereich ist gegen den Uhrzeigersinn zu beschreiben.<br>• Außenschutzbereich (nur zulässig für werkstückbezogene Schutzbereiche!)<br>Die Kontur für einen Außenschutzbereich ist im Uhrzeigersinn zu beschreiben.<br>Folgende Konturelemente sind zulässig:<br>• G0, G1 für gerade Konturelemente<br>• G2 für Kreisabschnitte im Uhrzeigersinn<br>Nur zulässig bei werkstückbezogenen Schutzbereichen!<br>Nicht zulässig bei werkzeugbezogenen Schutzbereichen, da sie nur konvex sein dürfen.<br>• G3 für Kreisabschnitte gegen den Uhrzeigersinn<br>**Hinweis:**<br>Ein Schutzbereich kann nicht durch einen Vollkreis beschrieben werden. Ein Vollkreis muss in zwei Teilkreise aufgeteilt werden.<br>**Hinweis:**<br>Die Folge G2 → G3 bzw. G3 → G2 ist nicht zulässig! Zwischen die beiden Kreissätze muss ein kurzer G1-Satz eingefügt werden. | | |
| EXECUTE(<Var>): | Vordefinierte Prozedur, die das Ende der Definition markiert<br>Mit EXECUTE wird zur normalen Programmbearbeitung zurückgeschaltet. | | |

## Beispiel

Siehe Beispiel unter "Schutzbereiche aktivieren/deaktivieren (CPROT, NPROT) (Seite 587)".

## Weitere Informationen

### Maschinenspezifische Schutzbereiche

Ein maschinenspezifischer Schutzbereich bzw. dessen Kontur wird mittels der Geometrieachsen definiert, d. h. bezogen auf das Basiskoordinatensystem (BKS) eines Kanals. Damit eine korrekte Schutzbereichsüberwachung in allen Kanälen, in denen der maschinenspezifische Schutzbereich aktiv ist, stattfinden kann, muss das Basiskoordinatensystem (BKS) aller betroffenen Kanäle identisch sein:

- Lage des Koordinatenursprungs bezogen auf den Maschinennullpunkt
- Orientierung der Koordinatenachsen

### Bezugspunkt der Konturbeschreibung

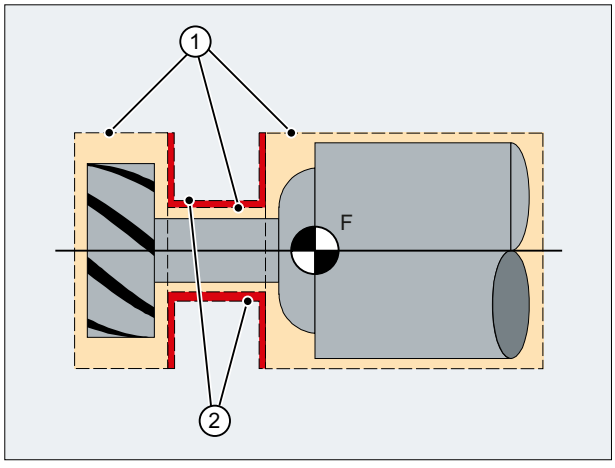
- Werkzeugbezogene Schutzbereiche
  Koordinaten für **werkzeug**bezogene Schutzbereiche sind absolut, bezogen auf den **Werkzeugträgerbezugspunkt F**, anzugeben.
- Werkstückbezogene Schutzbereiche
  Koordinaten für **werkstück**bezogene Schutzbereiche sind absolut, bezogen auf den Nullpunkt des **Basiskoordinatensystems (BKS)**, anzugeben.

### Rotationssymmetrische Schutzbereiche

Bei rotationssymmetrischen Schutzbereichen (z. B. Spindelfutter) muss die Gesamtkontur beschrieben werden, nicht nur die Kontur bis zur Drehmitte.

### Werkzeugbezogene Schutzbereiche

Werkzeugbezogene Schutzbereiche müssen immer konvex sein. Falls ein konkaver Schutzbereich gewünscht ist, ist dieser in mehrere konvexe Schutzbereiche zu zerlegen.




① Konvexe Schutzbereiche
② Konkave Schutzbereiche (**nicht zulässig!**)
F Werkzeugträgerbezugspunkt

**Randbedingungen**

Während der Definition eines Schutzbereichs dürfen folgende Funktionen nicht aktiv sein bzw. verwendet werden:

- Werkzeugradiuskorrektur (Fräserradiuskorrektur, Schneidenradiuskorrektur)
- Transformation
- Referenzpunktanfahren (`G74`)
- Festpunktanfahren (`G75`)
- Verweilzeit (`G4`)
- Satzvorlauf-Stopp (`STOPRE`)
- Programmende (`M17`, `M30`)
- M-Funktionen: `M0`, `M1`, `M2`

## 3.6.2 Schutzbereiche aktivieren/deaktivieren (CPROT, NPROT)

Vorher im Teileprogramm definierte Schutzbereiche können jederzeit aktiviert bzw. für eine spätere Aktivierung durch das PLC-Anwenderprogramm voraktiviert werden. Aktive Schutzbereiche können jederzeit wieder deaktiviert werden.

Bei der Aktivierung bzw. Voraktivierung besteht zudem die Möglichkeit, den Bezugspunkt des Schutzbereichs relativ zu verschieben.

**Hinweis**

Ein Schutzbereich wird erst nach dem Referenzieren aller Geometrieachsen des Kanals, in dem er aktiviert wurde, berücksichtigt.

**Hinweis**

**Überwachung der Schutzbereiche**

Ist kein werkzeugbezogener Schutzbereich aktiv, wird die Werkzeugbahn gegen die werkstückbezogenen Schutzbereiche geprüft.

Ist kein werkstückbezogener Schutzbereich aktiv, findet keine Schutzbereichsüberwachung statt.

### Syntax

```
CPROT(<n>,<Status>,<XMov>,<YMov>,<ZMov>)
NPROT(<n>,<Status>,<XMov>,<YMov>,<ZMov>)
```

## Bedeutung

| | | | |
|---|---|---|---|
| CPROT: | Vordefinierte Prozedur zum Aktivieren eines **kanal**spezifischen Schutzbereichs | | |
| NPROT: | Vordefinierte Prozedur zum Aktivieren eines **maschinen**spezifischen Schutzbereichs | | |
| <n>: | Nummer des Schutzbereichs | | |
| | Datentyp: | INT | |
| <Status>: | Mit diesem Parameter wird der kanalspezifische Aktivierungsstatus gesetzt | | |
| | Datentyp: | INT | |
| | Wert: | 0 | Schutzbereich deaktivieren |
| | | 1 | Schutzbereich voraktivieren |
| | | 2 | Schutzbereich aktivieren |
| | | 3 | Schutzbereich voraktivieren mit bedingtem Stopp |
| <XMov>,<YMov>,<ZMov>: | Additive Verschiebungswerte in X/Y/Z-Richtung<br>Die Verschiebung kann in 1, 2 oder 3 Dimensionen erfolgen. Die Verschiebungswerte beziehen sich auf:<br>• den Maschinennullpunkt bei **werkstück**bezogenem Schutzbereich<br>• den Werkzeugträgerbezugspunkt F bei **werkzeug**bezogenem Schutzbereich | | |
| | Datentyp: | REAL | |

## Beispiel

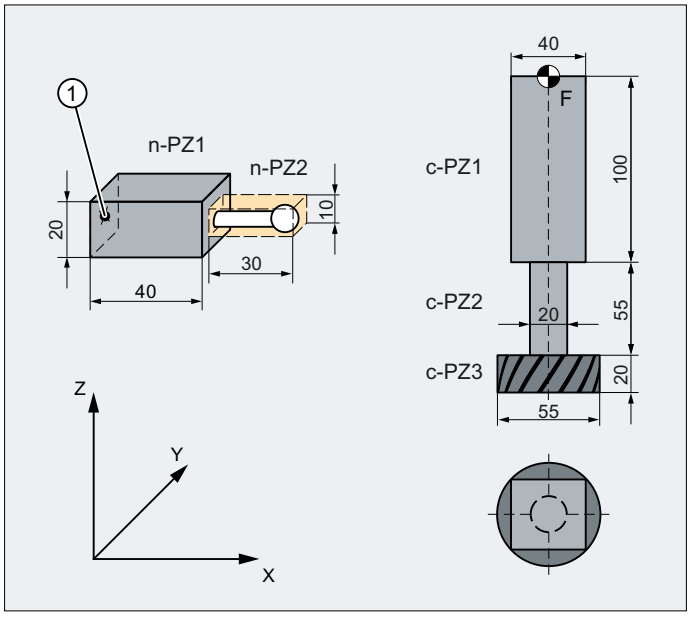
Für eine Fräsmaschine soll eine mögliche Kollision des Fräsers mit dem Messtaster überwacht werden. Die Lage des Messtasters soll bei der Aktivierung durch eine Verschiebung angegeben werden.

Es werden dafür folgende Schutzbereiche definiert:

- Jeweils ein maschinenspezifischer und werkstückbezogener Schutzbereich für den Messtasterhalter (n-PZ1) und für den Messtaster selbst (n-PZ2).
- Jeweils ein kanalspezifischer und werkzeugbezogener Schutzbereich für den Fräserhalter (c-PZ1), den Fräserschaft (c-PZ2) und für den Fräser selbst (c-PZ3).

Die Orientierung aller Schutzbereiche liegt in Z-Richtung.

Die Lage des Bezugspunkts des Messtasters bei der Aktivierung soll bei X = -120, Y = 60 und Z = 80 liegen.




① Bezugspunkt für den Schutzbereich des Messtasters

F Werkzeugträgerbezugspunkt

| **Programmcode** | **Kommentar** |
|---|---|
| DEF INT PROTZONE | ; Definition einer Hilfsvariablen |
| G17 | ; Arbeitsebene XY |
| | |
| ; Definition der Schutzbereiche: | |
| NPROTDEF(1,FALSE,3,10,-10) | ; Schutzbereich n-PZ1 |
| G01 X0 Y-10 | |
| X40 | |
| Y10 | |
| X0 | |
| Y-10 | |
| EXECUTE(PROTZONE) | |
| NPROTDEF(2,FALSE,3,5,-5) | ; Schutzbereich n-PZ2 |
| G01 X40 Y-5 | |
| X70 | |
| Y5 | |
| X40 | |
| Y-5 | |
| EXECUTE(PROTZONE) | |
| CPROTDEF(1,TRUE,3,0,-100) | ; Schutzbereich c-PZ1 |
| G01 X-20 Y-20 | |
| X20 | |
| Y20 | |
| X-20 | |
| Y-20 | |
| EXECUTE(PROTZONE) | |

| **Programmcode** | **Kommentar** |
|---|---|
| CPROTDEF(2,TRUE,3,-100,-150) | ; Schutzbereich c-PZ2 |
| G01 X0 Y-10 | |
| G03 X0 Y10 J10 | |
| X0 Y-10 J-10 | |
| EXECUTE(PROTZONE) | |
| CPROTDEF(3,TRUE,3,-150,-170) | ; Schutzbereich c-PZ3 |
| G01 X0 Y-27.5 | |
| G03 X0 Y27.5 J27.5 | |
| X0 Y27.5 J-27.5 | |
| EXECUTE(PROTZONE) | |
| | |
| ; Aktivierung der Schutzbereiche: | |
| NPROT(1,2,-120,60,80) | ; Schutzbereich n-PZ1 mit Verschiebung aktivieren |
| NPROT(2,2,-120,60,80) | ; Schutzbereich n-PZ2 mit Verschiebung aktivieren |
| CPROT(1,2,0,0,0) | ; Schutzbereich c-PZ1 aktivieren |
| CPROT(2,2,0,0,0) | ; Schutzbereich c-PZ2 aktivieren |
| CPROT(3,2,0,0,0) | ; Schutzbereich c-PZ3 aktivieren |

## Weitere Informationen

### Aktivierungsstatus nach Hochlaufen der Steuerung

Ein Schutzbereich kann bereits nach dem Hochlaufen der Steuerung und dem Referenzieren der Achsen aktiv sein. Dies ist der Fall, wenn für den Schutzbereich die folgende Systemvariable auf TRUE gesetzt ist:

- $SN_PA_ACTIV_IMMED[<n>] (für maschinenspezifischen Schutzbereich) bzw.
- $SC_PA_ACTIV_IMMED[<n>] (für kanalspezifischen Schutzbereich)
  Der Index "<n>" entspricht der Nummer des Schutzbereichs: 0 = 1. Schutzbereich

Der Schutzbereich wird mit dem Status = 2 und ohne Verschiebung aktiviert.

### Mehrfaches Aktivieren eines Schutzbereichs

Ein maschinenspezifischer Schutzbereich kann gleichzeitig in mehreren Kanälen wirksam sein (z. B. Schutzbereich der Pinole bei zwei gegenüberliegenden Schlitten). Die Überwachung der Schutzbereiche erfolgt nur, wenn alle Geometrieachsen referenziert sind.

Ein Schutzbereich ist in einem Kanal nicht gleichzeitig mit verschiedenen Verschiebungen aktivierbar.

### Schutzbereichsüberwachung bei aktiver Werkzeugradiuskorrektur

Bei aktiver Werkzeugradiuskorrektur ist eine funktionsfähige Schutzbereichsüberwachung nur möglich, wenn die Ebene der Werkzeugradiuskorrektur identisch ist mit der Ebene der Schutzbereichsdefinitionen.

## 3.6.3 Überprüfung auf Schutzbereichsverletzung, Arbeitsfeldbegrenzung und Software-Endschalter (CALCPOSI)

### Funktion

Die Funktion CALCPOSI überprüft im Werkstückkoordinatensystem (WKS), ob ausgehend von der Startposition die **Geometrieachsen** einen vorgegebenen Weg verfahren können, ohne aktive Begrenzungen zu verletzen. Für den Fall, dass der Verfahrweg aufgrund von Begrenzungen nicht vollständig abgefahren werden kann, werden ein positiver, dezimal codierter Statuswert und der maximal mögliche Verfahrweg zurückgegeben.

### Definition

```
INT CALCPOSI(VAR REAL[3] <Start>, VAR REAL[3] <Dist>, VAR REAL[5]
<Limit>, VAR REAL[3] <MaxDist>, BOOL <MeasSys>, INT <TestLim>)
```

### Syntax

```
<Status> = CALCPOSI(VAR <Start>, VAR <Dist>, VAR <Limit>, VAR
<MaxDist>, <MeasSys>, <TestLim>)
```

### Bedeutung

| | | |
|---|---|---|
| `CALCPOSI(...)`: | Vordefinierte Funktion für den Test auf Begrenzungsverletzungen bezüglich der Geometrieachsen | |
| | Vorlaufstopp: | nein |
| | Alleine im Satz: | ja |

| | | | |
|---|---|---|---|
| <Status>: (Teil 1) | Rückgabewert der Funktion. Negative Werte zeigen Fehlerzustände an. | | |
| | Datentyp: | | INT |
| | Wertebereich: | | -8 ≤ x ≤ 100000 |
| | Wert: | 0 | Der Verfahrweg kann vollständig abgefahren werden |
| | | -1 | In <Limit> ist mindestens eine Komponente negativ. |
| | | -2 | Fehler bei einer Transformationsberechnung.<br>Beispiel: Der Verfahrweg führt durch eine Singularität, sodass die Achspositionen nicht definiert sind. |
| | | -3 | Der angegebene Verfahrweg <Dist> und der maximal mögliche Verfahrweg <MaxDist> sind linear abhängig.<br>**Hinweis**<br>Kann nur im Zusammenhang mit <TestLim>, Bit 4 == 1 auftreten. |
| | | -4 | Die Projektion der in <Dist> enthaltenen Verfahrrichtung auf die Begrenzungsfläche ist der Nullvektor bzw. die Verfahrrichtung steht senkrecht auf der verletzten Begrenzungsfläche.<br>**Hinweis**<br>Kann nur im Zusammenhang mit <TestLim>, Bit 5 == 1 auftreten. |
| | | -5 | In <TestLim> sind Bit 4 == 1 UND Bit 5 == 1 |
| | | -6 | Mindestens eine Maschinenachse, die für die Überprüfung der Verfahrgrenzen betrachtet werden muss, ist nicht referenziert. |
| | | -7 | Funktion Kollisionsvermeidung: Ungültige Definition der kinematischen Kette oder der Schutzbereiche. |
| | | -8 | Funktion Kollisionsvermeidung: Die Funktion kann wegen Speichermangel nicht ausgeführt werden. |
| <Status>: (Teil 2) | Einerstelle<br>**Hinweis**<br>Sind gleichzeitig mehrere Grenzen verletzt, wird diejenige gemeldet, die zur stärksten Einschränkung des vorgegebenen Verfahrwegs führt. | | |
| | Wert: | 1 | Software-Endschalter begrenzen den Verfahrweg |
| | | 2 | Arbeitsfeldbegrenzung begrenzt den Verfahrweg |
| | | 3 | Schutzbereiche begrenzen den Verfahrweg |
| | | 4 | Funktion Kollisionsvermeidung: Schutzbereiche begrenzen den Verfahrweg |
| | Zehnerstelle | | |
| | Wert: | 1x | Der Anfangswert verletzt die Grenze |
| | | 2x | Die vorgegebene Gerade verletzt die Grenze.<br>Dieser Wert wird auch dann zurückgegeben, wenn der Endpunkt selbst keine Grenze verletzt, auf dem Weg vom Start- zum Endpunkt aber eine Verletzung eines Grenzwerts auftreten würde (z. B. Durchfahren eines Schutzbereichs, gekrümmte Software-Endschalter im WKS bei nichtlinearen Transformationen, z. B. Transmit). |

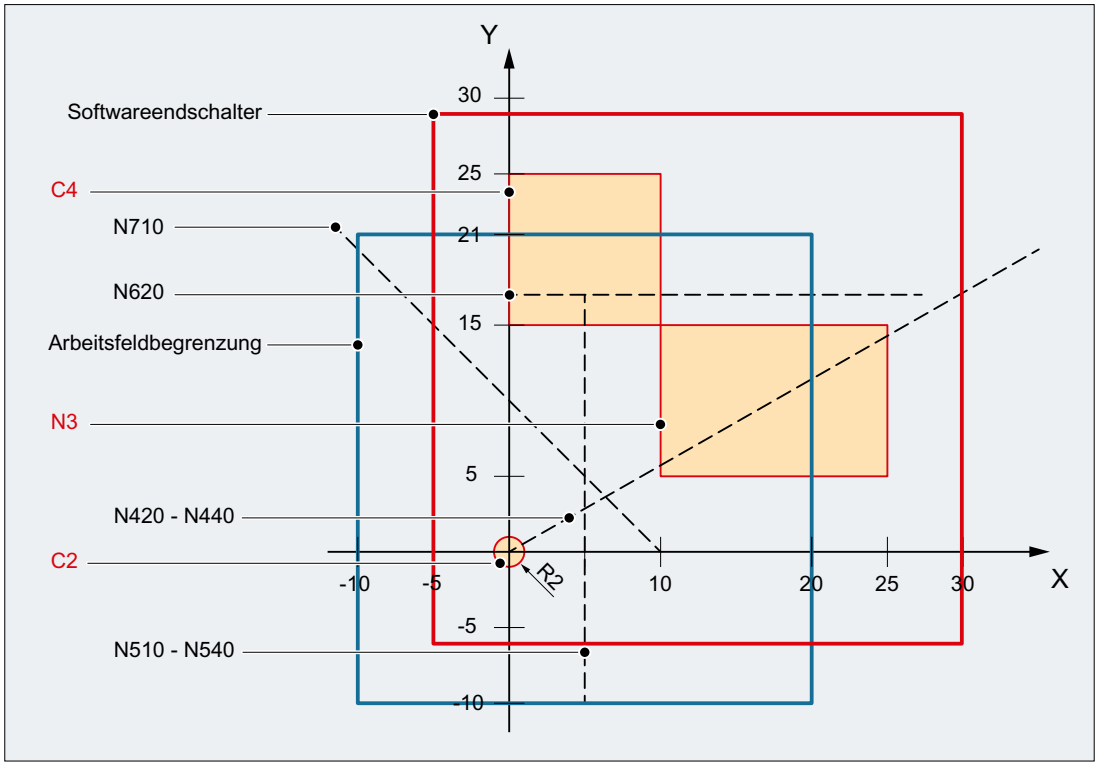
| | | | |
|---|---|---|---|
| <Status>: (Teil 3) | Hunderterstelle | | |
| | Wert: | 1xx | UND Einerstelle == 1 oder 2:<br>Der positive Grenzwert ist verletzt. |
| | | | UND Einerstelle == 3 [1]:<br>Es ist ein NC-spez. Schutzbereich verletzt. |
| | | 2xx | UND Einerstelle == 1 oder 2:<br>Der negative Grenzwert ist verletzt. |
| | | | UND Einerstelle == 3 [1]:<br>Es ist ein kanalspezifischer Schutzbereich verletzt. |
| <Status>: (Teil 4) | Tausenderstelle | | |
| | Wert: | 1xxx | UND Einerstelle == 1 oder 2:<br>Faktor, mit dem die Achsnummer multipliziert wird, die die Grenze verletzt. Die Zählung der Achsen beginnt bei 1.<br>Bezug:<br>• Software-Endschalter: Maschinenachsen<br>• Arbeitsfeldbegrenzung: Geometrieachsen<br>UND Einerstelle == 3 [1]:<br>Faktor, mit dem die Nummer des verletzten Schutzbereichs multipliziert wird. |
| <Status>: (Teil 5) | Einhunderttausenderstelle | | |
| | Wert: | 0xxxxx | Einhunderttausenderstelle == 0: <Dist> bleibt unverändert |
| | | 1xxxxx | In <Dist> wird ein Richtungsvektor zurückgegeben, der die weitere Bewegungsrichtung auf der Begrenzungsfläche definiert.<br>Kann nur bei folgenden Randbedingungen auftreten:<br>• Software-Endschalter oder Arbeitsfeldbegrenzung verletzt (nicht im Startpunkt)<br>• Eine Transformation ist **nicht** aktiv<br>• <TestID>, Bit 4 oder Bit 5 == 1 |
| <Start>: | Referenz auf einen Vektor mit den Startpositionen:<br>• <Start> [0]: 1. Geometrieachse<br>• <Start> [1]: 2. Geometrieachse<br>• <Start> [2]: 3. Geometrieachse | | |
| | Parametertyp: | | Eingang |
| | Datentyp: | | VAR REAL[3] |
| | Wertebereich: | | -max. REAL-Wert ≤ x[<n>] ≤ +max. REAL-Wert |

| | | |
|---|---|---|
| <Dist>: | Referenz auf einen Vektor. | |
| | **Eingang**: inkrementeller Verfahrweg<br>• <Dist> [0]: 1. Geometrieachse<br>• <Dist> [1]: 2. Geometrieachse<br>• <Dist> [2]: 3. Geometrieachse | |
| | **Ausgang** (nur bei gesetzter Einhunderttausenderstelle im <Status>): | |
| | <Dist> enthält als Ausgangswert einen Einheitsvektor **v**, der die weitere Verfahrrichtung im WKS definiert. | |
| | **Fall 1**: Bildung des Vektors **v** bei <TestID>, Bit 4 == 1<br>Die Eingangsvektoren <Dist> und <MaxDist> spannen die Bewegungsebene auf. Diese Ebene wird mit der verletzten Begrenzungsfläche geschnitten. Die Schnittgerade der beiden Ebenen definiert die Richtung des Vektors **v**. Dabei wird die Orientierung (Vorzeichen) so gewählt, dass der Winkel zwischen dem Eingangsvektor <MaxDist> und **v** nicht größer als 90 Grad ist. | |
| | **Fall 2**: Bildung des Vektors **v** bei <TestID>, Bit 5 == 1<br>Der Vektor **v** ist der Einheitsvektor in Richtung der Projektion des in <Dist> enthaltenen Verfahrvektors auf die Begrenzungsfläche. Ist die Projektion des Verfahrvektors auf die Begrenzungsfläche der Nullvektor, wird ein Fehler zurückgeben. | |
| | Parametertyp: | Ein/Ausgang |
| | Datentyp: | VAR REAL[3] |
| | Wertebereich: | -max. REAL-Wert ≤ x[<n>] ≤ +max. REAL-Wert |
| <Limit>: | Referenz auf ein Feld der Länge 5.<br>• <Limit> [0 - 2]: Mindestabstände der Geometrieachsen zu den Begrenzungen:<br>– <Limit> [0]: 1. Geometrieachse<br>– <Limit> [1]: 2. Geometrieachse<br>– <Limit> [2]: 3. Geometrieachse<br>Die Mindestabstände werden eingehalten bei:<br>– Arbeitsfeldbegrenzung: Keine Einschränkungen<br>– Software-Endschalter: Wenn keine Transformation aktiv ist, oder eine Transformation aktiv ist, bei der eine eindeutige Zuordnung der Geometrieachsen zu linearen Maschinenachsen möglich ist, z. B. 5-Achs-Transformationen.<br>• <Limit> [3]: Enthält den Mindestabstand für lineare Maschinenachsen, die z. B. auf Grund einer nichtlinearen Transformation keiner Geometrieachse zugeordnet werden können. Dieser Wert wird außerdem als Grenzwert bei der Überwachung der konventionellen Schutzbereiche und der Schutzbereiche der Kollisionsvermeidung verwendet.<br>• <Limit> [4]: Enthält den Mindestabstand für rotatorische Maschinenachsen, die z. B. auf Grund einer nichtlinearen Transformation keiner Geometrieachse zugeordnet werden können.<br>**Hinweis**<br>Dieser Wert wird nur bei der Überwachung der Software-Endschalter von speziellen Transformationen wirksam. | |
| | Parametertyp: | Eingang |
| | Datentyp: | VAR REAL[5] |
| | Wertebereich: | -max. REAL-Wert ≤ x[n] ≤ +max. REAL-Wert |

| | | | |
|---|---|---|---|
| <MaxDist>: | Referenz auf einen Vektor mit dem inkrementellen Verfahrweg, bei dem der vorgegebene Mindestabstand von einer Achsgrenze von allen beteiligten Maschinenachsen nicht unterschritten wird:<br>• <Dist> [0]: 1. Geometrieachse<br>• <Dist> [1]: 2. Geometrieachse<br>• <Dist> [2]: 3. Geometrieachse<br>Ist der Verfahrweg nicht eingeschränkt, ist der Inhalt dieses Rückgabeparameters gleich dem Inhalt von <Dist>. | | |
| | Bei <TestID>, Bit 4 == 1: <Dist> und <MaxDist><br><MaxDist> und <Dist> müssen als Eingangswerte Vektoren enthalten, die eine Bewegungsebene aufspannt. Die beiden Vektoren müssen voneinander linear unabhängig sein. Der Betrag von <MaxDist> ist beliebig. Zur Berechnung der Bewegungsrichtung siehe die Beschreibung zu <Dist>. | | |
| | Parametertyp: | | Ausgang |
| | Datentyp: | | VAR REAL[3] |
| | Wertebereich: | | -max. REAL-Wert ≤ x[<n>] ≤ +max. REAL-Wert |
| <MeasSys>: | Maßsystem (inch / metrisch) für Positions- und Längenangaben (**optional**) | | |
| | Datentyp: | | BOOL |
| | Wert: | FALSE (Default) | Maßsystem entsprechend des aktuell aktiven G-Befehls aus G-Gruppe 13 (G70, G71, G700, G710).<br>**Hinweis**<br>Bei aktivem G70 und Grundsystem metrisch oder aktivem G71 und Grundsystem inch werden die Systemvariablen $AA_IW und $AA_MW im Grundsystem geliefert und müssen bei Verwendung für CALCPOSI gegebenenfalls umgerechnet werden. |
| | | TRUE | Maßsystem gemäß eingestelltem Grundsystem:<br>MD52806 $MN_ISO_SCALING_SYSTEM |
| <TestLim>: | Bitcodierte Auswahl der zu überwachenden Begrenzungen (**optional**) | | |
| | Datentyp: | | INT |
| | Defaultwert: | | Bit 0, 1, 2, 3, 6, 7 == 1 (207) |
| | **Bit** | **Dezimal** | **Bedeutung** |
| | 0 | 1 | Software-Endschalter |
| | 1 | 2 | Arbeitsfeldbegrenzung |
| | 2 | 4 | Aktivierte konventionelle Schutzbereiche |
| | 3 | 8 | Voraktivierte konventionelle Schutzbereiche |
| | 4 | 16 | Bei verletzten Software-Endschaltern bzw. Arbeitsfeldbegrenzungen in <Dist> die Verfahrrichtung entsprechend **Fall 1** (siehe oben) zurückliefern. |
| | 5 | 32 | Bei verletzten Software-Endschaltern bzw. Arbeitsfeldbegrenzungen in <Dist> die Verfahrrichtung entsprechend **Fall 2** (siehe oben) zurückliefern. |
| | 6 | 64 | Aktivierte Schutzbereiche der Kollisionsvermeidung |
| | 7 | 128 | Voraktivierte Schutzbereiche der Kollisionsvermeidung |
| | 8 | 256 | Paare von aktivierten und voraktivierten Schutzbereichen der Kollisionsvermeidung |
| [1] Sind mehrere Schutzbereiche verletzt, wird der Schutzbereich zurückgegeben, der zur stärksten Einschränkung des vorgegebenen Verfahrwegs führt. | | | |

## Beispiel

### Begrenzungen




Im Beispiel sind die wirksamen Software-Endschalter und Arbeitsfeldbegrenzungen in der X-Y-Ebene und folgende drei Schutzbereiche dargestellt:

- C2: Werkzeugbezogener kanalspezifischer Schutzbereich, aktiv, kreisförmig, Radius = 2 mm
- C4: Werkstückbezogener, kanalspezifischer Schutzbereich, voraktiviert, quadratisch, Seitenlänge = 10 mm
- N3: Maschinenspezifischer Schutzbereich, aktiv, rechteckig, Seitenlänge = 10 mm x 15 mm

### NC-Programm

Im NC-Programm werden zunächst die Schutzbereiche und Arbeitsfeldbegrenzungen definiert. Anschließend wird die Funktion `CALCPOSI()` mit verschiedenen Parametrierungen aufgerufen.

**Programmcode**

```
N10 DEF REAL _START[3]
N20 DEF REAL _DIST[3]
N30 DEF REAL _LIMIT[5]
N40 DEF REAL _MAXDIST[3]
N50 DEF INT _PA
N60 DEF INT _STATUS
```

**Programmcode**

```
; Werkzeugbezogener Schutzbereich C2
N70 CPROTDEF(2, TRUE, 0)
N80 G17 G1 X-2 Y0
N90 G3 I2 X2
N100 I-2 X-2
N110 EXECUTE(_PA)
; Werkstückbezogener Schutzbereich C4
N120 CPROTDEF(4, FALSE, 0)
N130 G17 G1 X0 Y15
N140 X10
N150 Y25
N160 X0
N170 Y15
N180 EXECUTE(_PA)
; Maschinenspezifischer Schutzbereich N3
N190 NPROTDEF(3, FALSE, 0)
N200 G17 G1 X10 Y5
N210 X25
N220 Y15
N230 X10
N240 Y5
N250 EXECUTE(_PA)
; Schutzbereiche aktivieren bzw. voraktivieren
N260 CPROT(2, 2, 0, 0, 0)
N270 CPROT(4, 1, 0, 0, 0)
N280 NPROT(3, 2, 0, 0, 0)
; Arbeitsfeldbegrenzungen definieren
N290 G25 XX=-10 YY=-10
N300 G26 XX=20 YY=21
N310 _START[0] = 0.
N320 _START[1] = 0.
N330 _START[2] = 0.
N340 _DIST[0] = 35.
N350 _DIST[1] = 20.
N360 _DIST[2] =  0.
N370 _LIMIT[0] = 0.
N380 _LIMIT[1] = 0.
N390 _LIMIT[2] = 0.
N400 _LIMIT[3] = 0.
N410 _LIMIT[4] = 0.
N420 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST)
N430 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,3)
N440 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,1)
N450 _START[0] =  5.
N460 _START[1] = 17.
N470 _START[2] =  0.
N480 _DIST[0] =  0.
N490 _DIST[1] =-27.
N500 _DIST[2] =  0.
N510 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,14)
N520 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,, 6)
N530 _LIMIT[1] = 2.
```

**Programmcode**

```
N540 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,, 6)
N550 _START[0] = 27.
N560 _START[1] = 17.1
N570 _START[2] =  0.
N580 _DIST[0] =-27.
N590 _DIST[1] =  0.
N600 _DIST[2] =  0.
N610 _LIMIT[3] = 2.
N620 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,12)
N630 _START[0] = 0.
N640 _START[1] = 0.
N650 _START[2] = 0.
N660 _DIST[0] =  0.
N670 _DIST[1] = 30.
N680 _DIST[2] =  0.
N690 TRANS X10
N700 AROT Z45
N710 _STATUS = CALCPOSI(_START,_DIST, _LIMIT, _MAXDIST)
; Frames aus N690 und N700 wieder löschen
N720 TRANS
N730 _START[0] =  0.
N740 _START[1] = 10.
N750 _START[2] =  0.
; Vektoren _DIST und _MAXDIST definieren die Bewegungsebene
N760 _DIST[0] = 30.
N770 _DIST[1] = 30.
N780 _DIST[2] =  0.
N790 _MAXDIST[0] = 1.
N800 _MAXDIST[1] = 0.
N810 _MAXDIST[2] = 1.
N820 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,17)
N830 M30
```

### Ergebnisse von CALCPOSI()

| N... | <Status> | <MaxDist>[0] ≙ X | <MaxDist>[1] ≙ Y | Bemerkungen |
|---|---|---|---|---|
| 420 | 3123 | 8.040 | 4.594 | N3 wird verletzt. |
| 430 | 1122 | 20.000 | 11.429 | Keine Überwachung der Schutzbereiche, Arbeitsfeldbegrenzung wird verletzt. |
| 440 | 1121 | 30.000 | 17.143 | Nur noch Überwachung der Software-Endschalter aktiv. |
| 510 | 4213 | 0.000 | 0.000 | Startpunkt verletzt C4 |
| 520 | 0000 | 0.000 | -27.000 | Voraktivierter C4 wird nicht überwacht. Der vorgegebene Weg kann vollständig verfahren werden. |
| 540 | 2222 | 0.000 | -25.000 | Wegen _LIMIT[1] = 2 wird der Verfahrweg durch die Arbeitsfeldbegrenzung eingeschränkt. |

| N... | <Status> | <MaxDist>[0] ≙ X | <MaxDist>[1] ≙ Y | Bemerkungen |
|---|---|---|---|---|
| 620 | 4223 | -13.000 | 0.000 | Abstand zu C4 wegen C2 und _LIMIT[3] insgesamt 4 mm. Abstand C2 → N3 von 0,1 mm führt nicht zur Beschränkung des Verfahrwegs. |
| 710 | 1221 | 0.000 | 21.213 | Frame mit Translation und Rotation aktiv. Der zulässige Verfahrweg in _DIST gilt im verschobenen und gedrehten WKS. |
| 820 | 102121 | 18.000 | 18.000 | Der Software-Endschalter der Y-Achse wird verletzt. Mit <_TESTLIM> = 17 wird die Berechnung einer Weiterfahrrichtung angefordert. Diese Richtung steht in <_DIST> (0.707, 0.0, 0.707). Sie ist gültig, da in <_STATUS> die Einhunderttausenderstelle gesetzt ist. |

## Weitere Informationen

### Achsstatus "Referenziert"

Alle von `CALCPOSI()` betrachteten Maschinenachsen müssen referenziert sein.

### Kreisbezogene Wegangaben

Alle kreisbezogenen Wegangaben werden **immer** als Radiusangabe interpretiert. Dies ist insbesondere bei Planachsen mit aktivierter Durchmesserprogrammierung (`DIAMON` / `DIAM90`) zu berücksichtigen.

### Verfahrwegreduzierung

Wird der angegebene Verfahrweg einer Achsen begrenzt, sind im Rückgabewert `<MaxDist>` auch die Verfahrwege der anderen Achsen anteilsmäßig reduziert. Der resultierende Endpunkt liegt dadurch weiterhin auf der vorgegebenen Bahn.

### Rundachsen

Rundachsen werden nur überwacht, wenn es keine Modulo-Rundachsen sind.

Es ist zulässig, dass für eine oder mehrere der beteiligten Achsen keine Software-Endschalter bzw. Arbeitsfeldbegrenzungen oder Schutzbereiche definiert sind.

### Software-Endschalter- und Arbeitsfeldbegrenzungs-Status

Software-Endschalter und Arbeitsfeldbegrenzungen werden nur berücksichtigt, wenn diese bei Ausführung von `CALCPOSI()` aktiv sind. Der Status kann z. B. beeinflusst werden über:

- Maschinendaten: MD21020 $MC_WORKAREA_WITH_TOOL_RADIUS
- Settingdaten: $AC_WORKAREA_CS_...
- NC/PLC-Nahtstellensignale: DB31, ... DBX12.2 / 3
- Befehle: `WALIMON` / `WALIMOF`

### Software-Endschalter und Transformationen

Bei `CALCPOSI()` können bei verschiedenen kinematischen Transformationen (z. B. `TRANSMIT`) aufgrund von Mehrdeutigkeiten an bestimmten Stellen des Verfahrbereichs die

Positionen der Maschinenachsen (MKS) nicht immer eindeutig aus den Positionen der Geometrieachsen (WKS) bestimmt werden. Im normalen Verfahrbetrieb ergibt sich die Eindeutigkeit in der Regel aus der Vorgeschichte und der Bedingung, dass einer kontinuierlichen Bewegung im WKS eine kontinuierliche Bewegung der Maschinenachsen im MKS entsprechen muss. Bei der Überwachung der Software-Endschalter wird deshalb in derartigen Fällen die zum Ausführungszeitpunkt von `CALCPOSI()` vorliegende Maschinenposition zur Auflösung der Mehrdeutigkeit verwendet.

**Hinweis**

**Vorlaufstopp**

Bei Verwendung von `CALCPOSI()` im Zusammenhang mit Transformationen liegt es in der alleinigen Verantwortung des Anwenders, zur Synchronisation der Maschinenachspositionen mit dem Vorlauf vor `CALCPOSI()` einen Vorlaufstopp (`STOPRE`) zu programmieren.

### Schutzbereichsabstand und konventionelle Schutzbereiche

Bei konventionellen Schutzbereichen ist **nicht** gewährleistet, dass bei einer Verfahrbewegung auf dem vorgegebenen Verfahrweg der im Parameter `<Limit>[3]` angegebene Sicherheitsabstand gegenüber allen Schutzbereichen eingehalten wird. Es ist nur gewährleistet, dass bei Verlängerung des in `<Dist>` zurückgelieferten Endpunkts um den Sicherheitsabstand in Verfahrrichtung kein Schutzbereich verletzt wird. Die Gerade kann aber in ihrem Verlauf beliebig dicht an einem Schutzbereich vorbei führen.

### Schutzbereichsabstand und Schutzbereiche der Kollisionsvermeidung

Bei Schutzbereichen der Kollisionsvermeidung ist gewährleistet, dass bei einer Verfahrbewegung auf dem vorgegebenen Verfahrweg, der im Parameter `<Limit>[3]` angegebene Sicherheitsabstand gegenüber allen Schutzbereichen eingehalten wird.

Der im Parameter `<Limit>[3]` angegebene Sicherheitsabstand wird nur wirksam, wenn gilt:

`<Limit>[3]` **>** (MD10619 $MN_COLLISION_TOLERANCE)

Ist in Parameter `<TestLim>` Bit 4 gesetzt (Berechnung der weiteren Verfahrrichtung), ist der in `<DIST>` enthaltene Richtungsvektor nur dann gültig, wenn die Einhunderttausenderstelle im Rückgabewert der Funktion (`<Status>`) gesetzt ist. Kann eine solche Richtung nicht ermittelt werden, weil entweder Schutzbereiche verletzt wurden oder eine Transformation aktiv ist, bleibt der Eingangswert in `<DIST>` unverändert. Eine weitere Fehlermeldung erfolgt nicht.

# 3.7 Spezielle Wegbefehle

## 3.7.1 Codierte Positionen anfahren (CAC, CIC, CDC, CACP, CACN)

Mit den Wegbefehlen zum "Anfahren von codierten Positionen" ist es möglich, Linear- und Rundachsen unter Angabe von Positionsnummern auf feste, in Maschinendaten-Tabellen hinterlegte Achspositionen zu verfahren.

### Syntax

```
CAC(<n>)
CIC(<n>)
CACP(<n>)
CACN(<n>)
```

### Bedeutung

| | |
|---|---|
| `CAC(<n>)`: | Codierte Position von Positionsnummer n anfahren |
| `CIC(<n>)`: | Codierte Position, ausgehend von der aktuellen Positionsnummer, n-Positionsplätze vor (+n) oder zurück (–n) anfahren |
| `CDC(<n>)`: | Codierte Position von Positionsnummer n auf kürzestem Weg anfahren (nur für Rundachsen) |
| `CACP(<n>)`: | Codierte Position von Positionsnummer n in positiver Richtung anfahren (nur für Rundachsen) |
| `CACN(<n>)`: | Codierte Position von Positionsnummer n in negativer Richtung anfahren (nur für Rundachsen) |
| `<n>`: | Positionsnummer innerhalb der Maschinendaten-Tabelle<br>Wertebereich: 0, 1, … (max. Anzahl Tabellenplätze - 1) |

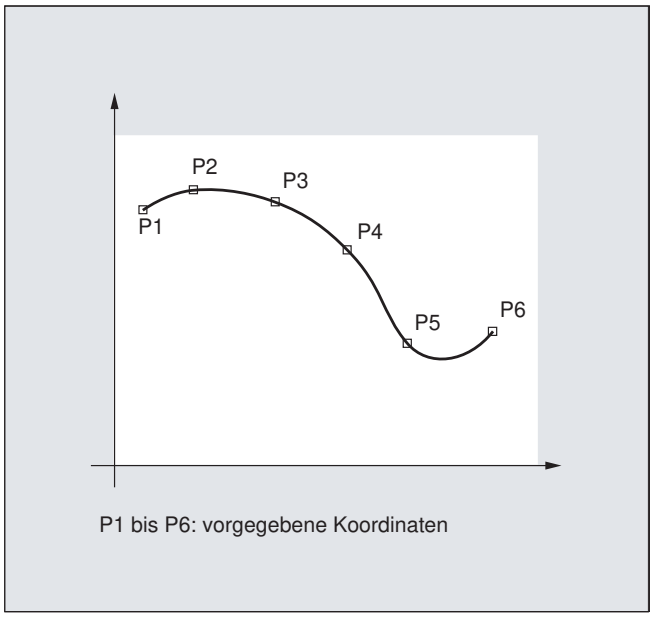
### Beispiel: Anfahren von codierten Positionen einer Positionierachse

| **Programmiercode** | **Kommentar** |
|---|---|
| `N10 FA[B]=300` | `; Vorschub für Positionierachse B` |
| `N20 POS[B]=CAC(10)` | `; Codierte Position von Positionsnummer 10 anfahren` |
| `N30 POS[B]=CIC(-4)` | `; Codierte Position von "aktuelle Positionsnummer" - 4 anfahren` |

## 3.7.2 Spline-Interpolation (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

Beliebig gekrümmte Konturen an Werkstücken können nicht analytisch exakt beschrieben werden. Derartige Konturen werden daher durch eine begrenzte Anzahl von Stützpunkten, z. B. beim Digitalisieren von Oberflächen, angenähert. Zur Erzeugung der digitalisierten Oberfläche an einem Werkstück müssen die Stützpunkte zu einer Konturbeschreibung verbunden werden. Dies ermöglicht die Spline-Interpolation.

Ein Spline definiert eine Kurve, die aus Polynomen 2. oder 3. Grades zusammengesetzt wird. Die Eigenschaften an den Stützpunkten eines Splines sind **abhängig vom verwendeten Spline-Typ** definierbar.




Folgende Spline-Typen stehen bei SINUMERIK solution line zur Verfügung:

- A-Spline
- B-Spline
- C-Spline

## Syntax

Allgemein:

```
ASPLINE X... Y... Z... A... B... C...
BSPLINE X... Y... Z... A... B... C...
CSPLINE X... Y... Z... A... B... C...
```

Bei B-Spline zusätzlich programmierbar:

```
PW=<n>
SD=2
PL=<Wert>
```

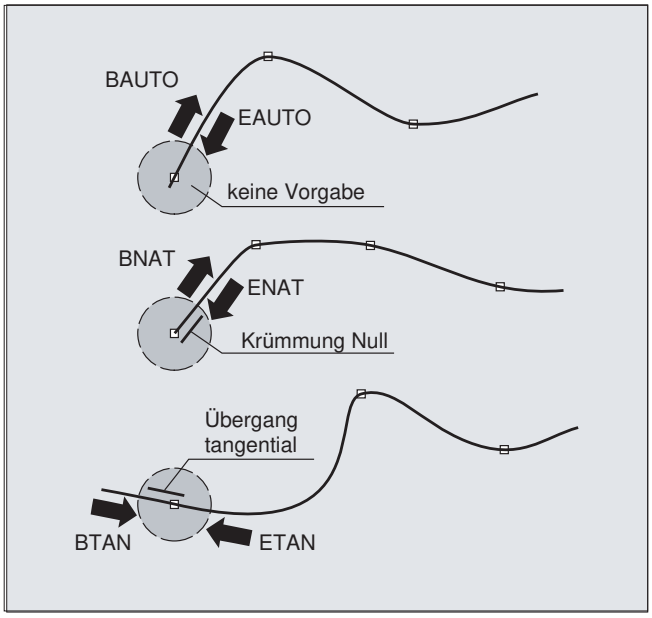
Bei A- und C-Spline zusätzlich programmierbar:

```
BAUTO / BNAT / BTAN
EAUTO / ENAT / ETAN
```

## Bedeutung

| **Spline-Interpolationstyp:** | |
|---|---|
| `ASPLINE`: | Befehl zum Einschalten der A-Spline-Interpolation |

| | | | |
|---|---|---|---|
| BSPLINE: | Befehl zum Einschalten der B-Spline-Interpolation | | |
| CSPLINE: | Befehl zum Einschalten der C-Spline-Interpolation | | |
| | Die Befehle ASPLINE, BSPLINE und CSPLINE sind modal wirksam und gehören zur Gruppe der Wegbefehle. | | |
| **Stützpunkte bzw. Kontrollpunkte:** | | | |
| X... Y... Z...<br>A... B... C... | Positionen in kartesischen Koordinaten | | |
| **Punktgewicht (nur B-Spline):** | | | |
| PW: | Mit dem Befehl PW ist für jeden Stützpunkt die Programmierung eines sogenannten "Punktgewichts" möglich. | | |
| <n>: | "Punktgewicht" | | |
| | Wertebereich: | 0 ≤ n ≤ 3 | |
| | Schrittweite: | 0.0001 | |
| | Wirkung: | n > 1 | Die Kurve wird vom Kontrollpunkt stärker angezogen. |
| | | n < 1 | Die Kurve wird vom Kontrollpunkt weniger stark angezogen. |
| **Spline-Grad (nur B-Spline):** | | | |
| SD: | Standardmäßig wird ein Polygon 3. Grades verwendet. Durch Programmierung von SD=2 kann aber auch ein Polygon 2. Grades verwendet werden. | | |
| **Knotenabstand (nur B-Spline):** | | | |
| PL: | Die Knotenabstände werden intern geeignet berechnet. Die Steuerung kann aber auch vorgegebene Knotenabstände verarbeiten, die mit dem Befehl PL als sog. Parameter-Intervall-Länge angegeben werden. | | |
| <Wert>: | Parameter-Intervall-Länge | | |
| | Wertebereich: | wie Wegmaß | |
| **Übergangsverhalten am Beginn der Spline-Kurve (nur A- oder C-Spline):** | | | |
| BAUTO: | Keine Vorgabe für das Übergangsverhalten. Der Anfang ergibt sich aus der Lage des ersten Punkts. | | |
| BNAT: | Krümmung Null | | |
| BTAN: | Tangentialer Übergang zum Satz vorher (Löschstellung) | | |
| **Übergangsverhalten am Ende der Spline-Kurve (nur A- oder C-Spline):** | | | |
| EAUTO: | Keine Vorgabe für das Übergangsverhalten. Das Ende ergibt sich aus der Lage des letzten Punkts. | | |
| ENAT: | Krümmung Null | | |

| ETAN: | Tangentialer Übergang zum Satz vorher (Löschstellung) |
|---|---|
| |  |

**Hinweis**

Das programmierbare Übergangsverhalten hat keinen Einfluss auf den B-Spline. Der B-Spline ist in Start- und Endpunkt immer tangential zum Kontrollpolygon.

## Randbedingungen

- Die Werkzeugradiuskorrektur ist einsetzbar.
- Kollisionsüberwachung erfolgt in der Projektion auf die Ebene.

## Beispiele

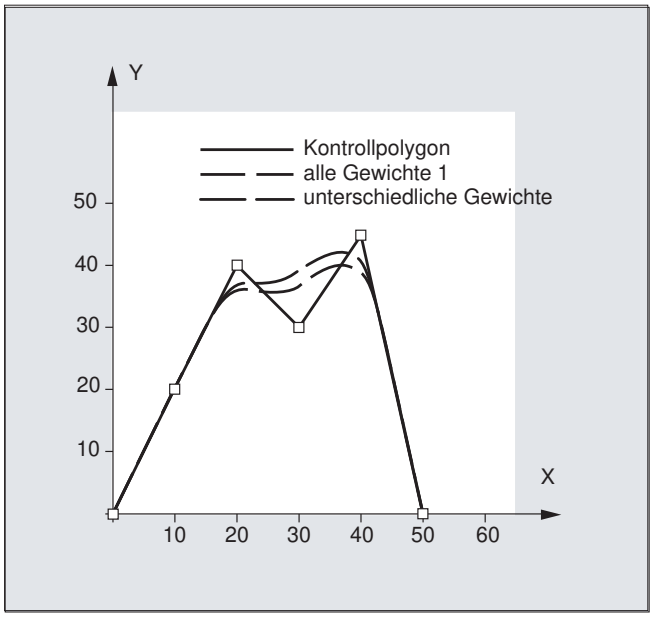
### Beispiel 1: B-Spline

```
Programmcode 1 (alle Gewichte 1)
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
N60 X40 Y45
N70 X50 Y0
```

**Programmcode 2 (unterschiedliche Gewichte)**

```
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20 PW=2
N40 X20 Y40
N50 X30 Y30 PW=0.5
N60 X40 Y45
N70 X50 Y0
```

| **Programmcode 3 (Kontrollpolygon)** | **Kommentar** |
|---|---|
| N10 G1 X0 Y0 F300 G64 | |
| N20 | ; entfällt |
| N30 X10 Y20 | |
| N40 X20 Y40 | |
| N50 X30 Y30 | |
| N60 X40 Y45 | |
| N70 X50 Y0 | |

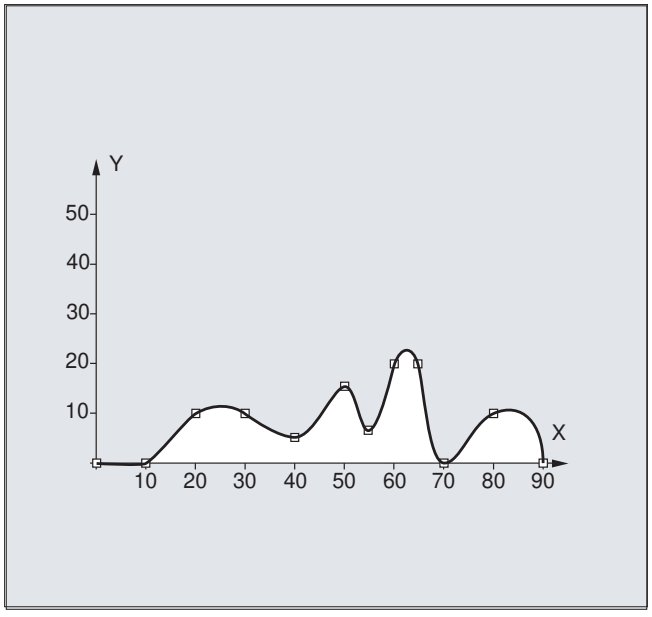



### Beispiel 2: C-Spline, am Anfang und am Ende Krümmung Null

**Programmcode**

```
N10 G1 X0 Y0 F300
N15 X10
N20 BNAT ENAT
N30 CSPLINE X20 Y10
```

```
Programmcode
N40 X30
N50 X40 Y5
N60 X50 Y15
N70 X55 Y7
N80 X60 Y20
N90 X65 Y20
N100 X70 Y0
N110 X80 Y10
N120 X90 Y0
N130 M30
```




### Beispiel 3: Spline-Interpolation (A-Spline) und Koordinatentransformation (ROT)

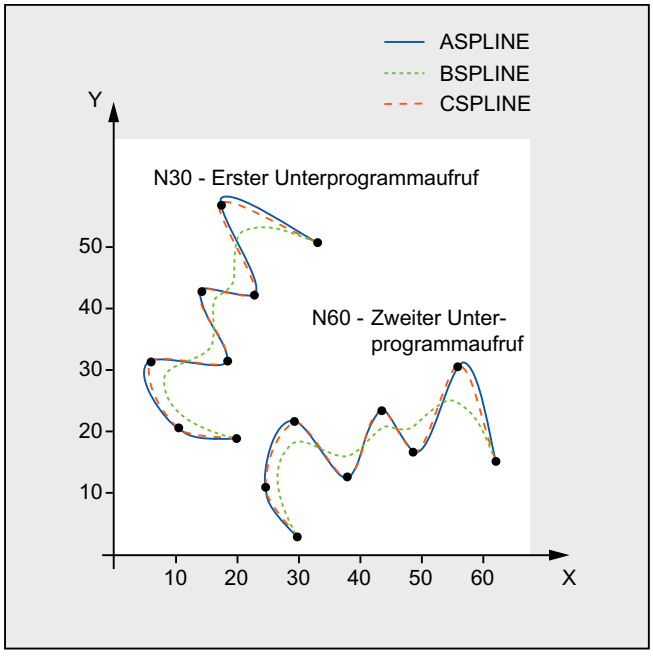
Hauptprogramm:

| Programmcode | Kommentar |
|---|---|
| N10 G00 X20 Y18 F300 G64 | ; Startpunkt anfahren. |
| N20 ASPLINE | ; Interpolationstyp A-Spline aktivieren. |
| N30 KONTUR | ; Erster Aufruf des Unterprogramms. |
| N40 ROT Z-45 | ; Koordinatentransformation: Drehung des WKS um -45° um die Z-Achse. |
| N50 G00 X20 Y18 | ; Konturstartpunkt anfahren. |
| N60 KONTUR | ; Zweiter Aufruf des Unterprogramms. |
| N70 M30 | ; Programmende |

Unterprogramm "Kontur" (enthält die Stützpunkt-Koordinaten):

```
Programmcode
N10 X20 Y18
N20 X10 Y21
N30 X6 Y31
N40 X18 Y31
N50 X13 Y43
N60 X22 Y42
N70 X16 Y58
N80 X33 Y51
N90 M1
```

In der folgenden Abbildung sind neben der Spline-Kurve, die aus dem Programmbeispiel resultiert (ASPLINE), auch die Spline-Kurven enthalten, die sich bei Aktivierung einer B- oder C-Spline-Interpolation ergeben hätten (BSPLINE, CSPLINE):



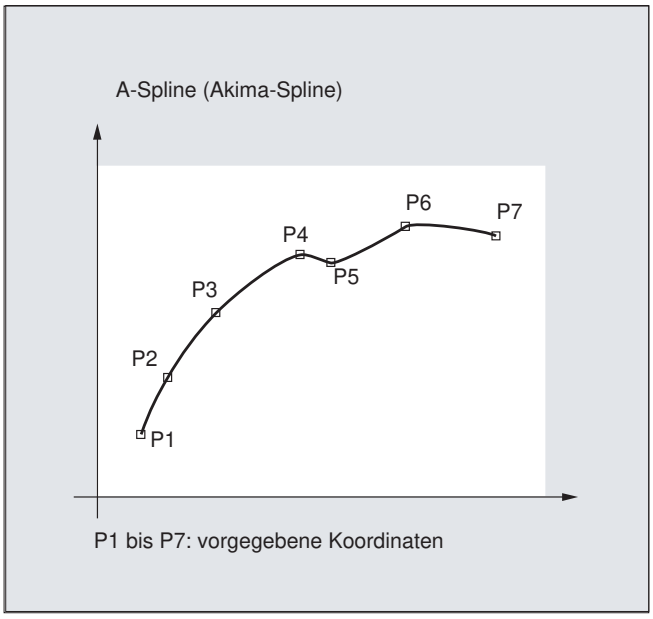

## Weitere Informationen

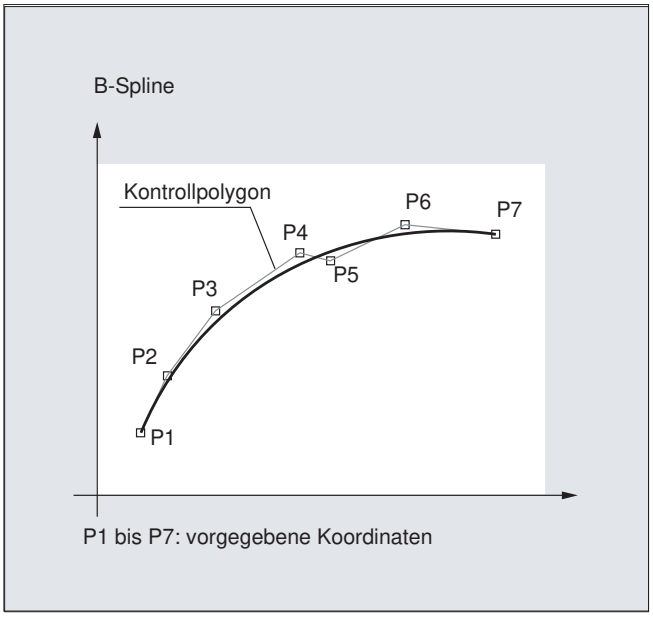
### Vorteile der Spline-Interpolation

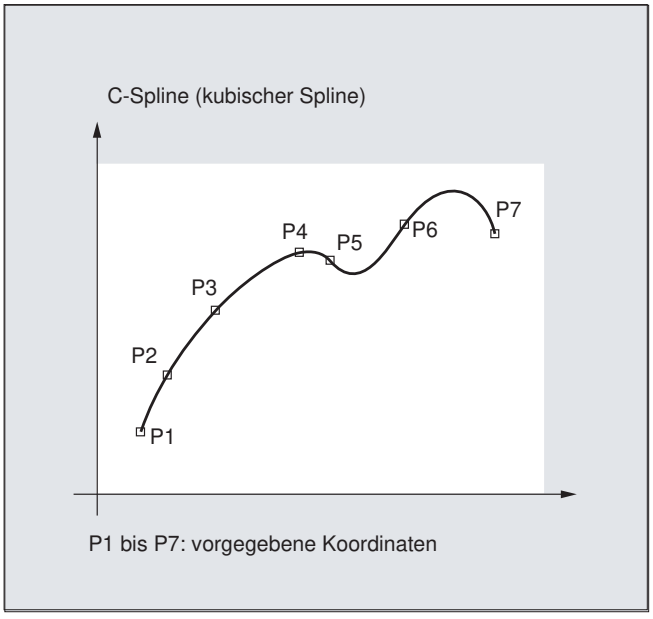
Durch Verwendung der Spline-Interpolation lassen sich, im Gegensatz zur Verwendung von Geradensätzen G01, folgende Vorteile erzielen:

- Reduzierung der Anzahl von benötigten Teileprogrammsätzen zur Beschreibung der Kontur
- Weicher, mechanikschonender Kurvenverlauf beim Übergange zwischen den Teileprogrammsätzen

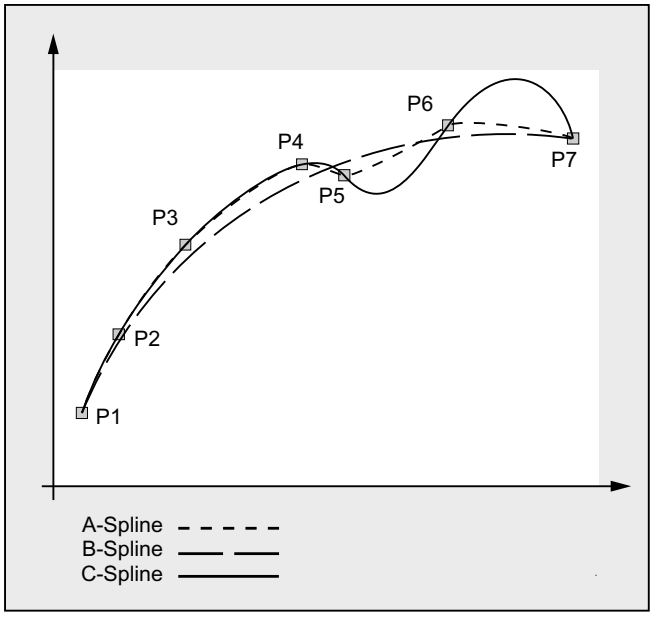
### Eigenschaften und Anwendung der verschiedenen Spline-Typen

| Spline-Typ | Eigenschaften und Anwendung |
|---|---|
| **A-Spline** |  <br>**Eigenschaften:**<br>• Verläuft exakt durch die vorgegebenen Stützpunkte.<br>• Der Kurvenverlauf ist tangenten- aber nicht krümmungsstetig.<br>• Erzeugt kaum ungewollte Schwingungen.<br>• Der Einflussbereich von Stützpunktänderungen ist lokal, d. h. Veränderung eines Stützpunkts wirkt sich nur auf bis zu max. 6 benachbarte Stützpunkte aus.<br><br>**Anwendung:**<br>Der A-Spline eignet sich vor allem für die Interpolation von Kurvenverläufen mit großen Steigungsänderungen (z. B. treppenförmige Kurvenverläufe). |

| Spline-Typ | Eigenschaften und Anwendung |
|---|---|
| **B-Spline** |  <br>**Eigenschaften:**<br>• Verläuft nicht durch die vorgegebenen Stützpunkte, sondern nur in deren Nähe. Die Kurve wird durch die Stützpunkte angezogen. Durch Gewichtung der Stützpunkte mit einem Faktor, kann der Kurvenverlauf zusätzlich beeinflusst werden.<br>• Der Kurvenverlauf ist tangenten- und krümmungsstetig.<br>• Erzeugt keine ungewollten Schwingungen.<br>• Der Einflussbereich von Stützpunktänderungen ist lokal, d. h. Veränderung eines Stützpunkts wirkt sich nur auf bis zu max. 6 benachbarte Stützpunkte aus.<br><br>**Anwendung:**<br>Der B-Spline ist primär als Schnittstelle zu CAD-Systemen gedacht. |

| Spline-Typ | Eigenschaften und Anwendung |
|---|---|
| **C-Spline** |  <br>**Eigenschaften:**<br>• Verläuft exakt durch die vorgegebenen Stützpunkte.<br>• Der Kurvenverlauf ist tangenten- und krümmungsstetig.<br>• Erzeugt häufig ungewollten Schwingungen, besonders an Stellen mit großen Steigungsänderungen.<br>• Der Einflussbereich von Stützpunktänderungen ist global, d. h. Veränderung eines Stützpunkts wirkt sich auf den gesamten Kurvenverlauf aus.<br><br>**Anwendung:**<br>Der C-Spline kann dann gut eingesetzt werden, wenn die Stützpunkte auf einer analytisch bekannten Kurve liegen (Kreis, Parabel, Hyperbel) |

**Gegenüberstellung der drei Spline-Typen bei gleichen Stützpunkten**




**Mindestanzahl an Spline-Sätzen**

Die G-Befehle ASPLINE, BSPLINE und CSPLINE verbinden Satzendpunkte mit Splines. Dazu müssen im Vorlauf eine Reihe von Sätzen (Endpunkte) gleichzeitig berechnet werden. Die Größe des Puffers für die Berechnung beträgt standardmäßig 10 Sätze. Nicht jede Satzinformation ist ein Spline-Endpunkt. Die Steuerung benötigt jedoch von 10 Sätzen eine bestimmte Anzahl an Spline-Endpunkt-Sätzen:

| Spline-Typ | Mindestanzahl an Spline-Sätzen |
|---|---|
| A-Spline: | Von je 10 Sätzen müssen mindestens **4** Spline-Sätze sein.<br>Kommentarsätze und Parameterrechnungen zählen hierbei nicht. |
| B-Spline: | Von je 10 Sätzen müssen mindestens **6** Spline-Sätze sein.<br>Kommentarsätze und Parameterrechnungen zählen hierbei nicht. |
| C-Spline: | Die benötigte Mindestanzahl an Spline-Sätzen ergibt sich aus folgender Summe:<br>Wert aus MD20160 $MC_CUBIC_SPLINE_BLOCKS + 1<br>Im MD20160 wird die Anzahl der Punkte eingetragen, über die der Spline-Abschnitt berechnet wird. Die Standardeinstellung beträgt 8. Von je 10 Sätzen müssen daher im Standardfall mindestens **9** Spline-Sätze sein. |

**Hinweis**

Bei Unterschreitung des tolerierbaren Werts wird ein Alarm ausgegeben, ebenso, wenn eine am Spline beteiligte Achse als Positionierachse programmiert wird.

**Zusammenfassung kurzer Spline-Sätze**

Bei der Spline-Interpolation können kurze Spline-Sätze entstehen, die zu einer unnötigen Reduzierung der Bahngeschwindigkeit führen. Mit der Funktion "Zusammenfassung kurzer Spline-Sätze" können diese Sätze so zusammengefasst werden, dass die resultierende Satzlänge ausreichend groß ist und nicht zu einer Verringerung der Bahngeschwindigkeit führt.

Die Funktion wird aktiviert über das kanalspezifische Maschinendatum:

MD20488 $MC_SPLINE_MODE (Einstellung für Spline-Interpolation)

**Weitere Informationen:** Funktionshandbuch Basisfunktionen

## 3.7.3 Spline-Verbund (SPLINEPATH)

Die im Spline-Verbund zu interpolierenden Achsen werden mit dem Befehl `SPLINEPATH` ausgewählt. Bis zu acht Bahnachsen sind bei der Spline-Interpolation möglich.

**Hinweis**

Wird `SPLINEPATH` nicht explizit programmiert, so werden die ersten drei Achsen des Kanals als Spline-Verbund verfahren.

### Syntax

Die Festlegung des Spline-Verbundes erfolgt in einem gesonderten Satz:

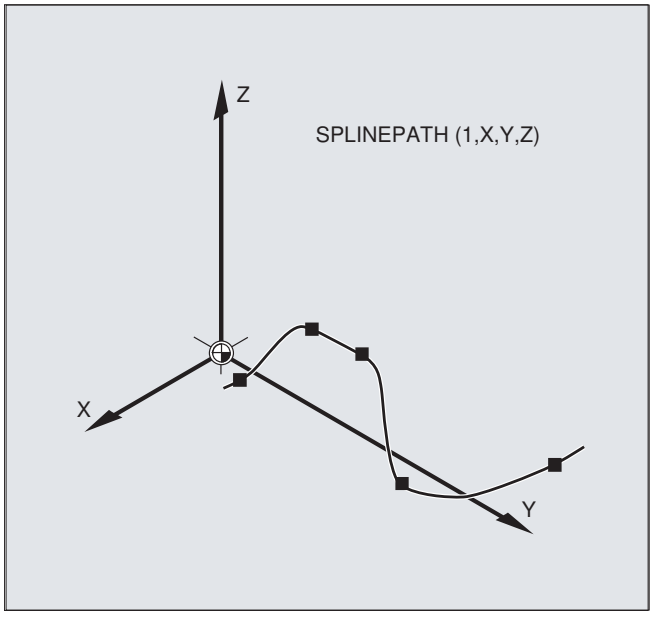
```
SPLINEPATH(n,X,Y,Z,…)
```

### Bedeutung

| | |
|---|---|
| `SPLINEPATH`: | Befehl zur Festlegung eines Spline-Verbundes |
| `n`: | =1 (fester Wert) |
| `X,Y,Z,…` : | Bezeichner der im Spline-Verbund zu interpolierenden Bahnachsen |

### Beispiel: Spline-Verbund mit drei Bahnachsen

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 G1 X10 Y20 Z30 A40 B50 F350` | |
| `N11 SPLINEPATH(1,X,Y,Z)` | `; Spline-Verbund` |
| `N13 CSPLINE BAUTO EAUTO X20 Y30 Z40 A50 B60` | `; C-Spline` |
| `N14 X30 Y40 Z50 A60 B70` | `; Stützpunkte` |
| … | |
| `N100 G1 X… Y…` | `; Abwahl Spline-Interpolation` |




## 3.7.4 NC-Satz-Kompression ein-/ausschalten (COMPON, COMPCURV, COMPCAD, COMPSURF, COMPOF)

Die Funktionen zur Kompression von Linearsätzen (und abhängig von der Parametrierung auch Kreis- und/oder Eilgangsätzen) werden mit den G-Befehlen der G-Gruppe 30 ein-/ausgeschaltet. Die Befehle sind modal wirksam.

### Syntax

```
COMPON / COMPCURV / COMPCAD / COMPSURF
...
COMPOF
```

### Bedeutung

| | |
|---|---|
| COMPON: | Einschalten der Kompressor-Funktion COMPON |
| COMPCURV: | Einschalten der Kompressor-Funktion COMPCURV |
| COMPCAD: | Einschalten der Kompressor-Funktion COMPCAD |
| COMPSURF: | Einschalten der Kompressor-Funktion COMPSURF |
| COMPOF: | Ausschalten der aktuell aktiven Kompressor-Funktion |

**Hinweis**

Zur zusätzlichen Verbesserung der Oberflächengüte kann die Überschleiffunktion G642 und die Ruckbegrenzung SOFT verwendet werden. Diese Befehle sind am Programmanfang zu schreiben.

### Beispiel: COMPCAD

```
Programmcode                    Kommentar
N10 G00 X30 Y6 Z40
N20 G1 F10000 G642              ; Einschalten: Überschleiffunktion G642
N30 SOFT                        ; Einschalten: Ruckbegrenzung SOFT
N40 COMPCAD                     ; Einschalten: Kompressor-Funktion COMPCAD
N50 FIFOCTRL
N24050 Z32.499                  ; 1. Verfahrsatz
N24051 X41.365 Z32.500          ; 2. Verfahrsatz
...
N99999 X... Z...                ; letzter Verfahrsatz
COMPOF                          ; Kompressor-Funktion aus.
...
```

## 3.7.5 Polynom-Interpolation (POLY, POLYPATH, PO, PL)

Im eigentlichen Sinn handelt es sich bei der Polynom-Interpolation (`POLY`) nicht um eine Spline-Interpolationsart. Sie ist in erster Linie als Schnittstelle für die Programmierung extern erzeugter Spline-Kurven gedacht. Hierbei können die Spline-Abschnitte direkt programmiert werden.

Diese Interpolationsart entlastet die NC von der Berechnung der Polynom-Koeffizienten. Sie ist dann optimal einsetzbar, wenn die Koeffizienten direkt von einem CAD-System oder Post-Prozessor kommen.

### Syntax

Polynom 3. Grades:

```
POLY PO[X]=(xe,a2,a3) PO[Y]=(ye,b2,b3) PO[Z]=(ze,c2,c3) PL=n
```

Polynome 5.Grades und neue Polynomsyntax:

```
POLY X=PO(xe,a2,a3,a4,a5) Y=PO(ye,b2,b3,b4,b5) Z=PO(ze,c2,c3,c4,c5)
PL=n
POLYPATH("AXES","VECT")
```

**Hinweis**

Die Summe der in einem NC-Satz programmierten Polynom-Koeffizienten und Achsen darf die maximal erlaubte Achsanzahl pro Satz nicht überschreiten.

### Bedeutung

| | |
|---|---|
| `POLY` : | Einschalten der Polynom-Interpolation mit einem Satz mit POLY. |
| `POLYPATH` : | Polynom-Interpolation selektierbar für die beiden Achsgruppen AXIS oder VECT |

| | |
|---|---|
| `PO[Achsbezeichner/Variable]:` | Endpunkte und Polynom-Koeffizienten |
| `X, Y, Z:` | Achsbezeichner |
| `xe, ye, ze :` | Angabe der Endposition für die jeweilige Achse; Wertebereich wie Wegmaß |
| `a2, a3, a4, a5:` | Die Koeffizienten $a_2$, $a_3$, $a_4$, und $a_5$ werden mit ihrem Wert geschrieben; Wertebereich wie Wegmaß. Der jeweils letzte Koeffizient kann entfallen, wenn er den Wert Null hat. |
| `PL:` | Länge des Parameterintervalls, auf dem die Polynome definiert sind (Definitionsbereich der Funktion f(p)).<br>Das Intervall beginnt immer bei 0, p kann Werte von 0 bis PL annehmen.<br>Theoretischer Wertebereich für PL:<br>0,0001 … 99 999,9999<br>**Hinweis:**<br>Der PL-Wert gilt für den Satz, in dem er steht. Ist kein PL programmiert, wirkt PL=1. |

## Ein-/Ausschalten der Polynom-Interpolation

Die Polynom-Interpolation wird im Teileprogramm durch den G-Befehl `POLY` eingeschaltet.

Der G-Befehl `POLY` gehört zusammen mit `G0`, `G1`, `G2`, `G3`, `ASPLINE`, `BSPLINE` und `CSPLINE` zur 1. G-Gruppe.

Achsen, die nur mit Namen und Endpunkt programmiert sind (z.B. `X10`), werden linear verfahren. Sind alle Achsen eines NC-Satzes so programmiert, verhält sich die Steuerung wie bei G1.

Die Polynom-Interpolation wird durch die Programmierung eines anderen Befehls der 1. G-Gruppe (z. B. `G0`, `G1`) implizit wieder ausgeschaltet.

## Polynomkoeffizient

Der `PO`-Wert `(PO[]=)` bzw. `...=PO(...)` gibt alle Polynom-Koeffizienten für eine Achse an. Entsprechend dem Grad des Polynoms werden mehrere Werte durch Kommata getrennt angegeben. Innerhalb eines Satzes sind unterschiedliche Polynomgrade für verschiedene Achsen möglich.

## Unterprogramm POLYPATH

Mit `POLYPATH(...)` kann die Polynom-Interpolation selektiv für bestimmte Achsgruppen freigegeben werden:

| | |
|---|---|
| Nur Bahnachsen und Zusatzachsen: | `POLYPATH("AXES")` |
| Nur Orientierungsachsen:<br>(beim Verfahren mit Orientierungs-Transformation) | `POLYPATH("VECT")` |

Die jeweils nicht freigegebenen Achsen werden linear verfahren.

Standardmäßig ist die Polynom-Interpolation für beide Achsgruppen freigegeben.

Durch Programmierung ohne Parameter `POLYPATH( )` wird die Polynom-Interpolation für alle Achsen deaktiviert.

## Beispiel

| **Programmcode** | **Kommentar** |
|---|---|
| `N10 G1 X… Y… Z… F600` | |
| `N11 POLY PO[X]=(1,2.5,0.7) PO[Y]=(0.3,1,3.2) PL=1.5` | `; Polynom-Interpolation ein` |
| `N12 PO[X]=(0,2.5,1.7) PO[Y]=(2.3,1.7) PL=3` | |
| `...` | |
| `N20 M8 H126 …` | |
| `N25 X70 PO[Y]=(9.3,1,7.67) PL=5` | `; gemischte Angaben für die Achsen` |
| `N27 PO[X]=(10,2.5) PO[Y]=(2.3)` | `; kein PL programmiert; es wirkt PL=1` |
| `N30 G1 X… Y… Z.` | `; Polynom-Interpolation aus` |
| `…` | |

## Beispiel: Neue Polynomsyntax

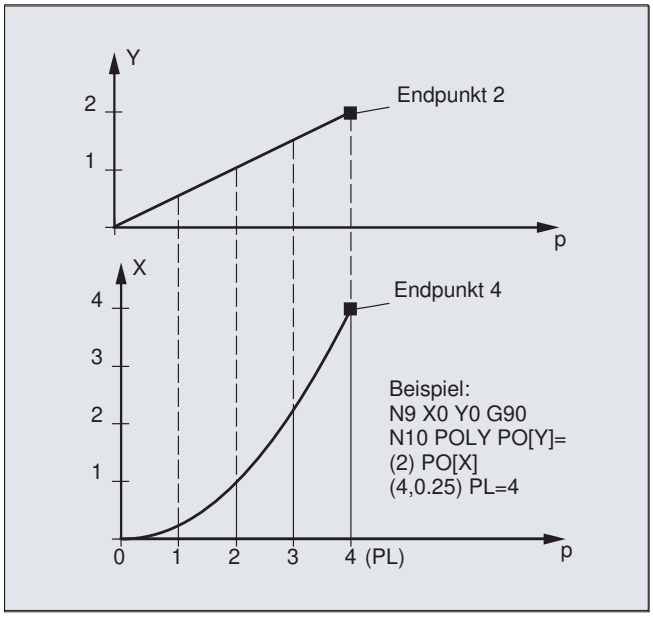
| Weiterhin gültige Polynomsyntax | Neue Polynomsyntax |
|---|---|
| `PO[Achsbezeichner]=(.. , ..)` | `Achsbezeichner=PO(.. , ..)` |
| `PO[PHI]=(.. , ..)` | `PHI=PO(.. , ..)` |
| `PO[PSI]=(.. , ..)` | `PSI=PO(.. , ..)` |
| `PO[THT]=(.. , ..)` | `THT=PO(.. , ..)` |
| `PO[]=(.. , ..)` | `PO(.. , ..)` |
| `PO[variable]=IC(.. , ..)` | `variable=PO IC(.. , ..)` |

## Beispiel: Kurve in der X/Y-Ebene

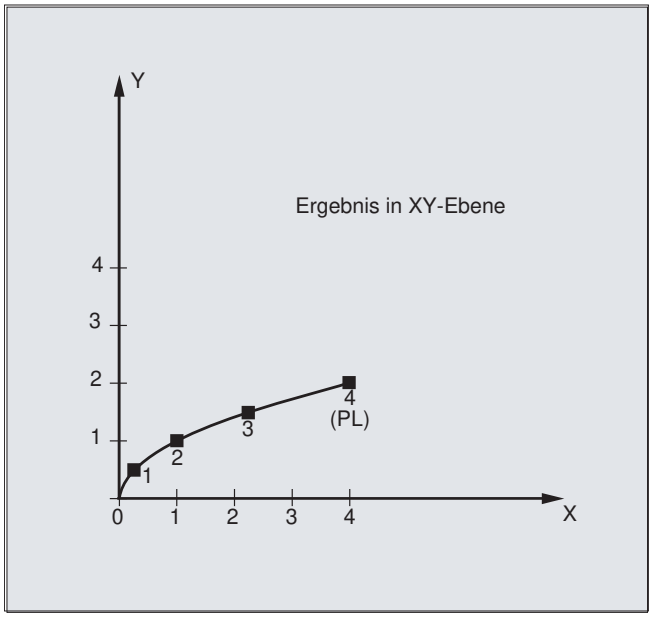
Programmierung

| **Programmcode** |
|---|
| `N9 X0 Y0 G90 F100` |
| `N10 POLY PO[Y]=(2) PO[X]=(4,0.25) PL=4` |

Verlauf der Kurven X(p) und Y(p)




Verlauf der Kurve in der XY-Ebene




## Beschreibung

Die allgemeine Form der Polynom-Funktion lautet:

$f(p)= a_0 + a_1p + a_2p^2 + . . . + a_np^n$

mit: $a_i$: konstante Koeffizienten (i = 0, 1, ..., n)
p: Parameter

In der Steuerung können maximal Polynome 5. Grades programmiert werden:

$$f(p)= a_0 + a_1p + a_2p^2 + a_3p^3 + a_4p^4 + a_5p^5$$

Durch Belegen der Koeffizienten mit konkreten Werten sind verschiedene Kurvenverläufe, wie Geraden, Parabeln und Potenzfunktionen, erzeugbar.

Eine Gerade wird erzeugt durch $a_2 = a_3 = a_4 = a_5 = 0$:

$$f(p)= a_0 + a_1p$$

Weiter gilt:

$a_{0:}$ Achsposition am Ende des vorangehenden Satzes
$p = PL$
$a_1 = (x_E - a_0 - a_2*p^2 - a_3*p^3) / p$

Es ist möglich Polynome zu programmieren, **ohne** dass die Polynom-Interpolation durch den G-Befehl `POLY` aktiviert wurde. In diesem Fall werden nicht die programmierten Polynome interpoliert, sondern die programmierten Endpunkte der Achsen linear angefahren (`G1`). Erst nach expliziter Aktivierung der Polynom-Interpolation im Teileprogramm (`POLY`) werden die programmierten Polynome auch als solche verfahren.

## Besonderheit: Nenner-Polynom

Für die Geometrieachsen kann mit `PO[]=(…)` ohne Angabe eines Achsnamens auch ein gemeinsames Nenner-Polynom programmiert werden, d. h. die Bewegung der Geometrieachsen wird als Quotient zweier Polynome interpoliert.

Damit lassen sich z. B. Kegelschnitte (Kreis, Ellipse, Parabel, Hyperbel) exakt darstellen.

**Beispiel:**

| **Programmcode** | **Kommentar** |
|---|---|
| `POLY G90 X10 Y0 F100` | `; Geometrieachsen verfahren linear auf die Position X10 Y0.` |
| `PO[X]=(0,-10) PO[Y]=(10) PO[]=(2,1)` | `; Geometrieachsen verfahren im Viertelkreis auf X0 Y10.` |

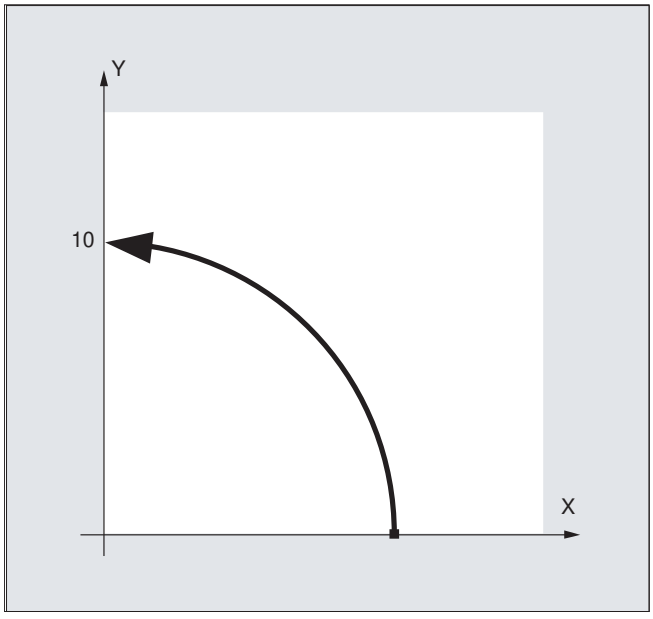
Der konstante Koeffizient ($a_0$) des Nenner-Polynoms wird stets mit 1 angenommen. Der programmierte Endpunkt ist unabhängig von `G90` / `G91`.

Aus den programmierten Werten berechnen sich X(p) und Y(p) zu:

$X(p) = (10 - 10 * p^2) / (1 + p^2)$
$Y(p) = 20 * p / (1 + p^2)$
mit $0 \leq p \leq 1$

Aufgrund der programmierten Anfangspunkte, Endpunkte, Koeffizient $a_2$ und PL=1 ergeben sich folgende Zwischenergebnisse:

Zähler (X) = $10 + 0 * p - 10 * p^2$
Zähler (Y) = $0 + 20 * p + 0 * p^2$
Nenner = $1 + p^2$




Bei eingeschalteter Polynom-Interpolation wird die Programmierung eines Nenner-Polynoms mit Nullstellen innerhalb des Intervalls `[0,PL]` mit einem Alarm abgelehnt. Auf die Bewegung von Zusatzachsen hat das Nenner-Polynom keinen Einfluss.

**Hinweis**

Eine Werkzeugradiuskorrektur ist bei der Polynom-Interpolation mit `G41`, `G42` einschaltbar und wie für Geraden- oder Kreisinterpolation verwendbar.

## 3.7.6 Einstellbarer Bahnbezug (SPATH, UPATH)

Bei Polynominterpolation (`POLY`, `ASPLINE`, `BSPLINE`, `CSPLINE`, `COMP...`) werden die Positionen der Bahnachsen i durch Polynome $p_i(U)$ vorgegeben. Der Kurvenparameter U bewegt sich dabei innerhalb eines NC-Satzes von 0 bis 1.

Durch `FGROUP` werden die Achsen (FGROUP-Achsen) ausgewählt, auf die sich der Bahnvorschub `F` bezieht. Eine Interpolation mit konstanter Geschwindigkeit auf dem Bahnweg S der FGROUP-Achsen bedeutet während der Polynominterpolation jedoch in der Regel eine nicht konstante Änderung des Kurvenparameters U. Für die nicht in `FGROUP` enthaltenen Achsen kann daher zwischen zwei Möglichkeiten gewählt werden, wie diese den FGROUP-Achsen folgen sollen:

- Synchron zum Bahnweg S (`SPATH`)
- Synchron zum Kurvenparameter U (`UPATH`)

### Syntax

```
SPATH
UPATH
```

## Bedeutung

| | |
|---|---|
| SPATH: | Die nicht in FGROUP enthaltenen Achsen werden bezogen auf den Bahnweg S verfahren |
| UPATH: | Die nicht in FGROUP enthaltenen Achsen werden bezogen auf den Kurvenparameter U verfahren |

**Hinweis**

UPATH und SPATH bestimmen auch den Zusammenhang des F-Wort-Polynoms (FPOLY, FCUB, FLIN) mit der Bahnbewegung.

## Randbedingungen

SPATH bzw. UPATH haben keine Bedeutung bei:

- Linearinterpolation (G1)
- Kreisinterpolation (G2, G3)
- Gewindesätzen (G33, G34, G35, G33x, G63)
- Alle Bahnachsen sind in FGROUP enthalten

## Beispiel

Das folgende Beispiel zeigt den Unterschied zwischen den beiden Arten der Bewegungsführung.

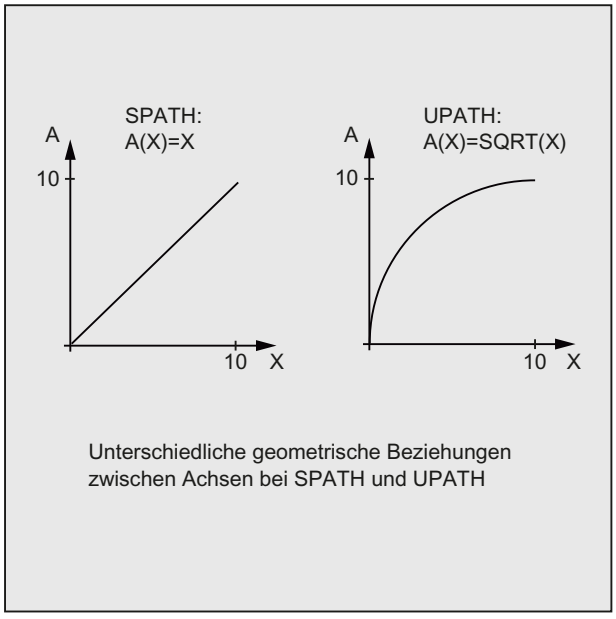
**Programmcode**

```
N10 FGROUP(X,Y,Z)
N15 G1 X0 A0 F1000 SPATH                ; SPATH
N20 POLY PO[X]=(10,10) A10
```

**Programmcode**

```
N10 FGROUP(X,Y,Z)
N15 G1 X0 A0 F1000 UPATH                ; UPATH
N20 POLY PO[X]=(10,10) A10
```

In beiden Programmausabschnitten hängt in N20 der Weg S der FGROUP-Achsen vom Quadrat des Kurvenparameters U ab. Daher ergeben sich entlang des Wegs von X unterschiedliche Positionen der Synchronachse A, je nachdem, ob SPATH oder UPATH aktiv ist.




Unterschiedliche geometrische Beziehungen zwischen Achsen bei SPATH und UPATH

## Weitere Informationen

**Steuerungsverhalten bei Reset und Maschinen-/Optionsdaten**

Nach Reset ist der durch MD20150 $MC_GCODE_RESET_VALUES[ 44 ] bestimmte G-Befehl wirksam (45. G-Gruppe).

Der Grundstellungswert für die Art des Überschleifens wird mit MD20150 $MC_GCODE_RESET_VALUES[9] festgelegt (10. G-Gruppe).

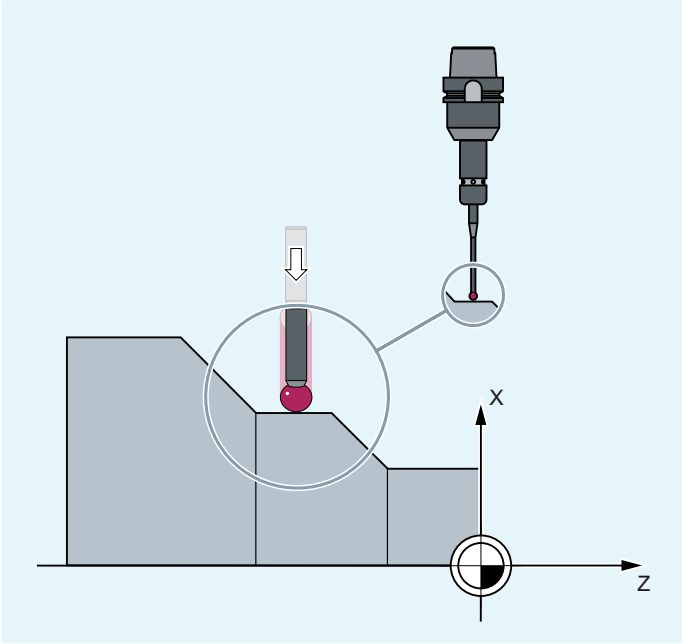
Das achsspezifische Maschinendatum MD33100 $MA_COMPRESS_POS_TOL[<n>] hat eine erweiterte Bedeutung: es enthält die Toleranzen für die Kompressorfunktion und für das Überschleifen mit G642.

## 3.7.7 Kanalspezifisches Messen (MEAS, MEAW)

Beim kanalspezifischen Messen erfolgt die Aktivierung des Messvorgangs für einen NC-Kanal stets aus dem Teileprogramm, das im betreffenden Kanal läuft. In einem Messsatz wird jeweils **ein** Trigger-Ereignis (steigende oder fallende Flanke des Messtasters) und **ein** Messmodus mit Restweglöschen (MEAS) oder ohne Restweglöschen (MEAW) programmiert. Am Messvorgang sind dann **alle** im Messsatz programmierten Achsen beteiligt.

Sobald ein Messsatz aktiv ist, wird der Messtaster an das Werkstück herangefahren. Bei der Schaltflanke des Messtasters werden für alle im Messsatz programmierten Achsen die Positionen gemessen und für jede Achse in die entsprechende Speicherzelle geschrieben.

Die Messergebnisse lassen sich im Teileprogramm oder mit Synchronaktionen sowohl im Maschinen- als auch im Werkstückkoordinatensystem lesen.




## Syntax

```
MEAS=<TE> G... X... Y... Z...
MEAW=<TE> G... X... Y... Z...
```

## Bedeutung

| | | | |
|---|---|---|---|
| MEAS: | Messen **mit** Restweglöschen | | |
| | Wirksamkeit: | | satzweise |
| MEAW: | Messen **ohne** Restweglöschen | | |
| | Wirksamkeit: | | satzweise |
| | Anwendung: | | Für Messaufgaben, bei denen in jedem Fall die programmierte Position angefahren werden soll. |
| <TE>: | Trigger-Ereignis zur Auslösung der Messung | | |
| | Typ: | INT | |
| | Wertebereich: | | -2, -1, 1, 2 |
| | Wert: | (+)1 | steigende Flanke von Messtaster 1 (auf Messeingang 1) |
| | | -1 | fallende Flanke von Messtaster 1 (auf Messeingang 1) |
| | | (+)2 | steigende Flanke von Messtaster 2 (auf Messeingang 2) |
| | | -2 | fallende Flanke von Messtaster 2 (auf Messeingang 2) |
| | **Hinweis:**<br>Es existieren maximal 2 Messtaster (je nach Ausbaustufe). | | |
| G...: | Interpolationsart, z. B. G0, G1, G2 oder G3 | | |
| X... Y... Z...: | Endpunkte in kartesischen Koordinaten | | |

**Hinweis**

MEAS und MEAW sind satzweise wirksam und werden zusammen mit Bewegungsanweisungen programmiert. Vorschub und Interpolationsart (G0, G1, …), ebenso wie die Anzahl der Achsen, müssen dabei dem jeweiligen Messproblem angepasst sein.

## Beispiel

```
Programmcode                                  Kommentar
...
N10 MEAS=1 G1 F1000 X100 Y730 Z40             ; Messsatz mit Messtaster des ersten Mes-
                                              seingangs und Geradeninterpolation. Vor-
                                              laufstopp wird automatisch erzeugt.
...
```

## Weitere Informationen

### Status abfragen

Ist im Programm eine Auswertung erforderlich, ob ein Messtaster ausgelenkt wurde bzw. geschaltet hat, kann der Status über folgende Systemvariablen abgefragt werden:

| Systemvariable | Bedeutung | Datentyp | Wert | |
|---|---|---|---|---|
| $A_PROBE[<n>] | Messtasterzustand | INT | 0 | Messtaster nicht ausgelenkt. |
| | | | 1 | Messtaster ausgelenkt. |
| $AC_MEA[<n>] | Schaltzustand des Messtasters<br>$AC_MEA[<n>] wird zu Beginn einer Messung automatisch zurückgesetzt. | INT | 0 | Messtaster hat nicht geschaltet. |
| | | | 1 | Messtaster hat geschaltet. |

<n> = Nummer des Messtasters

### Messwerteaufnahme

Bei der kanalspezifischen Messung werden die Positionen aller verfahrenen Bahn- und Positionierachsen des Satzes (maximale Anzahl an Achsen je nach Steuerungskonfiguration) erfasst. Bei MEAS wird die Bewegung nach dem Schalten des Messtasters definiert abgebremst.

**Hinweis**

Ist in einem Messsatz eine Geometrieachse programmiert, werden die Messwerte für alle aktuellen Geometrieachsen abgelegt.

Ist in einem Messsatz eine an einer Transformation beteiligte Achse programmiert, werden die Messwerte aller an dieser Transformation beteiligten Achsen abgelegt.

### Messergebnisse lesen

Die Messtaster-Messwerte der erfassten Achsen können über die folgenden Systemvariablen im Teileprogramm und in Synchronaktionen gelesen werden:

| Systemvariable | Bedeutung |
|---|---|
| $AA_MM[<Axis>] | Messtaster-Messwert im Maschinenkoordinatensystem. |
| $AA_MW[<Axis>] | Messtaster-Messwert im Werkstückkoordinatensystem. |

## 3.7.8 Achsspezifisches Messen (MEASA, MEAWA, MEAC) (Option)

Beim achsspezifischen Messen kann die Aktivierung im Teileprogramm **oder** in Synchronaktionen erfolgen. Stehen für die Achse zwei Messsysteme zur Verfügung, können beide für die Messung verwendet werden.

Folgende Messmethoden stehen zur Verfügung:

- Messen mit Restweglöschen (MEASA)
- Messen ohne Restweglöschen (MEAWA)
- Kontinuierliches Messen ohne Restweglöschen (MEAC)

Mit MEASA bzw. MEAWA werden für die jeweils programmierte Achse bis zu vier Messwerte pro Messung erfasst und passend zum Trigger-Ereignis in Systemvariablen abgelegt.

Beim kontinuierlichen Messen mit MEAC werden die Messergebnisse in FIFO-Variablen abgelegt.

### Syntax

```
MEASA[<Axis>]=(<Mode>,<TE1>,...,<TE4>)
MEAWA[<Axis>]=(<Mode>,<TE1>,...,<TE4>)
MEAC[<Axis>]=(<Mode>,<MeasMem>,<TE1>,...,<TE4>)
```

**Hinweis**

MEASA und MEAWA sind satzweise wirksam und können zusammen in einem Satz programmiert werden. Wird dagegen MEASA/MEAWA zusammen mit MEAS/MEAW in einem Satz programmiert, kommt es zu einer Fehlermeldung.

### Bedeutung

| | | |
|---|---|---|
| MEASA: | Achsspezifisches Messen mit Restweglöschen | |
| | Wirksamkeit: | satzweise |
| MEAWA: | Achsspezifisches Messen ohne Restweglöschen | |
| | Wirksamkeit: | satzweise |
| MEAC: | Achsspezifisches kontinuierliches Messen ohne Restweglöschen | |
| | Wirksamkeit: | satzweise |
| <Axis>: | Name der zu Messung verwendeten Kanalachse | |

| | | | |
|---|---|---|---|
| <Mode>: | Zweistellige (xx) Ziffer zur Angabe des Betriebsmodus (Messmodus und Messsystem) | | |
| | Einerdekade: Messmodus<br>Legt fest, ob die Trigger-Ereignisse in zeitlicher oder programmierter Reihenfolge aktiviert werden sollen. | | |
| | x**0** | Messauftrag abbrechen. | |
| | x**1** | Bis zu 4 **verschiedene** gleichzeitig aktivierbare Trigger-Ereignisse. | |
| | x**2** | Bis zu 4 nacheinander aktivierbare Trigger-Ereignisse. | |
| | x**3** | Bis zu 4 nacheinander aktivierbare Trigger-Ereignisse, jedoch keine Überwachung von Trigger-Ereignis 1 beim Start (Alarme 21700/21703 werden unterdrückt).<br>**Hinweis:**<br>Dieser Modus ist bei MEAC nicht möglich. | |
| | Zehnerdekade: Messsystem<br>Legt fest, mit welchem Messsystem die Messung durchgeführt werden soll. | | |
| | **0**x (oder keine Angabe) | aktives Messsystem | |
| | **1**x | Messsystem 1 | |
| | **2**x | Messsystem 2 | |
| | **3**x | beide Messsysteme | |
| <TE>: | Trigger-Ereignis zur Auslösung der Messung | | |
| | Typ: | INT | |
| | Wertebereich: | -2, -1, 1, 2 | |
| | (+)1 | steigende Flanke von Messtaster 1 | |
| | -1 | fallende Flanke von Messtaster 1 | |
| | (+)2 | steigende Flanke von Messtaster 2 | |
| | -2 | fallende Flanke von Messtaster 2 | |
| | **Hinweis:**<br>Wenn der Messvorgang mit zwei Messsystemen durchgeführt wird, sind maximal zwei Trigger- Ereignisse programmierbar (steigende oder fallende Flanke). Bei jedem der beiden Trigger- Ereignisse werden die Messwerte beider Messsysteme erfasst. | | |
| <MeasMem>: | Nummer des FIFO (Umlaufspeichers) | | |

## Beispiele

### Beispiel 1: Achsspezifisches Messen mit Restweglöschen im Modus 1 (Auswertung in zeitlicher Reihenfolge)

a) Messen mit einem Messsystem

| **Programmcode** | **Kommentar** |
|---|---|
| ... | |
| N100 **MEASA[X]=(1,1,-1)** G01 X100 F100 | ; Messen im Modus 1 mit aktivem Messsystem. Warten auf Messsignal mit steigender/ fallender Flanke von Messtaster 1 auf dem Verfahrweg nach X=100. |
| N110 IF $AC_MEA[1]==FALSE GOTOF ENDE | ; Erfolg der Messung kontrollieren. |

| Programmcode | Kommentar |
|---|---|
| N120 R10=$AA_MM1[X] | ; Zum ersten programmierten Trigger-Ereignis (steigende Flanke) gehörigen Messwert speichern. |
| N130 R11=$AA_MM2[X] | ; Zum zweiten programmierten Trigger-Ereignis (fallende Flanke) gehörigen Messwert speichern. |
| N140 ENDE: | |

b) Messen mit zwei Messsystemen

| Programmcode | Kommentar |
|---|---|
| ... | |
| N200 **MEASA[X]=(31,1,-1)** G01 X100 F100 | ; Messen im Modus 1 mit beiden Messsystemen. Warten auf Messsignal mit steigender/fallender Flanke von Messtaster 1 auf dem Verfahrweg nach X=100. |
| N210 IF $AC_MEA[1]==FALSE GOTOF ENDE | ; Erfolg der Messung kontrollieren. |
| N220 R10=$AA_MM1[X] | ; Messwert des Messsystems 1 bei steigender Flanke speichern. |
| N230 R11=$AA_MM2[X] | ; Messwert des Messsystems 2 bei steigender Flanke speichern. |
| N240 R12=$AA_MM3[X] | ; Messwert des Messsystems 1 bei fallender Flanke speichern. |
| N250 R13=$AA_MM4[X] | ; Messwert des Messsystems 2 bei fallender Flanke speichern. |
| N260 ENDE: | |

### Beispiel 2: Achsspezifisches Messen mit Restweglöschen im Modus 2 (Auswertung in programmierter Reihenfolge)

| Programmcode | Kommentar |
|---|---|
| ... | |
| N100 **MEASA[X]=(2,1,-1,2,-2)** G01 X100 F100 | ; Messen im Modus 2 mit aktivem Messsystem. Warten auf Messsignal in der Reihenfolge steigende Flanke von Messtaster 1, fallende Flanke Messtaster 1, steigende Flanke von Messtaster 2, fallende Flanke Messtaster 2 auf dem Verfahrweg nach X=100. |
| N110 IF $AC_MEA[1]==FALSE GOTOF MESSTASTER2 | ; Erfolg der Messung mit Messtaster 1 kontrollieren. |
| N120 R10=$AA_MM1[X] | ; Zum ersten programmierten Trigger-Ereignis (steigende Flanke Messtaster 1) gehörigen Messwert speichern. |
| N130 R11=$AA_MM2[X] | ; Zum zweiten programmierten Trigger-Ereignis (steigende Flanke Messtaster 1) gehörigen Messwert speichern. |
| N140 MESSTASTER2: | |

| Programmcode | Kommentar |
|---|---|
| N150 IF $AC_MEA[2]==FALSE GOTOF ENDE | ; Erfolg der Messung mit Messtaster 2 kontrollieren. |
| N160 R12=$AA_MM3[X] | ; Zum dritten programmierten Trigger-Ereignis (steigende Flanke Messtaster 2) gehörigen Messwert speichern. |
| N170 R13=$AA_MM4[X] | ; Zum vierten programmierten Trigger-Ereignis (steigende Flanke Messtaster 2) gehörigen Messwert speichern. |
| N180 ENDE: | |

### Beispiel 3: Achsspezifisches kontinuierliches Messen im Modus 1 (Auswertung in zeitlicher Reihenfolge)

a) Messen von bis zu 100 Messwerten

| Programmcode | Kommentar |
|---|---|
| ... | |
| N110 DEF REAL MESSWERT[100] | |
| N120 DEF INT Schleife=0 | |
| N130 **MEAC[X]=(1,1,-1)** G01 X1000 F100 | ; Messen im Modus 1 mit aktivem Messsystem, Speichern der Messwerte unter $AC_FIFO1, Warten auf Messsignal mit fallender Flanke von Messtaster 1 auf dem Verfahrweg nach X=1000. |
| N135 STOPRE | |
| N140 MEAC[X]=(0) | ; Messung nach Erreichen der Achsposition abbrechen. |
| N150 R1=$AC_FIFO1[4] | ; Anzahl aufgelaufener Messwerte in Parameter R1 speichern. |
| N160 FOR Schleife=0 TO R1-1 | |
| N170 MESSWERT[Schleife]=$AC_FIFO1[0] | ; Messwerte aus dem $AC_FIFO1 auslesen und abspeichern. |
| N180 ENDFOR | |

b) Messen mit Restweglöschen nach 10 Messwerten

| Programmcode | Kommentar |
|---|---|
| ... | |
| N10 WHEN $AC_FIFO1[4]>=10 DO MEAC[x]=(0) DELDTG(x) | ; Restweg löschen. |
| N20 **MEAC[x]=(1,1,1,-1)** G01 X100 F500 | |
| N30 MEAC[X]=(0) | |
| N40 R1=$AC_FIFO1[4] | ; Anzahl Messwerte. |
| ... | |

c) Messen einer fallenden/steigenden Zahnflanke mit 2 Messtastern

```
Programmcode                                    Kommentar
...
N110 DEF REAL MESSWERT[16]
N120 DEF INT Schleife=0
N130 MEAC[X]=(1,1,-1,2) G01 X100 F100           ; Messen im Modus 1 mit aktivem Messsys-
                                                tem, Speichern der Messwerte unter $AC_FI-
                                                FO1, Warten auf Messsignal in der Reihen-
                                                folge fallende Flanke von Messtaster 1,
                                                steigende Flanke Messtaster 2, auf dem
                                                Verfahrweg nach X=100.
N140 STOPRE                                     ; Vorlaufstopp.
N150 MEAC[X]=(0)                                ; Messung nach Erreichen der Achsposition
                                                abbrechen.
N160 R1=$AC_FIFO1[4]                            ; Anzahl aufgelaufener Messwerte in Para-
                                                meter R1 speichern.
N170 FOR Schleife=0 TO R1-1
N180 MESSWERT[Schleife]=$AC_FIFO1[0]            ; Messwerte aus dem $AC_FIFO1 auslesen
                                                und abspeichern.
N190 ENDFOR
```

## Weitere Informationen

### Messauftrag

Die Programmierung eines Messauftrags kann im Teileprogramm oder aus einer Synchronaktion heraus erfolgen. Pro Achse kann dabei zu ein- und demselben Zeitpunkt nur ein Messauftrag aktiv sein.

### Hinweis

Der Vorschub ist dem jeweiligen Messproblem anzupassen.

Bei MEASA und MEAWA können korrekte Ergebnisse nur bei Vorschüben gewährleistet werden, bei denen nicht mehr als ein gleiches und nicht mehr als 4 verschiedene Trigger-Ereignisse pro Lagereglertakt eintreffen.

Beim kontinuierlichen Messen mit MEAC darf das Verhältnis zwischen Interpolatortakt und Lagereglertakt nicht größer als 1:8 werden.

### Trigger-Ereignis

Ein Trigger-Ereignis setzt sich zusammen aus der Nummer des Messtasters und dem Auslösekriterium (steigende oder fallende Flanke) des Messsignals.

Für jede Messung können jeweils bis zu 4 Trigger-Ereignisse der angesprochenen Messtaster verarbeitet werden, also bis zu zwei Messtaster mit je zwei Messflanken. Die Reihenfolge der Verarbeitung sowie die maximale Anzahl der Trigger-Ereignisse sind dabei abhängig vom gewählten Modus.

Wenn der Messvorgang mit zwei Messsystemen durchgeführt wird, sind maximal zwei Trigger-Ereignisse programmierbar (steigende oder fallende Flanke). Bei jedem der beiden Trigger-Ereignisse werden die Messwerte beider Messtaster erfasst.

Bei der Verwendung von PROFIBUS-Telegramm 391 (Standardeinstellung für die PROFIBUS-Kommunikation) ist nur ein Messwert pro Trigger-Ereignis und Lageregler-Takt möglich.

Bei der Verwendung von PROFIBUS-Telegramm 395 kann bei MEAC im Betriebsmodus 1 die Anzahl der Messwerte pro Trigger-Ereignis auf insgesamt 8 Messwerte bei steigender und 8 bei fallender Flanke pro Trigger-Ereignis und Lageregler-Takt erhöht werden:

- Ein Messtaster: 8 Messwerte bei steigender und 8 bei fallender Flanke
- Zwei Messtaster: 4 Messwerte bei steigender und 4 bei fallender Flanke je Messtaster

Dadurch lassen sich bei Verwendung von PROFIBUS-Telegramm 395 höhere Vorschübe oder Drehzahlen realisieren.

#### Betriebsmodus

Mit der ersten Ziffer (Zehnerdekade) des Betriebsmodus wird das gewünschte Messsystem angewählt. Ist nur ein Messsystem vorhanden, jedoch das zweite programmiert, wird automatisch das vorhandene eingesetzt.

Mit der zweiten Ziffer (Einerdekade) wird der gewünschte Messmodus angewählt. Damit wird der Messvorgang an die Möglichkeiten der jeweiligen Steuerung angepasst:

- Modus 1
  Die Auswertung der Trigger-Ereignisse erfolgt in der zeitlichen Reihenfolge ihres Auftretens.
- Modus 2
  Die Auswertung der Trigger-Ereignisse erfolgt in der programmierten Reihenfolge.
- Modus 3
  Die Auswertung der Trigger-Ereignisse erfolgt in der programmierten Reihenfolge, jedoch keine Überwachung von Trigger-Ereignis 1 beim Start.

#### Messen mit und ohne Restweglöschen

Bei der Programmierung von MEASA wird Restweglöschen erst nach der Erfassung aller geforderten Messwerte durchgeführt.

Für spezielle Messaufgaben, bei denen in jedem Fall die programmierte Position angefahren werden soll, wird MEAWA eingesetzt.

#### Hinweis

MEASA ist nicht in Synchronaktionen programmierbar. Ersatzweise kann MEAWA plus Restweglöschen als Synchronaktion programmiert werden.

Wird der Messauftrag mit MEAWA aus den Synchronaktionen gestartet, sind die Messwerte nur im Maschinen-Koordinatensystem verfügbar.

#### Geometrieachsen / Transformationen

Soll das achsspezifische Messen für eine Geometrieachse gestartet werden, muss der gleiche Messauftrag explizit für alle restlichen Geometrieachsen programmiert werden. Das Gleiche gilt für Achsen, die an einer Transformation beteiligt sind.

Beispiele:
```
N10 MEASA[Z]=(1,1) MEASA[Y]=(1,1) MEASA[X]=(1,1) G0 Z100
```

oder
```
N10 MEASA[Z]=(1,1) POS[Z]=100
```

### Status abfragen

Ist im Programm eine Auswertung erforderlich, ob ein Messtaster ausgelenkt wurde bzw. geschaltet hat, kann der Status über folgende Systemvariablen abgefragt werden:

| Systemvariable | Bedeutung | Datentyp | Wert | |
|---|---|---|---|---|
| $A_PROBE[<n>] | Messtasterzustand | INT | 0 | Messtaster nicht ausgelenkt. |
| | | | 1 | Messtaster ausgelenkt. |
| $AC_MEA[<n>] | Schaltzustand des Messtasters<br>$AC_MEA[<n>] wird zu Beginn einer Messung automatisch zurückgesetzt. | INT | 0 | Messtaster hat nicht geschaltet. |
| | | | 1 | Messtaster hat geschaltet (alle im Messsatz programmierten Trigger-Ereignisse sind erfolgt). |

<n> = Nummer des Messtasters

### Hinweis

Wird Messen aus Synchronaktionen gestartet, wird $AC_MEA nicht mehr aktualisiert. In diesem Fall ist das NC/PLC-Nahtstellensignal DB31, … DBX62.3 bzw. die gleichwertige Variable $AA_MEAACT[<Axis>] abzufragen:

$AA_MEAACT==1: Messen aktiv

$AA_MEAACT==0: Messen nicht aktiv

### Messtasterbegrenzung

Mit der Systemvariablen $A_PROBE_LIMITED kann im NC-Programm oder Synchronaktion der Status der Messtasterbegrenzung bei Verwendung des PROFIBUS-Telegramms 395 gelesen werden:

$A_PROBE_LIMITED[<n>] == 0: Messtasterbegrenzung inaktiv/zurückgesetzt

$A_PROBE_LIMITED[<n>] == 1: Messtasterbegrenzung aktiv

<n> = Messtasternummer

### Messergebnisse für MEASA / MEAWA

Die Messtaster-Messwerte können über die folgenden Systemvariablen im Teileprogramm und in Synchronaktionen gelesen werden:

| Systemvariable | Bedeutung |
|---|---|
| $AA_MM1[<Axis>]<br>...<br>$AA_MM4[<Axis>] | Messtaster-Messwert bei Trigger-Ereignis 1 im MKS<br>...<br>Messtaster-Messwert bei Trigger-Ereignis 4 im MKS |
| $AA_MW1[<Axis>]<br>...<br>$AA_MW4[<Axis>] | Messtaster-Messwert bei Trigger-Ereignis 1 im WKS<br>...<br>Messtaster-Messwert bei Trigger-Ereignis 4 im WKS |

<Axis> = Messachse

Wird der Messauftrag mit zwei Messsystemen durchgeführt, wird jedes der beiden möglichen Trigger-Ereignisse von beiden Messsystemen erfasst. Die Belegung der Systemvariablen ist dann wie folgt:

| | | | |
|---|---|---|---|
| $AA_MM1[<Axis>] | bzw. | $AA_MW1[<Axis>] | Messwert von Messsystem 1 bei Trigger-Ereignis 1 |
| $AA_MM2[<Axis>] | bzw. | $AA_MW2[<Axis>] | Messwert von Messsystem 2 bei Trigger-Ereignis 1 |
| $AA_MM3[<Axis>] | bzw. | $AA_MW3[<Axis>] | Messwert von Messsystem 1 bei Trigger-Ereignis 2 |
| $AA_MM4[<Axis>] | bzw. | $AA_MW4[<Axis>] | Messwert von Messsystem 2 bei Trigger-Ereignis 2 |

### Kontinuierliches Messen (MEAC)

Die Messwerte liegen bei MEAC im Maschinenkoordinatensystem vor und werden im angegebenen FIFO[<n>]-Speicher (Umlaufspeicher) abgelegt. Sind für die Messung zwei Messtaster projektiert, werden die Messwerte des zweiten Messtasters getrennt im zusätzlich dafür projektierten (über MD einstellbar) FIFO[<n>+1]-Speicher abgelegt.

Der FIFO-Speicher ist ein Umlaufspeicher, in den Messwerte im Umlaufprinzip in $AC_FIFO-Variablen eingetragen werden.

---

**Hinweis**

Der FIFO-Inhalt kann nur einmal aus dem Umlaufspeicher ausgelesen werden. Zur Mehrfachverwendung der Messdaten müssen diese in den Anwenderdaten zwischengespeichert werden.

Überschreitet die Anzahl der Messwerte für den FIFO-Speicher die im Maschinendatum festgelegte Höchstzahl, so wird die Messung automatisch beendet.

Endloses Messen lässt sich durch zyklisches Auslesen von Messwerten realisieren. Das Auslesen muss dabei mindestens in der gleichen Häufigkeit wie der Eingang von neuen Messwerten erfolgen.

---

**Weitere Informationen:** Funktionshandbuch Synchronaktionen

**Schutz vor Fehlerprogrammierungen**

Folgende Fehlprogrammierungen werden erkannt und mit einem Fehler angezeigt:

- MEASA/MEAWA zusammen mit MEAS/MEAW in einem Satz programmiert
  Beispiel:
  ```
  N01 MEAS=1 MEASA[X]=(1,1) G01 F100 POS[X]=100
  ```
- MEASA/MEAWA mit Parameteranzahl <2 oder >5
  Beispiel:
  ```
  N01 MEAWA[X]=(1) G01 F100 POS[X]=100
  ```
- MEASA/MEAWA mit Trigger-Ereignis ungleich 1/ -1/ 2/ -2
  Beispiel:
  ```
  N01 MEASA[B]=(1,1,3) B100
  ```
- MEASA/MEAWA mit falschem Modus
  Beispiel:
  ```
  N01 MEAWA[B]=(4,1) B100
  ```
- MEASA/MEAWA mit doppelt programmiertem Trigger-Ereignis
  Beispiel:
  ```
  N01 MEASA[B]=(1,1,-1,2,-1) B100
  ```
- MEASA/MEAWA und fehlende Geometrieachse
  Beispiel:
  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) G01 X50 Y50 Z50 F100 ;GEO-Achse X/Y/Z
  ```
- Uneinheitlicher Messauftrag bei Geometrieachsen
  Beispiel:
  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) MEASA[Z]=(1,1,2) G01 X50 Y50 Z50 F100
  ```

### Siehe auch

Synchronaktionen (Seite 967)

## 3.7.9 Spezielle Funktionen für den OEM-Anwender (OMA1 ... OMA5, OEMIPO1, OEMIPO2, G810 ... G829)

### OEM-Adressen

Die Bedeutung der OEM-Adressen bestimmt der OEM-Anwender. Die Funktionalität wird über Compile-Zyklen eingebracht. 5 OEM-Adressen sind reserviert (`OMA1` ... `OMA5`). Die Adressbezeichner sind einstellbar. OEM-Adressen sind in jedem Satz zulässig.

### Reservierte G-Befehlsaufrufe

Für den OEM-Anwender sind folgende G-Befehlsaufrufe reserviert:

- `OEMIPO1`, `OEMIPO2` (aus G-Gruppe 1)
- `G810` ... `G819` (G-Gruppe 31)
- `G820` ... `G829` (G-Gruppe 32)

Die Funktionalität wird über Compile-Zyklen eingebracht.

### Funktionen und Unterprogramme

Zusätzlich können OEM-Anwender auch vordefinierte Funktionen und Unterprogramme mit Parameterübergabe anlegen.

**Hinweis**

**Werkstücksimulation**

Bis SW 4.4 werden keine, ab SW 4.4 nur ausgesuchte Compile-Zyklen (CC) bei der Werkstücksimulation unterstützt.

Sprachbefehle im Teileprogramm von nicht unterstützten Compile-Zyklen (`OMA1` ... `OMA5`, `OEMIPO1/2`, `G810` ... `G829`, eigene Prozeduren und Funktionen) führen daher ohne individuelle Behandlung zur Alarmmeldung und zum Abbruch der Simulation.

**Lösung:** Die fehlenden CC-spezifischen Sprachelemente im Teileprogramm individuell behandeln ($P_SIM-Abfrage).
Beispiel:

```
N1 G01 X200 F500
IF (1==$P_SIM)
N5 X300 ;bei Simulation CC nicht aktiv
ELSE
N5 X300 OMA1=10
ENDIF
```

## 3.7.10 Vorschubreduzierung mit Eckenverzögerung (FENDNORM, G62, G621)

Bei der automatischen Eckenverzögerung wird der Vorschub glockenförmig kurz vor der betreffenden Ecke abgesenkt. Außerdem kann das Ausmaß des für die Bearbeitung relevanten Werkzeugverhaltens über Settingdaten parametriert werden:

- Beginn und Ende der Vorschubreduzierung
- Override, mit dem der Vorschub reduziert wird
- Erkennung der relevanten Ecke

Als relevante Ecken werden diejenigen Ecken berücksichtigt, deren Innenwinkel kleiner als die über Settingdatum parametrierte Ecke ist.

Mit dem Defaultwert FENDNORM wird die Funktion des automatischen Eckenoverride ausgeschaltet.

### Syntax

```
FENDNORM
G62 G41
G621
```

### Bedeutung

| | |
|---|---|
| FENDNORM: | Automatischer Eckenverzögerung aus |
| G62: | Eckenverzögerung **an Innenecken** bei aktiver Werkzeugradiuskorrektur |
| G621: | Eckenverzögerung **an allen Ecken** bei aktiver Werkzeugradiuskorrektur |

G62 wirkt nur an den Innenecken mit aktiver Werkzeugradiuskorrektur G41/G42 und aktivem Bahnsteuerbetrieb G64/G641.

Die entsprechende Ecke wird mit dem abgesenkten Vorschub angefahren, der sich wie folgt berechnet:

F * (Override zur Vorschubreduzierung) * Vorschuboverride

Die maximal mögliche Vorschubabsenkung wird genau dann erreicht, wenn das Werkzeug, bezogen auf die Mittelpunktsbahn, den Richtungswechsel an der betreffenden Ecke vornehmen soll.

G621 wirkt analog zu G62 an jeder Ecke der durch FGROUP festgelegten Achsen.

## 3.7.11 Programmierbares Bewegungsendekriterium (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)

Ähnlich dem Satzwechselkriterium bei Bahninterpolation (G601, G602 und G603) kann das Bewegungsendekriterium bei Einzelachsinterpolation in einem Teileprogramm bzw. in Synchronaktionen für Kommando-/PLC-Achsen programmiert werden.

Je nachdem, welches Bewegungsendekriterium eingestellt ist, werden Teileprogrammsätze bzw. Technologiezyklussätze mit Einzelachsbewegungen unterschiedlich schnell beendet. Gleiches gilt für PLC über FC15/16/18.

### Syntax

```
FINEA[<Achse>]
COARSEA[<Achse>]
IPOENDA[<Achse>]
IPOBRKA(<Achse>[,<Zeitpunkt>])
ADISPOSA(<Achse>[,<Modus>,<Fenstergröße>])
```

## Bedeutung

| | | | |
|---|---|---|---|
| FINEA: | Bewegungsendekriterium: "Genauhalt fein" | | |
| | Wirksamkeit: | modal | |
| COARSEA: | Bewegungsendekriterium: "Genauhalt grob" | | |
| | Wirksamkeit: | modal | |
| IPOENDA: | Bewegungsendekriterium:"Interpolator-Stopp" | | |
| | Wirksamkeit: | modal | |
| IPOBRKA: | Satzwechselkriterium: Bremsrampe | | |
| | Wirksamkeit: | modal | |
| ADISPOSA: | Toleranzfenster zum Bewegungsendekriterium | | |
| | Wirksamkeit: | modal | |
| <Achse>: | Kanalachsname (X, Y, ....) | | |
| <Zeitpunkt>: | Zeitpunkt des Satzwechsels, bezogen auf die Bremsrampe in %:<br>• 100% = Beginn der Bremsrampe<br>• 0% = Ende der Bremsrampe, gleichbedeutend mit IPOENDA | | |
| | Typ: | REAL | |
| <Modus>: | Bezug des Toleranzfensters | | |
| | Wertebereich: | 0 | Toleranzfenster nicht aktiv |
| | | 1 | Toleranzfenster bezüglich Sollposition |
| | | 2 | Toleranzfenster bezüglich Istposition |
| | Typ: | INT | |
| <Fenstergröße>: | Größe des Toleranzfensters | | |
| | Typ: | REAL | |

## Beispiele

### Beispiel 1: Bewegungsendekriterium: "Interpolator-Stopp"

**Programmcode**

```
; Fahre Positionierachse X auf 100, Geschwindigkeit 200
m/min, Beschleunigung 90%,
; Bewegungsendekriterium: Interpolator-Stopp
N110 G01 POS[X]=100 FA[X]=200 ACC[X]=90 IPOENDA[X]


; Synchronaktion:
; IMMER WENN: Eingang 1 gesetzt ist
; DANN Fahre Positionierachse X auf 50, Geschwindigkeit
200 m/min, Beschleunigung 140%,
;      Bewegungsendekriterium: Interpolator-Stopp
N120 EVERY $A_IN[1] DO POS[X]=50 FA[X]=200 ACC[X]=140
IPOENDA[X]
```

### Beispiel 2: Satzwechselkriterium: "Bremsrampe"

| Programmcode | Kommentar |
|---|---|
| | ; Defaulteinstellung wirksam |
| N40 POS[X]=100 | ; Positionierbewegung von X auf Position 100 |
| | Satzwechselkriterium: Genauhalt fein |
| N20 IPOBRKA(X,100) | ; Satzwechselkriterium: "Bremsrampe", |
| | 100% = Beginn der Bremsrampe |
| N30 POS[X]=200 | ; Satzwechsel erfolgt, sobald Achse X zu bremsen beginnt |
| N40 POS[X]=250 | ; Achse X bremst nicht weiter auf Position 200, sondern fährt weiter auf Position 250. |
| | Sobald die Achse zu bremsen beginnt erfolgt der Satzwechsel. |
| N50 POS[X]=0 | ; Achse X bremst und fährt auf Position 0 zurück. |
| | Satzwechsel erfolgt bei Position 0 und "Genauhalt fein" |
| N60 X10 F100 | ; Achse X fährt als Bahnachse auf Position 10 |

## Weitere Informationen

### Systemvariable für Bewegungsendekriterium

Das wirksame Bewegungsendekriterium kann über die Systemvariable $AA_MOTEND gelesen werden.

### Satzwechselkriterium: "Bremsrampe" (IPOBRKA)

Ist bei der Aktivierung des Satzwechselkriteriums "Bremsrampe" für den optionalen Satzwechselzeitpunkt ein Wert programmierte, wird dieser für die nächste Positionierbewegung wirksam und hauptlaufsynchron in das Settingdatum geschrieben. Ist für den Satzwechselzeitpunkt kein Wert angegeben, wird der aktuelle Wert des Settingdatums wirksam.

SD43600 $SA_IPOBRAKE_BLOCK_EXCHANGE

Mit der nächsten Programmierung eines axialen Bewegungsendekriteriums (FINEA, COARSEA , IPOENDA) wird IPOBRKA für die entsprechende Achse deaktiviert.

### Zusätzliches Satzwechselkriterium: "Toleranzfenster" (ADISPOSA)

Mit ADISPOSA kann als zusätzliches Satzwechselkriterium ein Toleranzfenster um den Satzendpunkt (wahlweise Ist- oder Sollposition) definiert werden. Für den Satzwechsel müssen dann beide Bedingungen erfüllt sein:

- Satzwechselkriterium: "Bremsrampe"
- Satzwechselkriterium: "Toleranzfenster"

# 3.8 Koordinatentransformationen (Frames)

## 3.8.1 Koordinatentransformation über Framevariable

Neben den Frame-Anweisungen (Seite 310) wie z. B. ROT, AROT, SCALE, etc. kann das Werkstück-Koordinatensystem (WKS) auch über die Framevariablen $P_...FR (Datenhaltungsframes) und $P_...FRAME (Aktive Frames) transformiert werden.

Das folgende Bild gibt einen Überblick über die Strukturierung der Framevariablen:

- Datenhaltungsframes
- Aktive Frames
- Aktiver Gesamtframe: Verkettung aller aktiven Frames
- NCU-globale Frames
- Kanalspezifische Frames

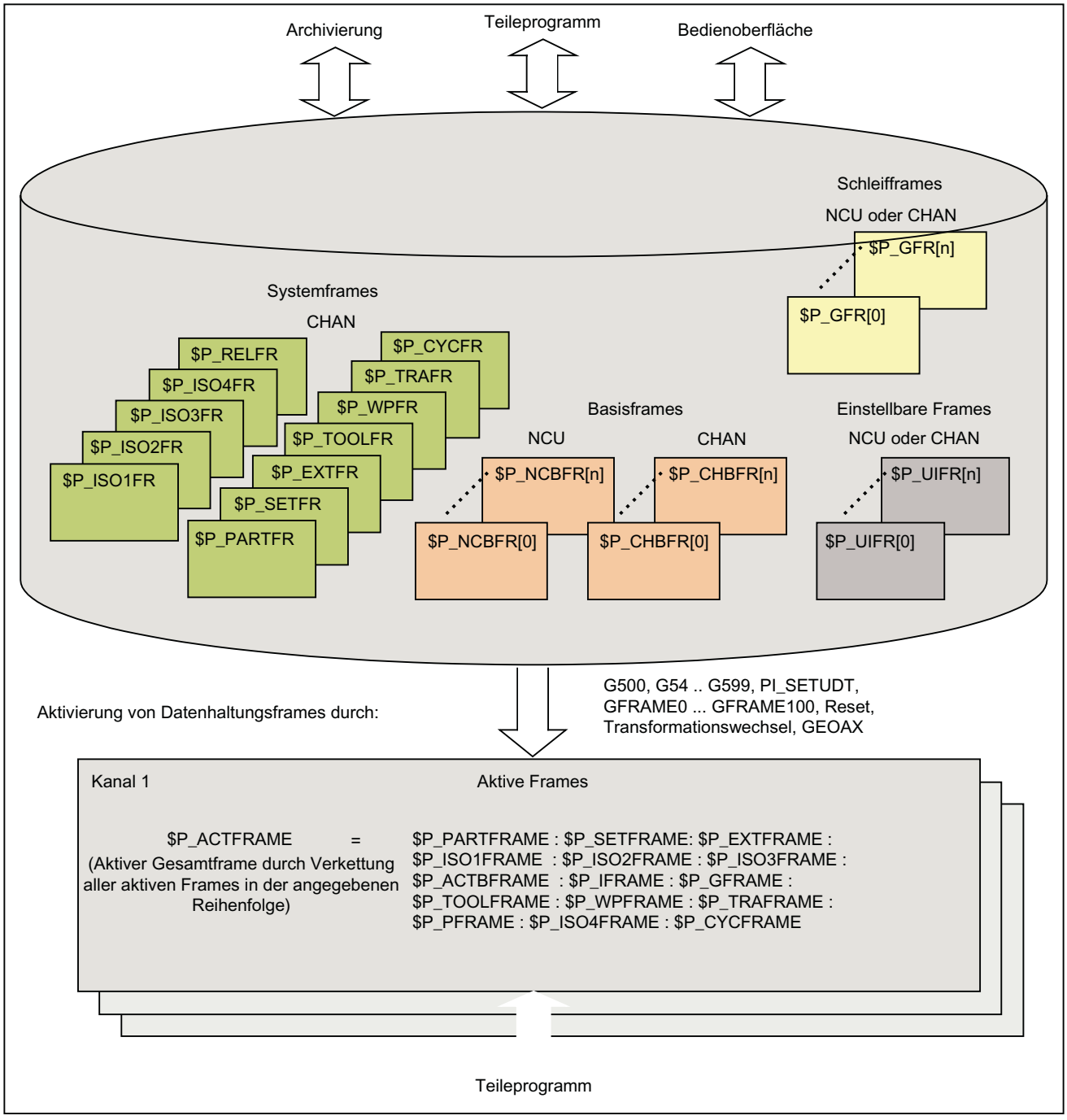



Bild 3-1 Übersicht der Framevariablen

### 3.8.1.1 Vordefinierte Framevariable ($P_CHBFRAME, $P_IFRAME, $P_PFRAME, $P_ACTFRAME)

#### Aktiv: kanalspezifische Basisframes $P_CHBFRAME[ <n> ] ($P_BFRAME)

**Hinweis**

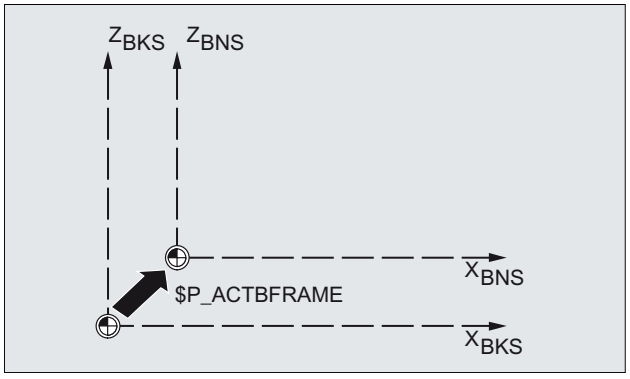
Die der aktuelle Basisframe `$P_BFRAME` und der Datenhaltungs-Basisframe `$P_UBFR` werden aus Kompatibliltätsgründen beibehalten.

- `$P_BFRAME` ≙ `$P_CHBFRAME[0]`
- `$P_UBFR` ≙ `$P_CHBFR[0]`.

Die Framevariablen $P_CHBFRAME[<n>] definieren den Bezug zwischen Basiskoordinatensystem (BKS) und Basis-Nullpunktsystem (BNS).

Soll der aktuelle kanalspezifische Basisframe $P_CHBFRAME[`<n>`] sofort im NC-Programm wirksam werden, stehen folgende Möglichkeiten zu Verfügung

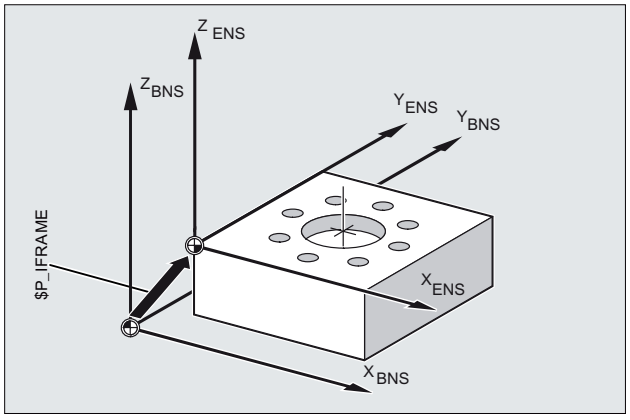
- Befehle:
  - `G500` (Ausschalten aller Einstellbaren Frames, die Basisframes bleiben aktiv)
  - `G54` ... `G599` (Einstellbare Nullpunktverschiebungen)
- Zuweisung eines kanalspezifischen Basisframes der Datenhaltung an einen aktuellen kanalspezifischen Basisiframe:
  `$P_CHBFRAME[<n>] = $P_CHBFR[<m>]`




#### Aktiv: kanalspezifischer Einstellbarer Frame $P_IFRAME

Die Framevariabln $P_IFRAME, definiert den Bezug zwischen Basis-Nullpunktsystem (BNS) und Einstellbarem Nullpunktsystem (ENS).

- `$P_IFRAME` entspricht `$P_UIFR[$P_IFRNUM]`
- `$P_IFRAME` enthält nach Programmierung von z. B. `G54` die durch `G54` definierte Translation, Rotation, Skalierung und Spiegelung.

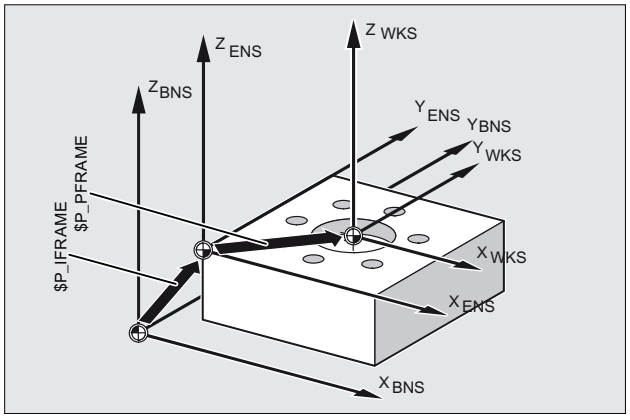



## Aktiv: kanalspezifischer Programmierbarer Frame $P_PFRAME

Die Framevariable $P_PFRAME, definiert den Bezug zwischen dem Einstellbaren Nullpunktsystem (ENS) und dem Werkstückkoordinatensystem (WKS).

`$P_PFRAME` enthält den resultierenden Frame, der sich

- **aus der Programmierung** von `TRANS/ATRANS`, `ROT/AROT`, `SCALE/ASCALE`, `MIRROR/AMIRROR` bzw.
- **aus der Zuweisung** von `CTRANS`, `CROT`, `CMIRROR`, `CSCALE` an den programmierbaren FRAME ergibt




## Aktiv: Gesamtframe $P_ACTFRAME

Der im Kanal wirksame Gesamtframe, ergibt sich aus der Verkettung aller im Kanal wirksamen Frames.

```
$P_ACTFRAME =    $P_PARTFRAME : $P_SETFRAME  : $P_EXTFRAME :
                 $P_ISO1FRAME : $P_ISO2FRAME : $P_ISO3FRAME :
                 $P_ACTBFRAME : $P_IFRAME    : $P_GFRAME :
                 $P_TOOLFRAME : $P_WPFRAME   : $P_TRAFRAME :
                 $P_PFRAME    : $P_ISO4FRAME : $P_CYCFRAME
```

`$P_ACTFRAME` beschreibt den Nujllpunkt des aktuellen Werkstückkoordinatensystems.

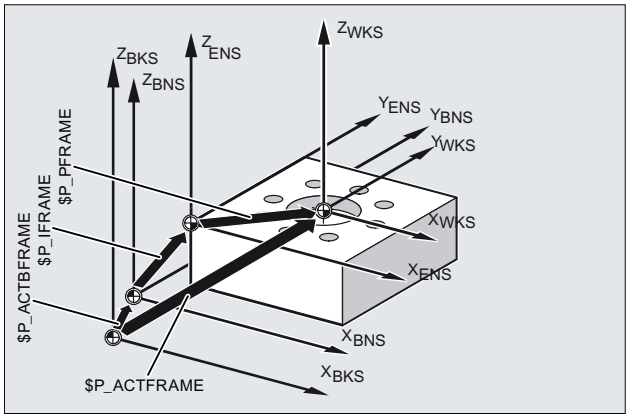
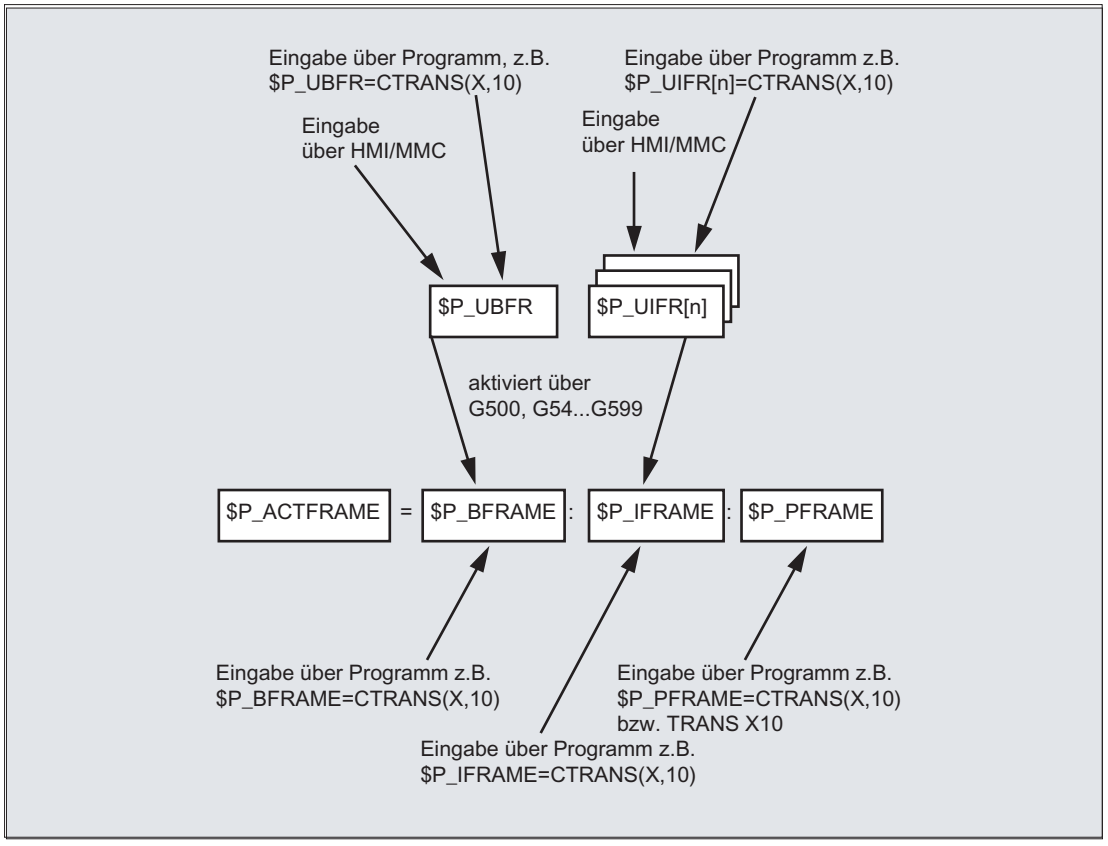



Bild 3-2 Framevariable $P_ACTFRAME

Wird einer der folgenden Frames `$P_BFRAME` / `$P_CHBFRAME[<n>]`, `$P_IFRAME` oder `$P_PFRAME` verändert, wird der aktuelle Gesamtframe `$P_ACTFRAME` neu berechnet.




Basisframe und Einstellbarer Frame wirken nach Reset, wenn das MD 20110 RESET_MODE_MASK folgendermaßen eingestellt ist:

Bit0=1, Bit14=1 --> `$P_UBFR` (Basisframe) wirkt

Bit0=1, Bit5=1 --> `$P_UIFR[$P_UIFRNUM]` (einst. Frame) wirkt

### Datenhaltung: Kanalspezifische Basisframes $P_CHBFR[<n>]

Über die Framevariablen $P_CHBFR[<n>] werden die Basisframes in der Datenhaltung gelesen / geschrieben. Der Datenhaltungsframe wird durch das Schreiben nicht sofort im Kanal aktiv. Das Aktivieren des geschriebenen Frames erfolgt bei:

- Kanal-Reset und MD20110 $MC_RESET_MODE_MASK,Bit0 == 1 und Bit14 == 1
- Befehl `G500`, `G54` ... `G57`, `G505` ... `G599` (Ein/Ausschalten von Basisframes mit anschließender Neuberechnung des Aktuellen Gesamtframes)

### Datenhaltung: Kanalspezifische Einstellbare Frames $P_UIFR[<n>]

Über die Framevariablen $P_UIFR[<n>] werden die Einstellbaren Frames in der Datenhaltung gelesen / geschrieben. Der Frame wird durch das Schreiben nicht sofort im Kanal aktiv. Das Einrechnen des geschriebenen Frames im Kanal erfolgt bei:

- Befehl `G500` (Ausschalten aller Einstellbaren Frames bzw. Nullpunktverschiebungen)
- Befehl `G54` ... `G57`, `G505` ... `G599` (Einschalten eines Einstellbaren Frames bzw. Nullpunktverschiebung)

| Aktiver Einstellbarer Frame | Datenhaltungsframe | (entspricht Befehl) |
|---|---|---|
| `$P_IFRAME` = | `$P_UIFR[0]` | `G500` |
| | `$P_UIFR[1]` | `G54` |
| | `$P_UIFR[2]` | `G55` |
| | `$P_UIFR[3]` | `G56` |
| | `$P_UIFR[4]` | `G57` |
| | `$P_UIFR[5]` | `G505` |
| | `$P_UIFR[6]` | `G506` |
| | ... | ... |
| | `$P_UIFR[99]` | `G599` |

## 3.8.2 Wertzuweisungen an Frames

### 3.8.2.1 Direkte Werte zuweisen (Achswert, Winkel, Maßstab)

Im NC-Programm können Sie direkt Frames oder Framevariablen mit Werten belegen.

### Syntax

#### Syntax

```
$P_PFRAME = CTRANS(X, <Verschiebungswert>, Y, <Verschiebungswert>,
Z, <Verschiebungswert>, ...)
$P_PFRAME = ROT(X, <Winkel>, Y, <Winkel>, Z, <Winkel>, ...)
$P_UIFR[..] = CROT(X, <Winkel>, Y, <Winkel>, Z, <Winkel>,  ...
```

```
$P_PFRAME = CSCALE(X, <Maßstab>, Y, <Maßstab>, Z, <Maßstab>, ...)
$P_PFRAME = CMIRROR(X, Y, Z)
```

Die Syntax für `$P_CHBFRAME[<n>]` ist identisch zu `$P_PFRAME`.

**Bedeutung**

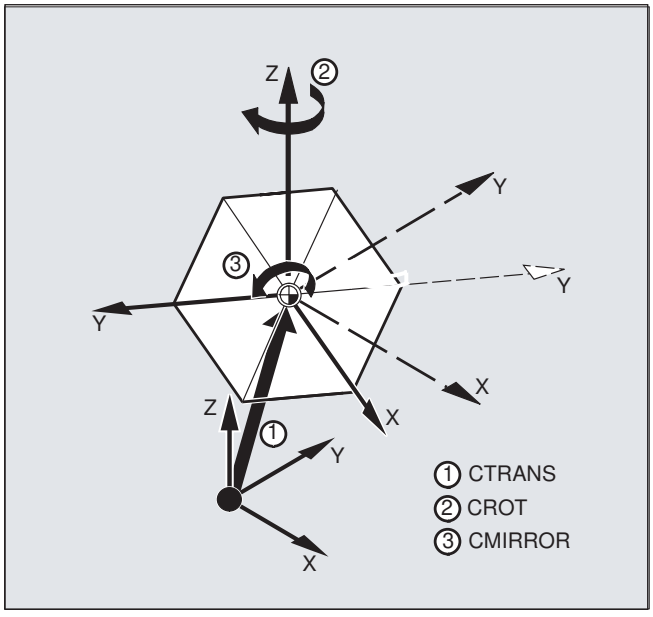
| | |
|---|---|
| `CTRANS`: | Verschiebung in den angegebenen Achsen |
| `CROT`: | Drehung um die angegebenen Achsen |
| `CSCALE`: | Maßstabsveränderung in den angegebenen Achsen |
| `CMIRROR`: | Richtungsumkehr der angegebenen Achse |
| `X, Y, Z`: | Verschiebewert in Richtung der angegebenen Geometrieachse |
| `<Verschiebungswert>`: | Verschiebungswert |
| `<Winkel>`: | Winkel um den gedreht wird |
| `<Maßstab>`: | Maßstabsangabe |

## Beispiele

### Wertzuweisungen an Framekomponenten des aktuellen Programmierbaren Frames

Wertzuweisung an die Framekomponenten Translation, Drehung und Spiegelung des aktuellen Programmierbaren Frames:

```
$P_PFRAME = CTRANS(X,10,Y,20,Z,5) : CROT(Z,45) : CMIRROR(Y)
```




### Rotationskmponenten eines Frames schreiben

Zuweisen von Werten an alle drei Achsen der Drehkomponente des Einstellbaren Datenhaltungsframes `$P_UIFR` mit `CROT` :

```
$P_UIFR[5] = CROT(X, 0, Y, 0, Z, 0)
```

Alternativ dazu die direkte Zuweisung der Einzelwerte direkt an die jeweilige Achse der Drehkomponente des Datenhaltungsframes:

```
$P_UIFR[5, Y, RT]=0
$P_UIFR[5, X, RT]=0
$P_UIFR[5, Z, RT]=0
```

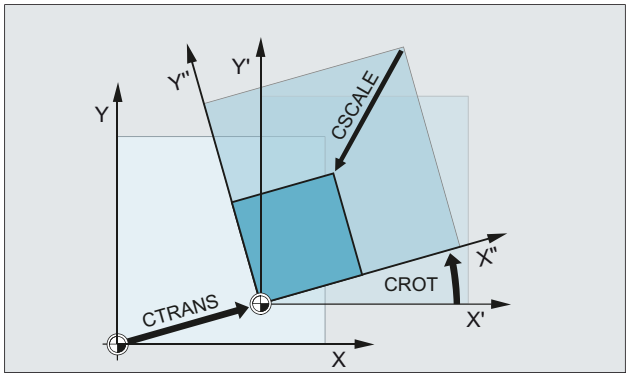
## Beschreibung

Mehrere Operationen auf einen Frame können durch den Kettungsoperator : miteinander verbunden werden. Die Operationen werden dann nacheinander von links nach rechts ausgeführt.

### Beispiel

Verkette Operationen auf $P_PFRAME mit Verschiebung, Rotation und Skalierung:

```
$P_PFRAME = CTRANS(...) : CROT(...) : CSCALE...
```




## 3.8.2.2 Framekomponenten lesen und verändern (TR, FI, RT, SC, MI)

Sie haben die Möglichkeit, auf **einzelne** Daten eines Frames, z. B. auf einen bestimmten Verschiebewert oder Drehwinkel zuzugreifen. Diese Werte können Sie verändern oder einer anderen Variablen zuweisen.

## Syntax

| | |
|---|---|
| `R10=$P_UIFR[$P_UIFNUM,X,RT]` | Der Drehwinkel RT um die X-Achse aus der aktuell gültigen einstellbaren Nullpunktverschiebung $P_UIFRNUM soll der Variablen R10 zugewiesen werden. |
| `R12=$P_UIFR[25,Z,TR]` | Der Verschiebewert TR in Z aus dem Datensatz des eingestellten Frames Nr. 25 soll der Variablen R12 zugewiesen werden. |

| | |
|---|---|
| `R15=$P_PFRAME[Y,TR]` | Der Verschiebewert TR in Y des aktuellen programmierbaren Frames soll der Variablen R15 zugewiesen werden. |
| `$P_PFRAME[X,TR]=25` | Der Verschiebewert TR in X des aktuellen programmierbaren Frames soll verändert werden. Ab sofort gilt X25. |

## Bedeutung

| | |
|---|---|
| `$P_UIFRNUM`: | `Mit dieser Variablen wird automatisch der Bezug zur aktuell gültigen einstellbaren Nullpunktverschiebung hergestellt.` |
| `P_UIFR[n,…,…]`: | `Durch Angabe der Framenummer n greifen Sie auf den einstellbaren Frame Nr. n zu.` |
| | `Angabe der Komponente, die gelesen oder verändert werden soll:` |
| `TR`: | `TR Translation` |
| `FI`: | `FI Translation Fine` |
| `RT`: | `RT Rotation` |
| `SC`: | `SC Scale Maßstabsveränderung` |
| `MI`: | `MI Spiegelung` |
| `X, Y, Z`: | `Zusätzlich (siehe Beispiele) wird die entsprechende Achse X, Y, Z angegeben.` |

### Wertebereich für Drehung RT

Drehung um 1. Geometrieachse: -180° bis +180°
Drehung um 2. Geometrieachse: -90° bis +90°
Drehung um 3. Geometrieachse: -180° bis +180°

## Beschreibung

### Frame aufrufen

Durch Angabe der Systemvariablen $P_UIFRNUM können Sie direkt auf die mit $P_UIFR bzw. G54, G55, … aktuell eingestellte Nullpunktverschiebung zugreifen ($P_UIFRNUM enthält die Nummer des aktuell eingestellten Frames).

Alle anderen gespeicherten einstellbaren Frames $P_UIFR rufen Sie durch Angabe der entsprechenden Nummer $P_UIFR[n] auf.

Für vordefinierte Framevariable und eigendefinierte Frames geben Sie den Namen an, z. B. $P_IFRAME.

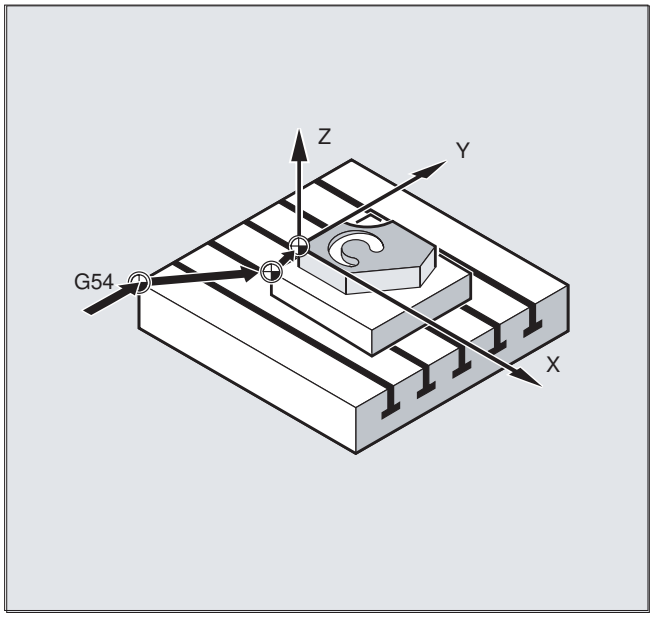
### Daten aufrufen

In den eckigen Klammern stehen Achsname und Framekomponente des Wertes, auf den Sie zugreifen oder den Sie verändern wollen, z. B. [X, RT] oder [Z, MI].

### 3.8.2.3 Rechnen mit Frames

Im NC-Programm kann ein Frame einem anderen Frame zugewiesen oder Frames miteinander verkettet werden.

Frameverkettungen eignen sich z. B. für die Beschreibung mehrerer Werkstücke, die auf einer Palette angeordnet sind und in einem Fertigungsablauf bearbeitet werden sollen.




Für die Beschreibung von Palettenaufgaben könnten die Framekomponenten z. B. nur bestimmte Teilwerte enthalten, durch deren Verkettung verschiedene Werkstücknullpunkte generiert werden.

## Beispiele

### Zuweisungen

| **Programmconde** | **Kommentar** |
|---|---|
| `DEF FRAME EINSTELLUNG_1` | `; Definition einer lokalen Framevariablen` |
| `EINSTELLUNG_1 = CTRANS(X,10)` | `; Zuweisung des Ergebnisses einer Funktion an die Framevariable` |
| `$P_PFRAME = EINSTELLUNG_1` | `; Zuweisung der Framevariablen an den Aktuellen Frame` |
| `DEF FRAME EINSTELLUNG_4` | `; Definition einer lokalen Framevariablen` |
| `EINSTELLUNG_4 = $P_PFRAME` | `; Zwischenspeicherung des Aktuellen Frames in der Framevariablen` |
| `...` | |
| `$P_PFRAME = EINSTELLUNG_4` | `; Rücklesen des Aktuellen Frames aus der Frameva-riablen` |

### Verkettungen

Die Frames werden in der programmierten Reihenfolge durch den Operator : miteinander verkettet. Die Framekomponenten wie z. B. Verschiebungen, Drehungen usw. werden nacheinander additiv ausgeführt.

| **Programmconde** | **Kommentar** |
|---|---|
| `$P_IFRAME = $P_UIFR[15] : $P_UIFR[16]` | `; Zuweisung des Ergebnis-Frames aus der Verkettung der`<br>`; beiden Einstellbaren Datenhaltungsframes an den aktiven`<br>`; Einstellbaren Gesamt-Frame.`<br>`; Anwendungsbeispiel:`<br>`; $P_UIFR[15]: Verschiebung`<br>`; $P_UIFR[16]: Drehung` |
| `$P_UIFR[3] = $P_UIFR[4] : $P_UIFR[5]` | `; Zuweisung des Ergebnis-Frames aus der Verkettung der`<br>`; beiden Einstellbaren Datenhaltungsframes an einen`<br>`; anderen Einstellbaren Datenhaltungsframe` |

## 3.8.2.4 Definition von Framevariablen (DEF FRAME)

Neben den vordefinierten Framevariablen können auch eigene Framevariablen definiert werden. Die selbstdefinierten Framevariablen sind Anwendervariablen vom Typ FRAME. Der Name des Frames kann im Rahmen der Regeln für Anwendervariablen frei vergeben werden.

Mit den Funktionen CTRANS, CROT, CSCALE, CMIRROR können die selbstdefinierten Framevariablen Werten zugewiesen werden.

### Syntax

```
DEF FRAME <Name>
```

### Bedeutung

| | |
|---|---|
| `DEF FRAME`: | Anwendervariablen vom Typ FRAME definieren. |
| <Name>: | Name der Framevariablen |

### Beispiel

Definition einer Framevariablen "PALETTE" und Zuweisung von Verschiebungs- und Drehungswerten:

| **Programmconde** | **Kommentar** |
|---|---|
| `DEF FRAME PALETTE` | `; Framevariable PALETTE definieren` |
| `PALETTE = CTRANS(...) : CROT(...)` | `; Zuweisung des Ergebnisframes der Verkettung von`<br>`; Verschiebung und Drehung an die Framevariable PALETTE` |

## 3.8.3 Grob- und Feinverschiebung (CTRANS, CFINE)

### Feinverschiebung

Eine Feinverschiebung `CFINE(...)` kann auf folgende Frames angewandt werden:

- Einstellbare Frames: $P_UIFR bzw. $P_IFRAME
- Basisframes: $P_NCBFR[<n>], $P_CHBFR[<n>] bzw. $P_CHBFRAMES[<n>] oder $P_ACTBFRAME
- Programmierbares Frame: $P_PFRAME

Die Feinverschiebung eines Frames wird mit dem Befehl `CFINE(...)` programmiert.

### Grobverschiebung

Die Grobverschiebung `CTRANS(...)` kann auf alle Frames angewandt werden.

### Gesamtverschiebung

Die Gesamtverschiebung ergibt sich aus der Addition von Grob- und Feinverschiebung.

## Maschinendaten

### Freigabe der Feinverschiebung

Die Feinverschiebung wird freigegeben mit dem Maschinendatum:

MD18600 $MN_MM_FRAME_FINE_TRANS = 1

## Syntax

### Feinverschiebung

- Gesamtframe
  - `<Frame> = CFINE(<K_1>,<Wert>)`
  - `<Frame> = CFINE(<K_1>,<Wert>, <K_2>, <Wert>)`
  - `<Frame> = CFINE(<K_1>,<Wert>, <K_2>, <Wert>, <K_3>, <Wert>)`
- Frame-Komponente
  - `<Frame>[<n>, <K_1>, FI] = <Wert>`

### Grobverschiebung

- Gesamtframe
  - `<Frame> = CTRANS(<K_1>,<Wert>)`
  - `<Frame> = CTRANS(<K_1>,<Wert>, <K_2,<Wert>)`
  - `<Frame> = CTRANS(<K_1>,<Wert>, <K_2,<Wert>, <K_3,<Wert>)`
- Frame-Komponente
  - `<Frame>[<n>,<K_1>,TR] = <Wert>`

Speziell für den Programmierbaren Frame `$P_PFRAME`:

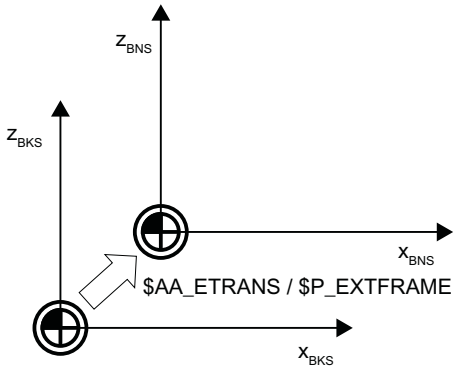
- `TRANS <K_1> <Wert>`
- `TRANS <K_1> <Wert> <K_2> <Wert>`
- `TRANS <K_1> <Wert> <K_2> <Wert> <K_3> <Wert>`

## Bedeutung

| | |
|---|---|
| `<Frame>`: | Frame z.B. Einstellbarer Frame der Datenhaltung `$P_UIFR[<n>]` |
| `CFINE`: | Feinverschiebung, addive Verschiebung. |
| `CTRANS`: | Grobverschiebung, absolute Verschiebung |
| `TRANS`: | Nur Programmierbarer Frame: Grobverschiebung, absolute Verschiebung |
| `<K_n>`: | Koordinatenachsen X, Y, Z |
| `<Wert>`: | Verschiebungswert |

## 3.8.4 Externe Nullpunktverschiebung ($AA_ETRANS)

Die Externen Nullpunktverschiebung ist eine lineare Verschiebung zwischen Basiskoordinatensystem (BKS) und Basisnullpunktsystem (BNS).




Die Externen Nullpunktverschiebung mittels $AA_ETRANS wirkt, abhängig von der Maschinendaten-Parametrierung, auf zwei Arten:

1. Die Systemvariablen $AA_ETRANS wirkt nach Aktivierung durch das NC/PLC-Nahtstellensignal direkt als Verschiebungswert
2. Der Wert der Systemvariablen $AA_ETRANS wird nach Aktivierung durch das NC/PLC-Nahtstellensignal in die aktiven Systemframes $P:EXTFRAME und den Datenhaltungsframe $P_EXTFR übernommen. Anschließend wird der aktive Gesamtframe $P_ACTFRAME neu berechnet.

### Maschinendaten

Im Zusammenhang mit der Systemvariablen $AA_ETRANS sind zwei Vorgehensweisen zu unterscheiden, die über das folgende Maschinendatum ausgewählt werden:

MD28082 $MC_MM_SYSTEM_FRAME_MASK,Bit1 = <Wert>

| <Wert> | Bedeutung |
|---|---|
| 0 | Funktion: Direktes Schreiben von $AA_ETRANS[<Achse>] durch PLC, HMI oder NC-Programm.<br>Freigabe zum Herausfahren der Nullpunktverschiebung von $AA_ETRANS[<Achse>] im nächstmöglichen Verfahrsatz: DB31, ... DBX3.0 |
| 1 | Funktion: Aktivierung des aktiven Systemframes $P:EXTFRAME und des Datenhaltungsframes $P_EXTFR<br>Freigabe zum Herausfahren der Nullpunktverschiebung von $AA_ETRANS[<Achse>] durch: DB31, ... DBX3.0. Daraufhin erfolgt im Kanal:<br>• Stop aller Verfahrbeweungen im Kanal (außer Kommando- und PLC-Achsen)<br>• Vorlaufstop mit anschließendem Reorganisieren (STOPRE)<br>• Grobverschiebung aktiver Frame $P_EXTFRAME[<Achse>] = $AA_ETRANS[<Achse>]<br>• Grobverschiebung Datenhaltungsframe $P_EXTFR[<Achse>] = $AA_ETRANS[<Achse>]<br>• Neuberechnung das aktiven Gesamtframes $P_ACTFRAME<br>• Herausfahren der Verschiebung in den programmierten Achsen.<br>• Fortsetzen der unterbrochenen Verfahrbewegung bzw. des NC-Programms |

### Programmierung

- Syntax
  `$AA_ETRANS[<Achse>] = <Wert>`
- Bedeutung

| | |
|---|---|
| `$AA_ETRANS`: | Systemvariable zum Zwischenspeichern der externen Nullpunktverschiebung |
| `<Achse>`: | Kanalachse |
| `<Wert>`: | Verschiebungswert |

### NC/PLC-Nahtstellensignal

DB31, ... DBX3.0 = 0 → 1 ⇒ $P_EXTFRAME[<Achse>] = $P_EXTFR[<Achse>] = $AA_ETRANS[<Achse>]

## 3.8.5 Istwertsetzen mit Verlust des Referenzierstatus (PRESETON)

Die Prozedur PRESETON() setzt für eine oder mehrere Achsen einen neuen Istwert im Maschinenkoordinatensystem (MKS). Dies entspricht einer Nullpunktverschiebung des MKS der Achse. Die Achse wird dadurch nicht verfahren.

Durch PRESETON wird ein Vorlaufstop mit Synchronisation ausgelöst. Die Istposition wird der Achse erst im Stillstand zugewiesen.

Ist die Achse bei PRESETON dem Kanal nicht zugeordnet, ist das weitere Vorgehen abhängig von der achsspezifischen Projektierung des Achstausverhaltens:

MD30552 $MA_AUTO_GET_TYPE

### Referenzierstatus

Durch das Setzen eines neuen Istwertes im Maschinenkoordinatensystem wird der Referenzierstatus der Maschinenachse zurückgesetzt:

DB31, ... DBX60.4 / .5 = 0 (Referenziert / Synchronisiert Messsystem 1 / 2)

Es wird empfohlen, PRESETON nur bei Achsen ohne Referenzpunktpflicht zu verwenden.

Zum Wiederherstellen des ursprünglichen Maschinenkoordinatensystems muss das Messsystem der Maschinenachse, z.B. durch Referenzpunktfahren aus dem Teileprogramm (G74), erneut referenziert werden.

**⚠ VORSICHT**

**Verlust des Referenzierstatus**

Durch das Setzen eines neuen Istwerts im Maschinenkoordinatensystem mit PRESETON wird der Referenzierstatus der Maschinenachse auf "nicht referenziert / synchronissiert" zurückgesetzt.

## Programmierung

### Syntax

```
PRESETON(<Achse_1>, <Wert_1> [, <Achse_2>, <Wert_2>, ... <Achse_8>,
<Wert_8>])
```

### Bedeutung

| | | |
|---|---|---|
| PRESETON: | Istwertsetzen mit Verlust des Referenzierstatus | |
| | Vorlaufstopp: | ja |
| | Alleine im Satz: | ja |
| <Achse_x>: | Maschinenachsname | |
| | Typ: | AXIS |
| | Wertebereich: | im Kanal definierte Maschinenachsnamen |
| <Wert_x>: | Neuer Istwert der Maschinenachse im Maschinenkoordinatensystem (MKS)<br>Die Eingabe erfolgt im aktuell gültigen Maßsystem (inch / metrisch)<br>Eine aktive Durchmesserprogrammierung (DIAMON) wird berücksichtigt | |
| | Typ: | REAL |

## Weitere Informationen

### PRESETON in NC-Programmen

Eine ausführliche Beschreibung von PRESETON in NC-Programmen findet sich im Funktionshandbuch Basisfunktionen.

### PRESETON in Synchronaktionen

Eine ausführliche Beschreibung von PRESETON in Synchronaktionen findet sich im Funktionshandbuch Synchronaktionen.

## 3.8.6 Istwertsetzen ohne Verlust des Referenzierstatus (PRESETONS)

Die Prozedur PRESETONS() setzt für eine oder mehrere Achsen einen neuen Istwert im Maschinenkoordinatensystem (MKS). Dies entspricht einer Nullpunktverschiebung des MKS der Achse. Die Achse wird dadurch nicht verfahren.

Durch PRESETONS wird ein Vorlaufstop mit Synchronisation ausgelöst. Die Istposition wird der Achse erst im Stillstand zugewiesen.

Ist die Achse bei PRESETONS dem Kanal nicht zugeordnet, ist das weitere Vorgehen abhängig von der achsspezifischen Projektierung des Achstauschverhaltens:

MD30552 $MA_AUTO_GET_TYPE

### Referenzierstatus

Durch das Setzen eines neuen Istwerts im Maschinenkoordinatensystem (MKS) mit PRESETONS wird der Referenzierstatus der Maschinenachse **nicht** verändert.

### Voraussetzungen

- **Gebertyp**
  PRESETONS ist nur bei folgenden Gebertypen des aktiven Messsystems möglich:
  - MD30240 $MA_ENC_TYPE[<Messsystem>] = 0 (Simulierter Geber)
  - MD30240 $MA_ENC_TYPE[<Messsystem>] = 1 (Rohsignalgeber)
- **Referenzier-Mode**
  PRESETONS ist nur bei folgenden Referenzier-Modes des aktiven Messsystems möglich:
  - MD34200 $MA_ENC_REFP_MODE[<Messsytem>] = 0 (kein Referenzpunktfahren möglich )
  - MD34200 $MA_ENC_REFP_MODE[<Messsytem>] = 1 (Referenzieren von inkrementellen, rotatorischen oder linearen Messsystemen: Nullimpuls auf der Geberspur)

## Programmierung

### Syntax

```
PRESETONS(<Achse_1>, <Wert_1> [, <Achse_2>, <Wert_2>, ... <Achse_8>,
<Wert_8>])
```

### Bedeutung

| | | |
|---|---|---|
| PRESETONS: | Istwertsetzen ohne Verlust des Referenzierstatus | |
| | Vorlaufstopp: | ja |
| | Alleine im Satz: | ja |
| <Achse_x>: | Maschinenachsname | |
| | Typ: | AXIS |
| | Wertebereich: | im Kanal definierte Maschinenachsnamen |

| | | |
|---|---|---|
| <Wert_x>: | Neuer aktueller Istwert der Maschinenachse im Maschinenkoordinatensystem (MKS)<br>Die Eingabe erfolgt im aktiven Maßsystem (inch / metrisch)<br>Eine aktive Durchmesserprogrammierung (DIAMON) wird berücksichtigt | |
| | Typ: | REAL |

## Weitere Informationen

### PRESETONS in NC-Programmen

Eine ausführliche Beschreibung von PRESETONS in NC-Programmen findet sich im Funktionshandbuch Basisfunktionen.

### PRESETONS in Synchronaktionen

Eine ausführliche Beschreibung von PRESETONS in Synchronaktionen findet sich im Funktionshandbuch Synchronaktionen.

## 3.8.7 Frame-Berechnung aus 3 Messpunkten im Raum (MEAFRAME)

Die Funktion MEAFRAME wird zur Unterstützung der Messzyklen eingesetzt. Sie berechnet den Frame aus drei idealen und den korrespondierenden gemessenen Punkten.

Wird ein Werkstück für die Bearbeitung positioniert, ist seine Position relativ zum kartesischen Maschinenkoordinatensystem bezüglich seiner Idealposition i. A. sowohl verschoben als auch gedreht. Für exakte Bearbeitung oder Messung ist entweder eine kostspielige physikalische Justierung oder Änderung der Bewegungen im Teileprogramm nötig.

Ein Frame kann durch Abtasten dreier Punkte im Raum festgelegt werden, deren Idealpositionen bekannt sind. Abgetastet wird mit einem Berührungs- oder optischen Sensor, der spezielle, auf der Trägerplatte präzise fixierte Löcher oder Messkugeln berührt.

## Syntax

```
MEAFRAME(<Ideal-Punkte>,<Messpunkte>,<Qualität>)
```

## Bedeutung

| | |
|---|---|
| MEAFRAME: | Funktionsaufruf |
| <Ideal-Punkte>: | 2-dim. REAL-Feld, das die drei Koordinaten der Ideal-Punkte enthält |
| <Messpunkte>: | 2-dim. REAL-Feld, das die drei Koordinaten der gemessenen Punkte enthält |

| | | | |
|---|---|---|---|
| <Qualität>: | Variable, mit der Informationen zur Qualität der FRAME-Berechnung zurückgegeben werden | | |
| | Typ: | VAR REAL | |
| | Wert: | -1 | Die idealen Punkte liegen nahezu auf einer Geraden: Der Frame konnte nicht berechnet werden. Die zurückgegebene FRAME-Variable enthält einen neutralen Frame. |
| | | -2 | Die Messpunkte liegen nahezu auf einer Geraden: Der Frame konnte nicht berechnet werden. Die zurückgegebene FRAME-Variable enthält einen neutralen Frame. |
| | | -4 | Die Berechnung der Rotationsmatrix schlägt aus einem anderen Grund fehl. |
| | | ≥ 0.0 | Summe der Verzerrungen (Abstände zwischen den Punkten), die zur Überführung des gemessenen Dreiecks in ein zum idealen Dreieck kongruentes benötigt wird. |

**Hinweis**

**Qualität der Messung**

Damit die gemessenen den idealen Koordinaten mit einer kombinierten Rotation/Translation zugeordnet werden können, muss das von den Messpunkten aufgespannte Dreieck kongruent zum idealen Dreieck sein. Dies wird bewerkstelligt von einem Kompensationsalgorithmus, der die Summe der Quadrate der Abweichungen minimiert, die das gemessene in das ideale Dreieck überführen.

Die effektiv benötigte Verzerrung der Messpunkte kann als Indikator für die Qualität der Messung dienen und wird deshalb als zusätzliche Variable von MEAFRAME ausgegeben.

**Hinweis**

Das von MEAFRAME erzeugte Frame kann durch die Funktion ADDFRAME in ein anderes Frame in der Frame-Kette transformiert werden (siehe Beispiel "Verkettung mit ADDFRAME").

## Beispiele

### Beispiel 1:

Teileprogramm 1:

**Programmcode**

```
...
DEF FRAME CORR_FRAME
```

Setzen von Messpunkten:

| **Programmcode** | **Kommentar** |
|---|---|
| DEF REAL IDEAL_POINT[3,3]=<br>SET(10.0,0.0,0.0,0.0,10.0,0.0,0.0,0.0,10.0) | |
| DEF REAL MEAS_POINT[3,3]=<br>SET(10.1,0.2,-0.2,-0.2,10.2,0.1,-0.2,0.2,9.8) | ; Für Test. |
| DEF REAL FIT_QUALITY=0 | |
| DEF REAL ROT_FRAME_LIMIT=5 | ; Erlaubt max. 5 Grad-Verdrehung der Teileposition. |
| DEF REAL FIT_QUALITY_LIMIT=3 | ; Erlaubt max. 3 mm-Verschiebung zwischen<br>dem idealen und dem gemessenen Dreieck. |
| DEF REAL SHOW_MCS_POS1[3] | |
| DEF REAL SHOW_MCS_POS2[3] | |
| DEF REAL SHOW_MCS_POS3[3] | |

| **Programmcode** | **Kommentar** |
|---|---|
| N100 G01 G90 F5000 | |
| N110 X0 Y0 Z0 | |
| N200 CORR_FRAME=MEAFRAME(IDEAL_POINT,MEAS_POINT,FIT_QUALITY) | |
| N230 IF FIT_QUALITY < 0 | |
| SETAL(65000) | |
| GOTOF NO_FRAME | |
| ENDIF | |
| N240 IF FIT_QUALITY > FIT_QUALITY_LIMIT | |
| SETAL(65010) | |
| GOTOF NO_FRAME | |
| ENDIF | |
| N250 IF CORR_FRAME[X,RT] > ROT_FRAME_LIMIT | ; Begrenzung des 1. RPY- Winkels. |
| SETAL(65020) | |
| GOTOF NO_FRAME | |
| ENDIF | |
| N260 IF CORR_FRAME[Y,RT] > ROT_FRAME_LIMIT | ; Begrenzung des 2. RPY-Winkels. |
| SETAL(65021) | |
| GOTOF NO_FRAME | |
| ENDIF | |
| N270 IF CORR_FRAME[Z,RT] > ROT_FRAME_LIMIT | ; Begrenzung des 3. RPY- Winkels. |
| SETAL(65022) | |
| GOTOF NO_FRAME | |
| ENDIF | |
| N300 $P_IFRAME=CORR_FRAME | ; Abtast-Frame mit einem setzbaren Frame aktivieren. |

| **Programmcode** | **Kommentar** |
|---|---|
| | ; Frame prüfen durch Positionieren der Geometrieachsen auf die idealen Punkte. |
| N400 X=IDEAL_POINT[0,0] Y=IDEAL_POINT[0,1] Z=IDEAL_POINT[0,2] | |
| N410 SHOW_MCS_POS1[0]=$AA_IM[X] | |
| N420 SHOW_MCS_POS1[1]=$AA_IM[Y] | |
| N430 SHOW_MCS_POS1[2]=$AA_IM[Z] | |
| N500 X=IDEAL_POINT[1,0] Y=IDEAL_POINT[1,1] Z=IDEAL_POINT[1,2] | |
| N510 SHOW_MCS_POS2[0]=$AA_IM[X] | |
| N520 SHOW_MCS_POS2[1]=$AA_IM[Y] | |
| N530 SHOW_MCS_POS2[2]=$AA_IM[Z] | |
| N600 X=IDEAL_POINT[2,0] Y=IDEAL_POINT[2,1] Z=IDEAL_POINT[2,2] | |
| N610 SHOW_MCS_POS3[0]=$AA_IM[X] | |
| N620 SHOW_MCS_POS3[1]=$AA_IM[Y] | |
| N630 SHOW_MCS_POS3[2]=$AA_IM[Z] | |
| N700 G500 | ; Setzbaren Frame deaktivieren, damit Nullframe (kein Wert eingetragen vorbesetzt). |
| No_FRAME | ; Setzbaren Frame deaktivieren, damit Nullframe (kein Wert eingetragen) vorbesetzt. |
| M0 | |
| M30 | |

### Beispiel 2: Verkettung von Frames

#### Verkettung von MEAFRAME für Korrekturen

Die Funktion MEAFRAME liefert ein Korrektur-Frame. Wird dieser Korrektur-Frame mit dem einstellbaren Frame $P_UIFR[1] verkettet, der bei Aufruf der Funktion aktiv war (z. B. G54), so erhält man einen einstellbaren Frame für weitere Umrechnungen zum Verfahren oder Bearbeiten.

#### Verkettung mit ADDFRAME

Soll dieser Korrektur-Frame in der Frame-Kette an einer anderen Stelle wirken oder sind vor dem einstellbaren Frame noch andere Frames aktiv, dann kann die Funktion ADDFRAME zum Einketten in einem der Kanal-Basisframes oder einem Systemframe genutzt werden.

In den Frames darf hierbei nicht aktiv sein:

- Spiegelung mit MIRROR
- Skalierung mit SCALE

Die Eingangsparameter für Soll- und Istwerte sind die Werkstückkoordinaten. Im Grundsystem der Steuerung sind diese Koordinaten stets metrisch oder in Inch (G71/G70) und als radiusbezogenes (DIAMOF) Maß anzugeben.

**Weitere Informationen** zu ADDFRAME siehe Funktionshandbuch Basisfunktionen.

## 3.8.8 Globale Frames

Globale Frames gibt es pro Steuerung nur einmal für alle Kanäle. Globale Frames können von allen Kanälen aus geschrieben und gelesen werden. Die Aktivierung der globalen Frames erfolgt im jeweiligen Kanal.

Durch globale Frames können **Kanalachsen und Maschinenachsen** mit Verschiebungen skaliert und gespiegelt werden.

#### Geometrische Zusammenhänge und Frameketten

Bei globalen Frames existiert kein geometrischer Zusammenhang zwischen den Achsen. Deshalb können keine Drehungen und keine Programmierung von Geometrieachsbezeichnern ausgeführt werden.

Auf globale Frames lassen sich keine Rotationen anwenden. Die Programmierung einer Rotation wird mit dem Alarm 18310 "Kanal %1 Satz %2 Frame: Rotation unzulässig" abgelehnt.

Die Verkettung von globalen Frames und kanalspezifischen Frames ist möglich. Der resultierende Frame enthält alle Frameanteile inklusive der Rotationen für alle Achsen. Die Zuweisung eines Frames mit Rotationsanteilen an einen globalen Frame wird mit dem Alarm "Frame: Rotation unzulässig" abgelehnt.

### Globale Frames

#### Globale Basisframes $P_NCBFR[n]

Es können bis zu 8 globale Basisframes projektiert werden:

Gleichzeitig können kanalspezifische Basisframes vorhanden sein.

Globale Basisframes können von allen Kanälen einer Steuerung geschrieben und gelesen werden. Beim Schreiben von globalen Frames ist vom Anwender für eine Kanalkoordinierung Sorge zu tragen. Dies kann z. B. durch Wait-Marken (WAITMC) realisiert werden.

**Hinweis**

**Maschinenhersteller**

Die Anzahl von globalen Basisframes wird über Maschinendaten projektiert.

**Weitere Informationen:** Funktionshandbuch Basisfunktionen

#### Einstellbare Frames $P_UIFR[n]

Alle einstellbaren Frames G500, G54...G599 können entweder global oder kanalspezifisch projektiert werden.

**Hinweis**

**Maschinenhersteller**

Alle einstellbaren Frames können mit Hilfe des Maschinendatums MD18601 $MN_MM_NUM_GLOBAL_USER_FRAMES zu globalen Frames umprojektiert werden.

Als Achsbezeichner bei den Frame-Programmbefehlen können Kanalachsbezeichner und Maschinenachsbezeichner verwendet werden. Die Programmierung von Geometrieachsbezeichnern wird mit einem Alarm abgelehnt.

### 3.8.8.1 Kanalspezifische Frames ($P_CHBFR, $P_UBFR)

Einstellbare Frames oder Basisframes können über das Teileprogramm und über BTSS von der Bedienung und von der PLC geschrieben und gelesen werden.

Die Feinverschiebung ist auch für die globalen Frames möglich. Die Unterdrückung von globalen Frames erfolgt ebenso wie bei kanalspezifischen Frames über G53, G153, SUPA und G500.

**Maschinenhersteller**

Über das Maschinendatum MD28081 $MC_MM_NUM_BASE_FRAMES kann die Anzahl der Basisframes im Kanal projektiert werden. Die Standardkonfiguration ist so ausgelegt, dass es mindestens ein Basisframe pro Kanal gibt. Maximal sind 8 Basisframes pro Kanal möglich. Zusätzlich zu den 8 Basisframes im Kanal kann es noch 8 NCU-globale Basisframes geben.

## Kanalspezifische Frames

**$P_CHBFR[n]**

Über die Systemvariable $P_CHBFR[n] können die Basisframes gelesen und geschrieben werden. Beim Schreiben eines Basisframes wird der verkettete Gesamt-Basisframe nicht aktiviert, sondern die Aktivierung erfolgt erst mit der Ausführung einer G500, G54 ... G599-Anweisung. Die Variable dient vorwiegend als Speicher für Schreibvorgänge auf das Basisframe von HMI oder PLC. Diese Frame-Variablen werden über die Datensicherung gesichert.

**Erster Basisframe im Kanal**

Ein Schreiben auf die vordefinierte Variable $P_UBFR aktiviert den Basisframe mit dem Feldindex 0 nicht gleichzeitig, sondern die Aktivierung erfolgt erst mit der Ausführung einer G500, G54 ... G599-Anweisung. Die Variable kann auch im Programm geschrieben und gelesen werden.

**$P_UBFR**

$P_UBFR ist identisch mit $P_CHBFR[0]. Standardmäßig gibt es immer einen Basisframe im Kanal, so dass die Systemvariable kompatibel zu älteren Ständen ist. Gibt es kein kanalspezifisches Basisframe, wird beim Schreiben oder Lesen der Alarm "Frame: Anweisung unzulässig" ausgegeben.

### 3.8.8.2 Im Kanal wirksame Frames

Im Kanal wirksame Frames werden vom Teileprogramm über die betreffenden Systemvariablen dieser Frames eingegeben. Hierzu gehören auch Systemframes. Über diese Systemvariablen kann im Teileprogramm das aktuelle Systemframe gelesen und geschrieben werden.

## Aktuelle im Kanal wirksame Frames

### Übersicht

| | |
|---|---|
| **Aktuelle Systemframes** | für: |
| $P_PARTFRAME | TCARR und PAROT |
| $P_SETFRAME | Istwertsetzen und Ankratzen |
| $P_EXTFRAME | Externe Nullpunktverschiebung |
| **$P_NCBFRAME[n]** | Aktuelle globale Basisframes |
| **$P_CHBFRAME[n]** | Aktuelle Kanal-Basisframes |
| **$P_BFRAME** | Aktueller 1. Basisframe im Kanal |
| **$P_ACTBFRAME** | Gesamt-Basisframe |
| **$P_CHBFRMASK und $P_NCBFRMASK** | Gesamt-Basisframe |
| **$P_IFRAME** | Aktueller einstellbarer Frame |
| **Aktuelle Systemframes** | für: |
| $P_TOOLFRAME | TOROT und TOFRAME |
| $P_WPFRAME | Werkstückbezugspunkte |
| $P_TRAFRAME | Transformationen |
| **$P_PFRAME** | Aktueller programmierbarer Frame |
| **Aktuelles Systemframe** | für: |
| $P_CYCFRAME | Zyklen |
| **P_ACTFRAME** | Aktueller Gesamtframe |
| **FRAME-Kettung** | Aktuelles Frame setzt sich aus dem Gesamt-Basisframe zusammen |

#### $P_NCBFRAME [n] Aktuelle globale Basisframes

Über die Systemvariable $P_NCBFRAME[n] können die aktuellen globalen Basisframe-Feldelemente gelesen und geschrieben werden. Das resultierende Gesamt-Basisframe wird durch den Schreibvorgang im Kanal eingerechnet.

Der geänderte Frame wird nur in dem Kanal, in dem der Frame programmiert wurde, aktiv. Soll der Frame für alle Kanäle einer Steuerung geändert werden, muss gleichzeitig $P_NCBFR[n] und $P_NCBFRAME[n] beschrieben werden. Die anderen Kanäle müssen dann noch den Frame mit z. B. G54 aktivieren. Beim Schreiben eines Basisframes wird der Gesamt-Basisframe neu berechnet.

#### $P_CHBFRAME[n] Aktuelle Kanal-Basisframes

Über die Systemvariable $P_CHBFRAME[n] können die aktuellen Kanal-Basisframe-Feldelemente gelesen und geschrieben werden. Der resultierende Gesamt-Basisframe wird durch den Schreibvorgang im Kanal eingerechnet. Beim Schreiben eines Basisframes wird der Gesamt-Basisframe neu berechnet.

#### $P_BFRAME Aktueller 1. Basisframe im Kanal

Über die vordefinierte Framevariable $P_BFRAME kann der aktuelle Basisframe mit dem Feldindex 0, der im Kanal gültig ist, im Teileprogramm gelesen und geschrieben werden. Der geschriebene Basisframe wird sofort eingerechnet.

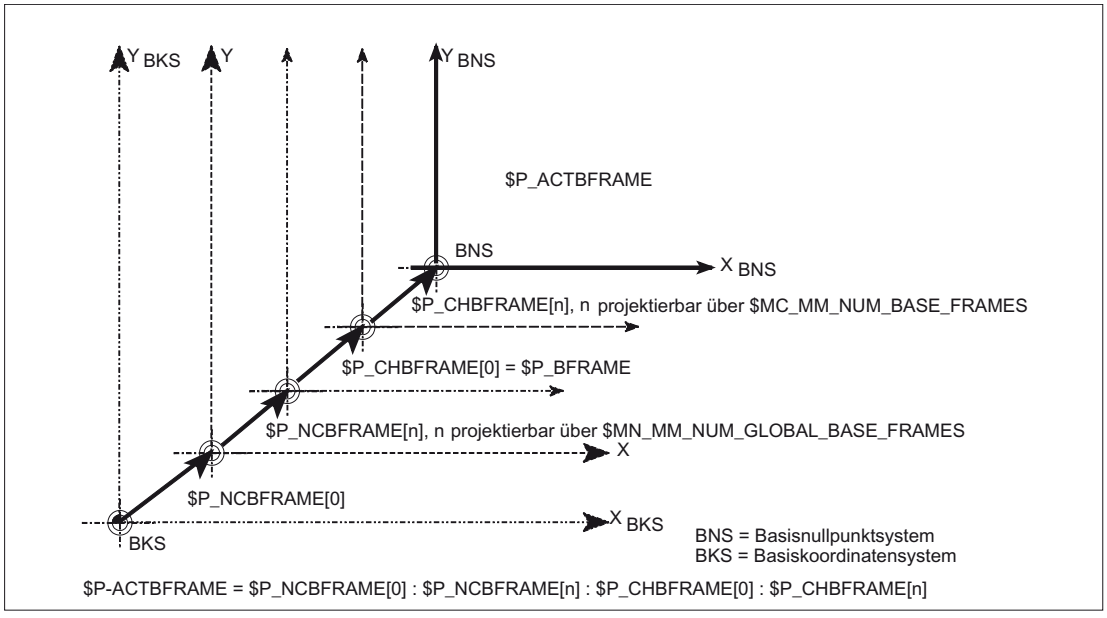
$P_BFRAME ist identisch mit $P_CHBFRAME[0]. Die Systemvariable hat standardmäßig immer einen gültigen Wert. Gibt es kein kanalspezifisches Basisframe, wird beim Schreiben oder Lesen der Alarm "Frame: Anweisung unzulässig" ausgegeben.

### $P_ACTBFRAME Gesamt-Basisframe

Die Variable $P_ACTFRAME ermittelt das verkettete Gesamt-Basisframe. Die Variable ist nur lesbar.

$P_ACTFRAME entspricht:

$P_NCBFRAME[0] : ... : $P_NCBFRAME[n] : $P_CHBFRAME[0] : ... : $P_CHBFRAME[n].




### $P_CHBFRMASK und $P_NCBFRMASK Gesamt-Basisframe

Über die Systemvariable $P_CHBFRMASK und $P_NCBFRMASK kann der Anwender auswählen, welche Basisframes er in die Berechnung des "Gesamt"-Basisframes mit einbeziehen möchte. Die Variablen können nur im Programm programmiert werden und über BTSS gelesen werden. Der Wert der Variablen wird als Bitmaske interpretiert und gibt an, welches Basisframe-Feldelement von $P_ACTFRAME in die Berechnung einfließt.

Mit $P_CHBFRMASK kann vorgegeben werden, welche kanalspezifischen Basisframes, und mit $P_NCBFRMASK, welche globalen Basisframes eingerechnet werden.

Mit der Programmierung der Variablen werden der Gesamt-Basisframe und der Gesamt-Frame neu berechnet. Nach Reset und in der Grundeinstellung ist der Wert von $P_CHBFRMASK und $P_NCBFRMASK wie folgt:

$P_CHBFRMASK = $MC_CHBFRAME_RESET_MASK

$P_NCBFRMASK = $MC_CHBFRAME_RESET_MASK

Beispiel:

```
$P_NCBFRMASK = 'H81' ;$P_NCBFRAME[0] : $P_NCBFRAME[7]
```

```
$P_CHBFRMASK = 'H11' ;$P_CHBFRAME[0] : $P_CHBFRAME[4]
```

### $P_IFRAME Aktueller einstellbarer Frame

Über die vordefinierte Framevariable $P_IFRAME kann der aktuelle einstellbare Frame, welcher im Kanal gültig ist, im Teileprogramm gelesen und geschrieben werden. Der geschriebene einstellbare Frame wird sofort eingerechnet.

Bei globalen einstellbaren Frames wirkt der geänderte Frame nur in dem Kanal, in dem der Frame programmiert wurde. Soll der Frame für alle Kanäle einer Steuerung geändert werden, muss gleichzeitig $P_UIFR[n] und $P_IFRAME beschrieben werden. Die anderen Kanäle müssen dann noch den entsprechenden Frame mit z. B. G54 aktivieren.

### $P_PFRAME Aktueller programmierbarer Frame

$P_PFRAME ist der programmierbare Frame, der sich aus der Programmierung von TRANS/ATRANS, G58/G59, ROT/AROT, SCALE/ASCALE, MIRROR/AMIRROR bzw. aus der Zuweisung von CTRANS, CROT, CMIRROR, CSCALE an den programmierbaren FRAME ergibt.

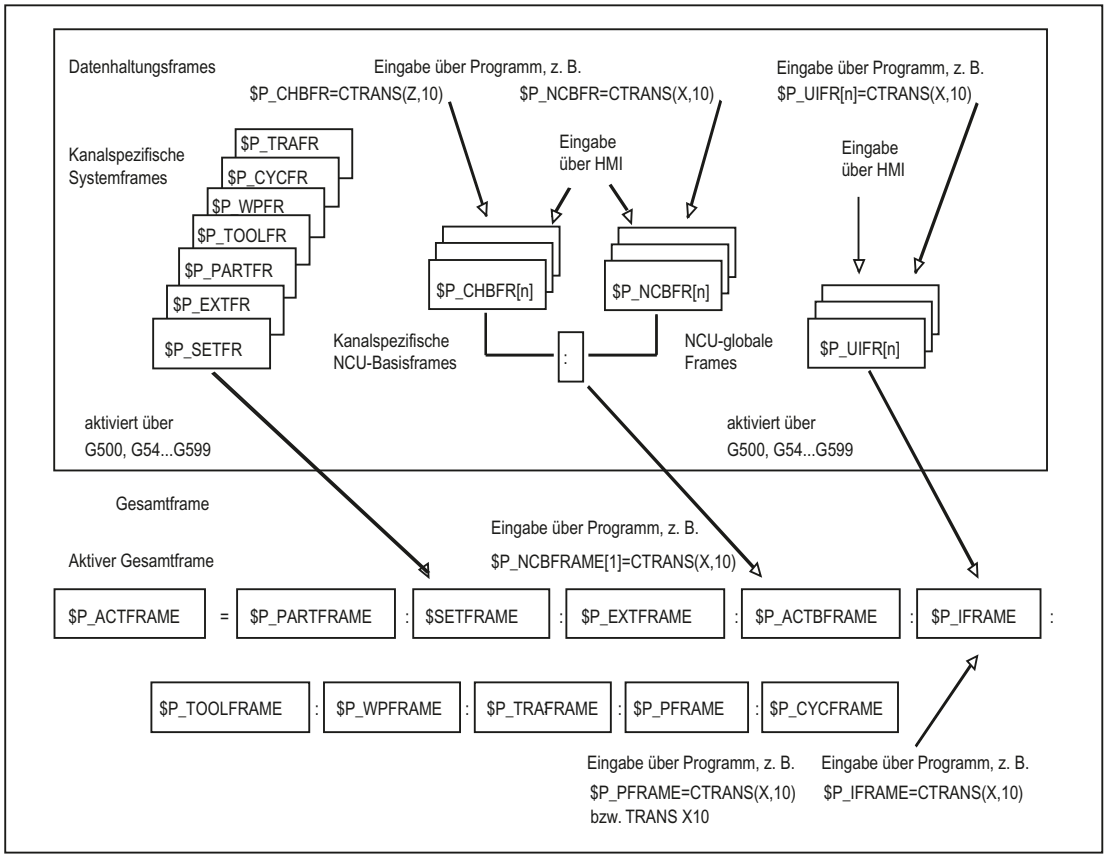
Aktuelle, programmierbare Framevariable, die den Bezug zwischen dem einstellbaren Nullpunktsystem (ENS) und dem Werkstückkoordinatensystem (WKS) herstellt.

### P_ACTFRAME Aktueller Gesamtframe

Der aktuelle resultierende Gesamtframe $P_ACTFRAME ergibt sich nun als Verkettung aller Basisframes, dem aktuellen einstellbaren Frame und dem programmierbaren Frame. Der aktuelle Frame wird immer dann aktualisiert, wenn sich ein Frameanteil ändert.

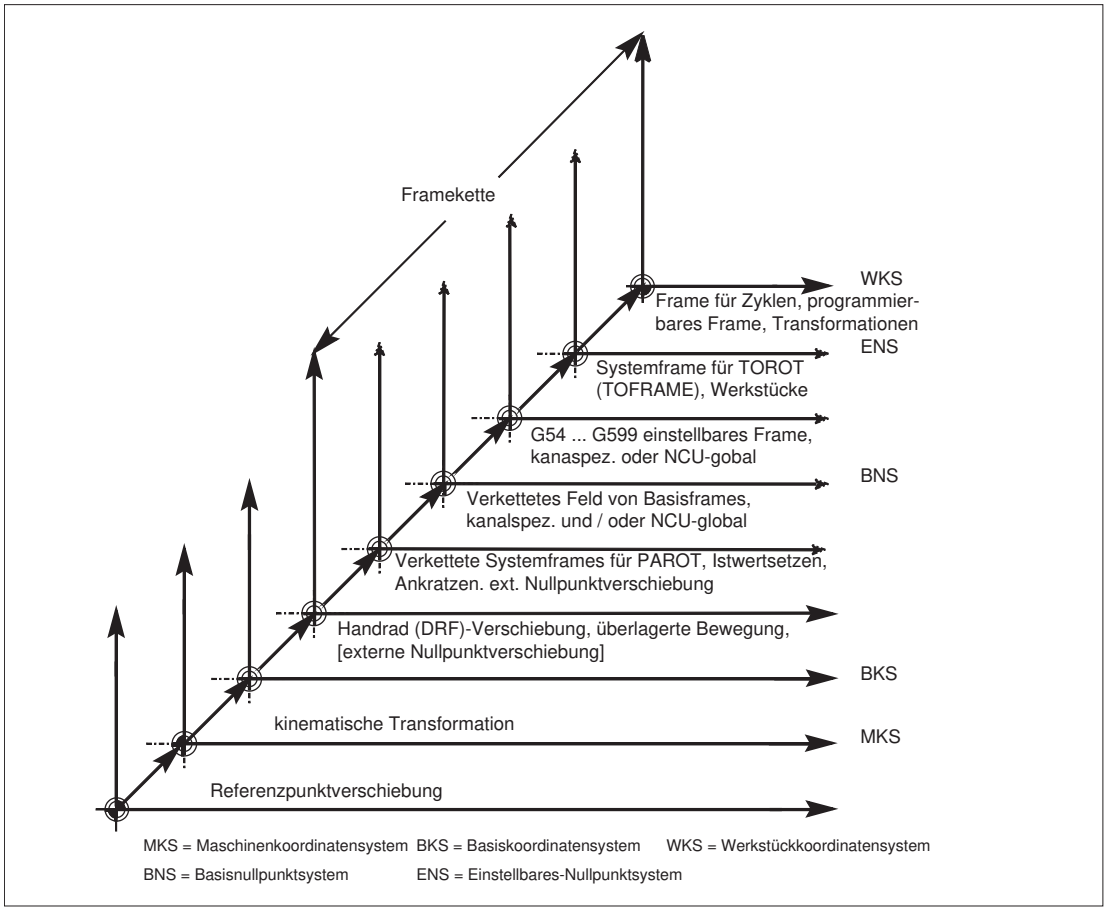
`$P_ACTFRAME` entspricht:

```
$P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME : $P_ACTBFRAME : $P_IFRAME :
$P_TOOLFRAME : $P_WPFRAME : $P_TRAFRAME : $P_PFRAME : $P_CYCFRAME
```


Datenhaltungsframes
Eingabe über Programm, z. B.
$P_CHBFR=CTRANS(Z,10)
$P_NCBFR=CTRANS(X,10)
Eingabe über Programm, z. B.
$P_UIFR[n]=CTRANS(X,10)
Kanalspezifische
Systemframes
$P_TRAFR
$P_CYCFR
$P_WPFR
$P_TOOLFR
$P_PARTFR
$P_EXTFR
$P_SETFR
Eingabe
über HMI
Eingabe
über HMI
$P_CHBFR[n]
$P_NCBFR[n]
$P_UIFR[n]
Kanalspezifische
NCU-Basisframes
:
NCU-globale
Frames
aktiviert über
G500, G54...G599
aktiviert über
G500, G54...G599
Gesamtframe
Eingabe über Programm, z. B.
$P_NCBFRAME[1]=CTRANS(X,10)
Aktiver Gesamtframe
$P_ACTFRAME
=
$P_PARTFRAME
:
$SETFRAME
:
$P_EXTFRAME
:
$P_ACTBFRAME
:
$P_IFRAME
:
$P_TOOLFRAME
:
$P_WPFRAME
:
$P_TRAFRAME
:
$P_PFRAME
:
$P_CYCFRAME
Eingabe über Programm, z. B.
$P_PFRAME=CTRANS(X,10)
bzw. TRANS X10
Eingabe über Programm, z. B.
$P_IFRAME=CTRANS(X,10)

## Frame-Kettung

Der aktuelle Frame setzt sich aus dem Gesamt-Basisframe, dem einstellbaren Frame, dem Systemframe und dem programmierbaren Frame gemäß oben angegebenen aktuellen Gesamtframe zusammen.



# 3.9 Transformationen

## 3.9.1 Allgemeine Programmierung der Transformationsarten

### Allgemeine Funktion

Zur Anpassung der Steuerung an verschiedene Maschinenkinematiken besteht die Auswahl Transformationsarten mit geeigneten Parametern zu programmieren. Über diese Parameter kann für die ausgewählte Transformation sowohl die Orientierung des Werkzeugs im Raum als auch die Orientierungsbewegungen der Rundachsen entsprechend vereinbart werden.

Bei den Drei-, Vier- und Fünf-Achs-Transformationen beziehen sich die programmierten, Positionsangaben immer auf die Spitze des Werkzeugs, welches orthogonal zur im Raum befindlichen Bearbeitungsfläche nachgeführt wird. Die kartesischen Koordinaten werden vom Basiskoordinatensystem ins Maschinenkoordinatensystem umgerechnet und beziehen sich auf die Geometrieachsen. Diese beschreiben den Arbeitspunkt. Virtuelle Rundachsen beschreiben die Orientierungen des Werkzeugs im Raum und werden mit TRAORI programmiert.

Bei der kinematischen Transformation können Positionen im kartesischen Koordinatensystem programmiert werden. Die Steuerung transformiert die mit TRANSMIT, TRACYL und TRAANG programmierten Verfahrbewegungen des kartesischen Koordinatensystems auf die Verfahrbewegungen der realen Maschinenachsen.

### Programmierung

#### Drei-, Vier- und Fünf-Achs-Transformationen TRAORI

Die vereinbarte Orientierungstransformation wird mit dem Befehl TRAORI und den drei möglichen Parametern für Trafonummer, Orientierungsvektor und Rundachsoffsets aktiviert.

```
TRAORI(Trafonummer, Orientierungsvektor, Rundachsoffsets)
```

#### Kinematische Transformationen

Zu den Kinematischen Transformationengehören die vereinbarten Transformationen
`TRANSMIT(Trafonummer)`

```
TRACYL(Arbeitsdurchmesser, Trafonummer)
```

```
TRAANG(Winkel der schräg stehenden Achse, Trafonummer)
```

#### Aktive Transformation ausschalten

Mit `TRAFOOF` kann die gerade aktive Transformation ausgeschaltet werden.

### Orientierungstransformation

#### Drei-, Vier- und Fünf- Achs-Transformationen TRAORI

Zur optimalen Bearbeitung räumlich geformter Flächen im Arbeitsraum der Maschine, benötigen Werkzeugmaschinen außer den drei Linearachsen X, Y und Z noch zusätzliche Achsen. Die zusätzlichen Achsen beschreiben die Orientierung im Raum und werden

nachfolgend Orientierungsachsen genannt. Sie stehen als Drehachsen bei vier Maschinentypen mit verschiedener Kinematik zur Verfügung.

1. Zweiachsen-Schwenkkopf, z. B. Kardanischer Werkzeugkopf mit einer Rundachse parallel zu einer Linearachse bei festem Werkzeugtisch.
2. Zweiachsen-Drehtisch, z. B. fester Schwenkkopf mit drehbarem Werkzeugtisch um zwei Achsen.
3. Einachs-Schwenkkopf und Einachs-Drehtisch, z. B. ein drehbarer Schwenkkopf mit gedrehtem Werkzeug bei drehbarem Werkzeugtisch um eine Achse.
4. Zweiachsen-Schwenkkopf und Einachs-Drehtisch, z. B. bei drehbarem Werkzeugtisch um eine Achse und ein drehbarer Schwenkkopf mit drehbarem Werkzeug um sich selbst.

Die **3- und 4-Achs-Transformationen** sind Sonderformen der 5-Achs-Transformation und werden analog zu den 5-Achs-Transformationen programmiert.

Die "**Generische 3-/4-/5-/6-Achs-Transformation**" deckt mit ihrem Funktionsumfang für rechtwinklig angeordnete Rundachsen sowie die Transformationen für den Kardanischen Fräskopf ab und kann wie jede andere Orientierungstransformation auch für diese vier Maschinentypen mit TRAORI aktiviert werden. Bei der generischen 5/6-Achs-Transformationhat die Werkzeugorientierung einen weiteren dritten Freiheitsgrad, bei dem zur Werkzeugrichtung beliebig im Raum, das Werkzeug um die eigene Achse gedreht werden kann.

**Weitere Informationen**

Funktionshandbuch Transformationen; Mehrachstransformationen

## Kinematikunabhängige Grundstellung der Werkzeugorientierung

### ORIRESET

Ist mit TRAORI eine Orientierungstransformation aktiv, dann können mit ORIRESET die Grundstellungen von bis zu 3 Orientierungsachsen mit optionalen Parametern A, B, C angegeben werden. Die Zuordnung der Reihenfolge der programmierten Parameter zu den Rundachsen erfolgt gemäß der durch die Transformation festgelegten Reihenfolge der Orientierungsachsen. Die Programmierung von ORIRESET(A, B, C) bewirkt, dass die Orientierungsachsen linear und synchron von ihrer momentanen Position zu der angegebenen Grundstellungsposition fahren.

## Kinematische Transformationen

### TRANSMIT und TRACYL

Bei Fräsbearbeitungen an Drehmaschinen kann für die vereinbarte Transformation entweder

1. eine stirnseitige Bearbeitung in der Drehaufspannung mit TRANSMIT oder
2. eine Bearbeitung von beliebig verlaufenden Nuten an zylindrischen Körpern mit TRACYL

programmiert werden.

### TRAANG

Soll die Zustellachse z. B. für die Technologie Schleifen auch schräg zustellbar sein, so kann mit TRAANG für die vereinbarte Transformation ein parametrierbarer Winkel programmiert werden.

**Kartesisches PTP-Fahren**

Zur kinematischen Transformation gehört auch das "Kartesisches PTP-Fahren" bei dem bis zu 8 unterschiedliche Gelenkstellungen STAT= programmiert werden können. Die Positionen werden im kartesischen Koordinatensystem programmiert, wobei die Bewegung der Maschine in Maschinenkoordinaten erfolgt.

**Weitere Informationen**
Funktionshandbuch Transformationen; Kinematische Transformation

## Verkettete Transformationen

Es können jeweils zwei Transformationen hintereinander geschaltet werden. Bei der hierdurch verketteten zweiten Transformation werden die Bewegungsanteile der Achsen aus der ersten Transformation übernommen.

Als erste Transformation sind möglich:

- Orientierungstransformation TRAORI
- Polartransformation TRANSMIT
- Zylindertransformation TRACYL
- Transformation Schräge Achse TRAANG

Die zweite Transformation muss Schräge Achse TRAANG sein

### 3.9.1.1 Orientierungsbewegungen bei den Transformationen

## Verfahrbewegungen und Orientierungsbewegungen

Die Verfahrbewegungen der programmierbaren Orientierungen hängen primär vom Maschinentyp ab. Bei der Drei-, Vier- und Fünf-Achs-Transformation mit TRAORI beschreiben die rotatorischen Achsen oder die schwenkbaren Linearachsen die Orientierungsbewegungen des Werkzeugs.

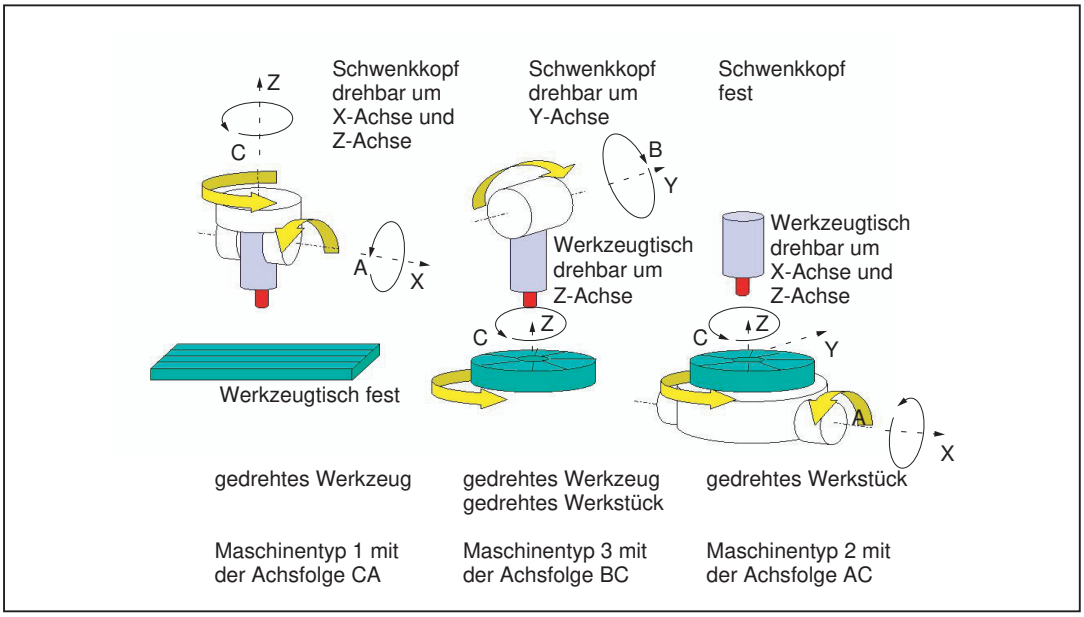
Änderungen der Positionen der an der Orientierungstransformation beteiligten Rundachsen führen zu Ausgleichsbewegungen der übrigen Maschinenachsen. Die Position der Werkzeugspitze bleibt dabei unverändert.

Orientierungsbewegungen des Werkzeugs können über die Rundachsbezeichner A…, B…, C… der virtuellen Achsen je nach Anwendung entweder durch Angabe von Euler- bzw. RPY-Winkeln oder Richtungs- bzw. Flächennormalenvektoren, Normierte Vektoren für die Drehachse eines Kegels oder für die Zwischenorientierung auf einer Kegelmantelfläche programmiert werden.

Bei der Kinematischen Transformation mit TRANSMIT, TRACYL und TRAANG transformiert die Steuerung die programmierten Verfahrbewegungen des kartesischen Koordinatensystems auf die Verfahrbewegungen der realen Maschinenachsen.

## Maschinenkinematik bei Drei-, Vier- und Fünf-Achs-Transformation TRAORI

Es kann entweder das Werkzeug oder der Werkzeugtisch mit bis zu zwei Rundachsen drehbar sein. Eine Kombination von jeweils einachsigem Schwenkkopf und Drehtisch ist auch möglich.




| Maschinentyp | Programmierung der Orientierung |
|---|---|
| Drei-Achs-Transformation Maschinentypen 1 und 2 | Programmierung der Werkzeugorientierung nur in der Ebene, die senkrecht zu der rotatorischen Achse ist. Es existieren **zwei** translatorischen Achsen (Linearachsen) und **einer** rotatorischen Achse (Rundachse). |
| Vier-Achs-Transformation Maschinentypen 1 und 2 | Programmierung der Werkzeugorientierung nur in der Ebene, die senkrecht zu der rotatorischen Achse ist. Es existieren **drei** translatorischen Achsen (Linearachsen) und **einer** rotatorischen Achse (Rundachse). |
| Fünf-Achs-Transformation Maschinentypen 3 Einachs-Schwenkkopf und Einachs-Drehtisch | Programmierung der Orientierungstransformation. Kinematik mit **drei** Linearachsen und **zwei** orthogonalen Rundachsen. Die Rundachsen sind parallel zu zwei der drei Linearachsen. Die erste Rundachse wird von zwei kartesischen Linearachsen bewegt. Sie dreht die dritte Linearachse mit dem Werkzeug. Die zweite Rundachse dreht das Werkstück. |

### Generische 5/6-AchsTransformationen

| Maschinentyp | Programmierung der Orientierungstransformation |
|---|---|
| Generische Fünf-/Sechs-Achs Transformation Maschinentypen 4 Zweiachs-Schwenkkopf mit drehbarem Werkzeug um sich selbst und Einachs-Drehtisch | Programmierung der Orientierungstransformation. Kinematik mit **drei** Linearachsen und **drei** orthogonalen Rundachsen. Die Rundachsen sind parallel zu zwei der drei Linearachsen. Die erste Rundachse wird von zwei kartesischen Linearachsen bewegt. Sie dreht die dritte Linearachse mit dem Werkzeug. Die zweite Rundachse dreht das Werkstück. Die Grundorientierung des Werkzeugs kann durch eine zusätzliche Drehung um sich selbst mit dem Drehwinkel THETA programmiert werden. |

Beim Aufruf der "Generischen Drei-, Vier- und Fünf-/Sechs-Achs Transformation" kann zusätzlich die Grundorientierung des Werkzeugs übergeben werden. Es gelten die Einschränkungen bezüglich der Richtungen der Rundachsen nicht mehr. Wenn die Rundachsen nicht exakt senkrecht aufeinander stehen oder vorhandene Rundachsen nicht exakt parallel zu den Linearachsen stehen, kann die "Generische Fünf-/Sechs-Achs Transformation" bessere Ergebnisse der Werkzeugorientierung liefern.

## Kinematische Transformationen TRANSMIT, TRACYL und TRAANG

Für Fräsbearbeitungen an Drehmaschinen oder einer schräg zustellbaren Achse beim Schleifen gelten abhängig von der Transformation im Standardfall folgende Achsanordnungen:

| TRANSMIT | Aktivierung der Polar-Transformation |
|---|---|
| stirnseitige Bearbeitung in der Drehaufspannung | eine Rundachse<br>eine Zustellachse senkrecht zur Drehachse<br>eine Längsachse parallel zur Drehachse |

| TRACYL | Aktivierung der Zylindermanteltransformation |
|---|---|
| Bearbeitung von beliebig verlaufenden Nuten an den zylindrischen Körper | eine Rundachse<br>eine Zustellachse senkrecht zur Drehachse<br>eine Längsachse parallel zur Drehachse |

| TRAANG | Aktivierung der Transformation Schräge Achse |
|---|---|
| Bearbeitung mit schräger Zustellachse | eine Rundachse<br>eine Zustellachse mit parametrierbaren Winkel<br>eine Längsachse parallel zur Drehachse |

## Kartesisches PTP-Fahren

Die Bewegung der Maschine erfolgt in Maschinenkoordinaten und wird programmiert mit:

| TRAORI | Aktivierung der Transformation |
|---|---|
| PTP Punkt-zu-Punkt-Fahren | Position im kartesischen Koordinatensystem (MKS) anfahren |
| CP | Bahnbewegung der kartesischen Achsen im (BKS) |
| STAT | Stellung der Gelenke ist abhängig von der Transformation |
| TU | Um welchen Winkel die Achsen auf den kürzesten Weg verfahren |

### PTP-Fahren bei generischer 5/6-AchsTransformation

Die Bewegung der Maschine erfolgt in Maschinenkoordinaten und die Werkzeugorientierung kann sowohl mit Rundachspositionen als auch mit von der Kinematik unabhängigen Vektoren Euler bzw. RPY-Winkel oder den Richtungsvektoren programmiert werden.

Dabei sind Rundachsinterpolation, Vektoreninterpolation mit Großkreisinterpolation oder Interpolation des Orientierungsvektors auf einer Kegelmantelfläche möglich.

### Beispiel Drei-, bis Fünf-Achs-Transformation bei einen Kardanischen Fräskopf

Die Werkzeugmaschine hat mindestens 5 Achsen, davon

- Drei translatorische Achsen für geradlinige Bewegungen, die den Arbeitspunkt an jede beliebige Position im Arbeitsaum bewegen.
- Zwei rotatorische Schwenkachsen, die unter einem projektierbaren Winkel (meist 45 Grad) angeordnet sind, ermöglichen dem Werkzeug Orientierungen im Raum einzunehmen, die sich bei 45 Grad Anordnung auf eine Halbkugel beschränken.

### 3.9.1.2 Übersicht der Orientierungstransformation TRAORI

## Mögliche Programmierungsarten im Zusammenhang mit TRAORI

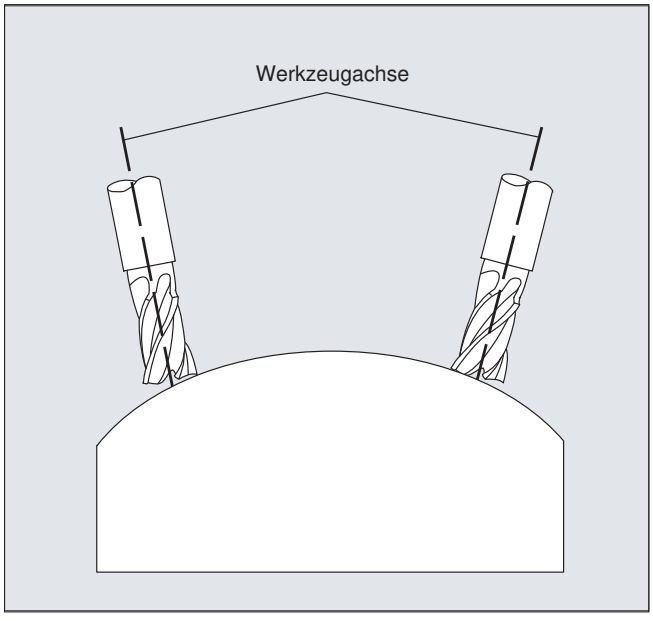
| Maschinentyp | Programmierung bei aktiver Transformation TRAORI |
|---|---|
| Maschinentypen 1, 2 oder 3 Zweiachs-Schwenkkopf oder Zweiachs-Drehtisch oder eine Kombination von jeweils einachsigem Schwenkkopf und Drehtisch. | Achsfolge der Orientierungsachsen und die Orientierungsrichtung des Werkzeugs ist und entweder<br>**maschinenbezogen** projektierbar über Maschinendaten<br>abhängig von der Maschinenkinemtik oder<br>**werkstückbezogen** mit programmierbarer Orientierung<br>unabhängig von der Maschinenkinemtik<br>Die Drehrichtungen der Orientierungsachsen im Bezugssystem wird programmiert mit:<br>- ORIMKS Bezugssystem = Maschinenkoordinatensystem<br>- ORIWKS Bezugssystem = Werkstückkoordinatensystem<br>Die Grundeinstellung ist ORIWKS.<br>Programmierung der Orientierungsachsen mit:<br>A, B, C der Maschinenachspositionen direkt<br>A2, B2, C2 Winkelprogrammierung virtueller Achsen mit<br>- ORIEULER über Euler-Winkel (Standard)<br>- ORIRPY über RPY-Winkel<br>- ORIVIRT1 über virtuelle Orientierungsachsen 1. Definition<br>- ORIVIRT2 über virtuelle Orientierungsachsen 2. Definition<br>mit Unterscheidung der Interpolationsart:<br>**lineare Interpolation**<br>- ORIAXES von Orientierungsachsen oder Maschinenachsen<br>**Großkreisinterpolation** (Interpolation des Orientierungsvektors)<br>- ORIVECT von Orientierungsachsen<br>Programmierung der Orientierungsachsen durch Angabe<br>A3, B3, C3 der Vektorkomponenten (Richtung-/Flächennormale)<br>Programmierung der resultierenden Werkzeugorientierung<br>A4, B4, C4 des Flächennormalvektors am Satzanfang<br>A5, B5, C5 des Flächennormalvektors am Satzende<br>LEAD Voreilwinkel für die Werkzeugorientierung<br>TILT Seitwärtswinkel für die Werkzeugorientierung |
| | **Interpolation des Orientierungsvektors** auf einer Kegelmantelfläche<br>Orientierungsänderungen auf einer **beliebig im Raum** befindlichen Kegelmantelfläche durch Interpolation:<br>- ORIPLANE in der Ebene (Großreisinterpolation)<br>- ORICONCW auf einer Kegelmantelfläche im Uhrzeigersinn<br>- ORICONCCW auf einer Kegelmantelfläche gegen Uhrzeigersinn<br>A6, B6, C6 Richtungsvektors (Drehachse des Kegels)<br>-OICONIO Interpolation auf einer Kegelmantelfläche mit:<br>A7, B7, C7 Zwischenvektoren (Start- und Endorientierung) oder<br>- ORICONTO auf einer Kegelmantelfläche tangentialer Übergang<br>Orientierungsänderungen bezogen **auf eine Bahn** mit<br>- ORICURVE Vorgabe der Bewegung zweier Kontaktpunkte über<br>PO[XH]=(xe, x2, x3, x4, x5) Orientierungspolynome bis 5.Grades<br>PO[YH]=(ye, y2, y3, y4, y5) Orientierungspolynome bis 5.Grades<br>PO[ZH]=(ze, z2, z3, z4, z5) Orientierungspolynome bis 5.Grades<br>- ORIPATHS Glättung des Orientierungsverlaufs mit<br>A8, B8, C8 Umorientierungsphase des Werkzeugs entspricht: Richtung und Weglänge des Werkzeugs bei der Abhebebewegung |

| Maschinentyp | Programmierung bei aktiver Transformation TRAORI |
|---|---|
| Maschinentypen 1 und 3 | Programmierung **der Drehungen** der Werkzeugorientierung mit<br>LEAD Voreilwinkel Winkel relativ zum Flächennormalenvektor<br>PO[PHI] Programmierung eines Polynoms bis 5.Grades<br>TILT Seitwärtswinkel Drehung um Bahntangente (Z-Richtung)<br>PO[PSI] Programmierung eines Polynoms bis 5.Grades |
| Weitere Maschinentypen mit zusätzlicher Drehung des Werkzeugs um sich selbst erfordern eine 3. Rundachse | THETA Drehwinkel (Drehung um die Werkzeugrichtung in Z)<br>THETA= Wert der am Satzende erreicht wird<br>THETA=AC(...) Satzweise auf Maßangabe absolut umschalten<br>THETA=IC(...) Satzweise auf Kettenmaßangabe umschalten<br>THETA=$\Theta_e$ Programmierter Winkel G90/G91 interpolieren<br>PO[THT]=(..) Programmierung eines Polynoms bis 5.Grades |
| Orientierungstransformation, wie z.B. generische 6-Achs-Transformation. Drehungen des Orientierungsvektors. | Programmierung des Drehvektors<br>- ORIROTA Drehung absolut<br>- ORIROTR relativer Drehvektor<br>- ORIROTT tangentialer Drehvektor |
| Bahnrelative Orientierung für Orientierungsänderungen relativ zur Bahn oder Drehung des Drehvektors tangential zur Bahn | Orientierungsänderungen **relativ zur Bahn** mit<br>- ORIPATH Werkzeugorientierung bezogen auf die Bahn<br>- ORIPATHS zusätzlich bei einen Knick im Orientierungsverlauf<br>Programmierung des Drehvektors<br>- ORIROTC tangentialer Drehvektor, Drehung zur Bahntangente |

## 3.9.2 Drei-, Vier- und Fünf-Achs-Transformation (TRAORI)

### 3.9.2.1 Allgemeine Zusammenhänge Kardanischer Werkzeugkopf

Um optimale Schnittbedingungen beim Bearbeiten räumlich gekrümmter Flächen zu erzielen, muss der Anstellwinkel des Werkzeugs veränderbar sein.

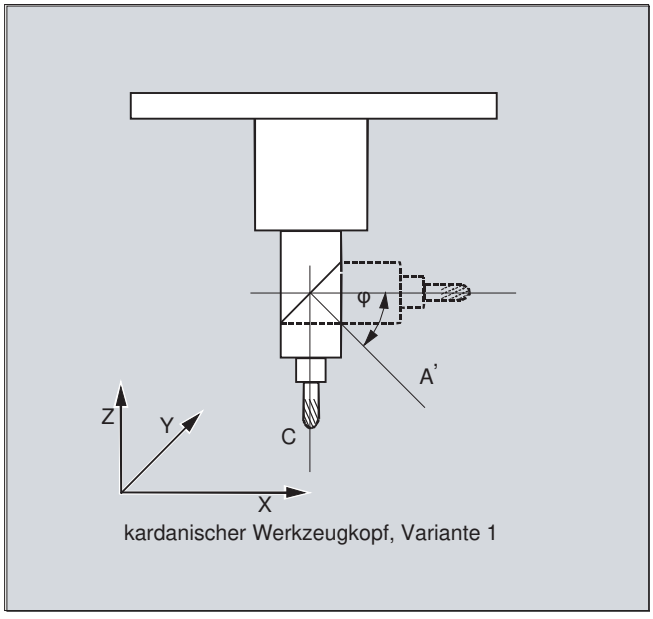



Mit welcher Maschinenkonstruktion dies erreicht wird, ist in den Achsdaten hinterlegt.

## 5-Achs-Transformation

### Kardanischer Werkzeugkopf

Hier legen drei Linearachsen (X, Y, Z) und zwei Orientierungsachsen (C, A) den Anstellwinkel und Arbeitspunkt des Werkzeugs fest. Eine der beiden Orientierungsachsen ist als Schrägachse angelegt, hier im Beispiel A' - in vielen Fällen als 45°-Anordnung.

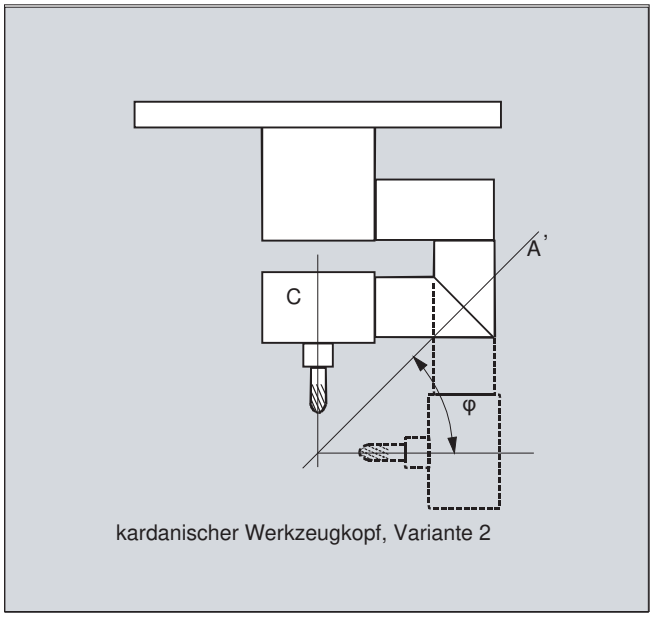


kardanischer Werkzeugkopf, Variante 1

In den hier gezeigten Beispielen sehen Sie die Anordnungen am Beispiel mit dem Kardanischen Werkzeugkopf der Maschinenkinematik CA!

### Maschinenhersteller

Die Achsfolge der Orientierungsachsen und die Orientierungsrichtung des Werkzeugs kann abhängig von der Maschinenkinematik über Maschinendaten eingestellt werden.



kardanischer Werkzeugkopf, Variante 2

In diesem Beispiel liegt A' unter dem Winkel φ zur X-Achse

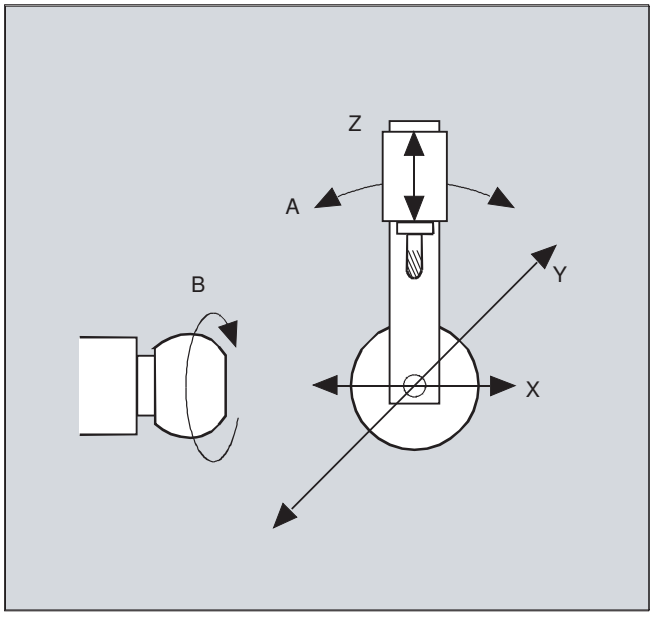
Allgemein gelten folgende mögliche Zusammenhänge:

| | |
|---|---|
| A' liegt unter dem Winkel φ zur | X-Achse |
| B' liegt unter dem Winkel φ zur | Y-Achse |
| C' liegt unter dem Winkel φ zur | Z-Achse |

Der Winkel φ kann im Bereich 0° bis +89° über Maschinendaten projektiert werden.

### Mit schwenkbarer Linearachse

Hierbei handelt es sich um eine Anordnung mit bewegtem Werkstück und bewegtem Werkzeug. Die Kinematik setzt sich aus drei Linearachsen (X, Y, Z) und zwei rechtwinklig angeordneten Drehachsen zusammen. Die erste Rundachse wird z. B. über einen Kreuzschlitten von zwei Linearachsen bewegt, das Werkzeug steht parallel zur dritten Linearachse. Die zweite Drehachse dreht das Werkstück. Die dritte Linearachse (Schwenkachse) liegt in der Ebene des Kreuzschlittens.




Die Achsfolge der rotatorischen Achsen und die Orientierungsrichtung des Werkzeugs kann abhängig von der Maschinenkinematik über Maschinendaten eingestellt werden.

Es gelten folgende mögliche Zusammenhänge:

| Achsen: | Achsfolgen: |
|---|---|
| 1. Rundachse | A A B B C C |
| 2. Rundachse | B C A C A B |
| Geschwenkte Linearachse | Z Y Z X Y X |

**Weitere Informationen** zu konfigurierbaren Achsfolgen für die Orientierungsrichtung des Werkzeugs siehe Funktionshandbuch Transformationen.

### 3.9.2.2 Drei, Vier, und Fünf- Achs-Transformation (TRAORI)

Der Anwender kann zwei bzw. drei translatorische Achsen und eine rotatorische Achse projektieren. Die Transformationen gehen davon aus, dass die rotatorische Achse orthogonal auf der Orientierungsebene steht.

Die Orientierung des Werkzeugs ist nur in der Ebene möglich, die senkrecht zur rotatorischen Achse ist. Die Transformation unterstützt die Maschinentypen mit beweglichem Werkzeug und beweglichem Werkstück.

Die Projektierung und Programmierung der Drei- und Vier-Achs-Transformationen sind analog zu den Fünf-Achs-Transformationen.

**Weitere Informationen:** Funktionshandbuch Transformationen

## Syntax

```
TRAORI(<n>)
TRAORI(<n>,<X>,<Y>,<Z>,<A>,<B>)
```

```
TRAFOOF
```

## Bedeutung

| | | |
|---|---|---|
| TRAORI: | Aktiviert die erste vereinbarte Orientierungstransformation | |
| TRAORI(<n>): | Aktiviert die mit n vereinbarte Orientierungstransformation | |
| <n>: | Nummer der Transformation | |
| | Wert: | 1 oder 2 |
| | Beispiel:<br>TRAORI(1) aktiviert Orientierungstransformation 1 | |
| <X>,<Y>,<Z>: | Komponente des Orientierungsvektors, in die das Werkzeug zeigt. | |
| <A>,<B>: | Programmierbarer Offset für die Rundachsen | |
| TRAFOOF: | Transformation ausschalten | |

### Werkzeugorientierung

Abhängig von der gewählten Orientierungsrichtung des Werkzeugs muss im NC-Programm die aktive Arbeitsebene (G17, G18, G19) so eingestellt werden, dass die Werkzeuglängenkorrektur in Richtung der Werkzeugorientierung wirkt.

### Hinweis

Nach dem Einschalten der Transformation beziehen sich Positionsangaben (X, Y, Z) immer auf die Spitze des Werkzeugs. Änderung der Positionen der an der Transformation beteiligten Rundachsen führen zu Ausgleichsbewegungen der übrigen Maschinenachsen, wodurch die Position der Werkzeugspitze unverändert bleibt.

Die Orientierungstransformation ist immer von der Werkzeugspitze zur Werkzeugaufnahme gerichtet.

### Offset für Orientierungsachsen

Bei Aktivierung der Orientierungstransformation kann ein zusätzlicher Offset für Orientierungsachsen direkt programmiert werden.

Es dürfen Parameter weggelassen werden, wenn bei der Programmierung die richtige Reihenfolge eingehalten wird.

Beispiel:

```
TRAORI(, , , ,A,B) ; wenn nur ein einziger Offset eingegeben werden soll
```

Alternativ zur direkten Programmierung kann der zusätzliche Offset für Orientierungsachsen auch aus der momentan aktiven Nullpunktverschiebung automatisch übernommen werden. Die Übernahme wird über Maschinendaten projektiert.

## Beispiele

```
TRAORI(1,0,0,1)          ; Die Grundorientierung des Werkzeugs zeigt in Z-Richtung
TRAORI(1,0,1,0)          ; Die Grundorientierung des Werkzeugs zeigt in Z-Richtung
```

```
TRAORI(1,0,1,1)          ; Die Grundorientierung des Werkzeugs zeigt in Y/Z-Richtung
                         (entspricht Stellung -45°)
```

### 3.9.2.3 Varianten der Orientierungsprogrammierung und Grundstellung (ORIRESET)

#### Orientierungsprogrammierung der Werkzeugorientierung bei TRAORI

In Verbindung mit einer programmierbaren Orientierungstransformation TRAORI können zusätzlich zu den Linearachsen X, Y, Z auch über die Rundachsbezeichner A.., B..., C... Achspositionen oder virtuelle Achsen mit Winkeln oder Vektorkomponenten programmiert werden. Für Orientierungs- und Maschinenachsen sind verschiedene Interpolationsarten möglich. Unabhängig davon, welche Orientierungspolynome PO[Winkel] und Achspolynome PO[Achse] gerade aktiv sind, können mehrere unterschiedliche Polynomarten wie z.B. G1, G2, G3, CIP oder POLY programmiert sein.

Die Änderung der Orientierung des Werkzeuges kann auch über Orientierungsvektoren programmiert werden. Hierbei kann die Endorientierung jedes Satzes entweder durch direkte Programmierung des Vektors oder durch Programmierung der Rundachspositionen erfolgen.

**Varianten der Orientierungsprogrammierung bei Drei- bis Fünf-Achs-Transformationen**

Folgende Varianten der Orientierungsprogrammierung schließen sich gegenseitig aus.

| | |
|---|---|
| `A, B, C` | Direkte Angabe der Rundachspositionen |
| `A2, B2, C2` | Winkelprogrammierung virtueller Achsen über Eulerwinkel oder RPY-Winkel |
| `A3, B3, C3` | Angabe der Vektorkomponenten |
| `LEAD, TILT` | Angabe der Voreil- und Seitwärtswinkel bezogen auf die Bahn und Oberfläche |
| `A4, B4, C4`<br>`A5, B5, C5` | Flächennormalenvektor am Satzanfang und am Satzende |
| `A6, B6, C6`<br>`A7, B7, C7` | Interpolation des Orientierungsvektors auf einer Kegelmantelfläche |
| `A8, B8, C8` | Umorientierung des Werkzeugs, Richtung und Weglänge der Abhebebewegung |

#### Grundstellung der Werkzeugorientierung anfahren (ORIRESET)

Durch `ORIRESET(...)` werden die Orientierungsachsen der jeweiligen Maschinenkinematik linear und synchron von ihrer aktuellen Position auf die programmierte Grundstellungsposition verfahren. Wenn für eine Achse keine Grundstellungsposition programmiert ist, dann wird die Position aus dem dazugehörigen Maschinendatum $MC_TRAFO5_ROT_AX_OFFSET_1/2 verwendet.

Aktive Frames der Rundachsen werden nicht berücksichtigt.

**Beispiele für Maschinenkinematik CA (Kanalachsnamen C, A)**

| Befehle | Beschreibung |
|---|---|
| `ORIRESET(90, 45)` | `Achse C: 90°`<br>`Achse A: 45°` |
| `ORIRESET(, 30)` | `Achse C: $MC_TRAFO5_ROT_AX_OFFSET_1/2[0]`<br>`Achse A: 30°` |
| `ORIRESET( )` | `Achse C: $MC_TRAFO5_ROT_AX_OFFSET_1/2[0]`<br>`Achse A: $MC_TRAFO5_ROT_AX_OFFSET_1/2[1]` |

**Beispiele für Maschinenkinematik CAC (Kanalachsnamen C, A, B)**

| Befehle | Beschreibung |
|---|---|
| `ORIRESET(90, 45, 90)` | `Achse C: 90°`<br>`Achse A: 45°`<br>`Achse B: 90°` |
| `ORIRESET( )` | `Achse C: $MC_TRAFO5_ROT_AX_OFFSET_1/2[0]`<br>`Achse A: $MC_TRAFO5_ROT_AX_OFFSET_1/2[1]`<br>`Achse B: $MC_TRAFO5_ROT_AX_OFFSET_1/2[2]` |

**Hinweis**

Das Anfahren der Grundstellung der Werkzeugorientierung mit `ORIRESET(...)` darf nur bei aktiver Orientierungstransformation `TRAORI(...)` erfolgen.

## Programmierung der Drehungen LEAD, TILT und THETA

### Voreilwinkel LEAD und Seitwärtswinkel TILT

Die Drehungen der Werkzeugorientierung werden bei der Drei- bis Fünf-Achs-Transformation mit den Voreilwinkel LEAD und den Seitwärtswinkel TILT programmiert.

### Drehwinkel THETA

Bei einer Transformation **mit dritter Rundachse** kann sowohl für die Orientierung mit Vektorkomponenten als auch mit Programmierung der Winkel LEAD, TILT die Drehung des Werkzeugs um sich selbst mit dem Drehwinkel THETA programmiert werden.

## 3.9.2.4 Programmierung der Werkzeugorientierung (A..., B..., C..., LEAD, TILT)

Für die Programmierung der Orientierung des Werkzeugs gibt es folgende Möglichkeiten:

1. Direkte Programmierung der Bewegung der Rundachsen. Die Orientierungsänderung erfolgt immer im Basis- bzw. Maschinen-Koordinatensystem. Die Orientierungsachsen werden als Synchronachsen verfahren.
2. Programmierung in Euler- oder RPY-Winkeln gemäß Winkeldefinition über `A2, B2, C2`.
3. Programmierung des Richtungsvektors über `A3, B3, C3`. Der Richtungsvektor zeigt von der Werkzeugspitze in Richtung Werkzeugaufnahme.
4. Programmierung des Flächennormalenvektors am Satzanfang mit `A4, B4, C4` und am Statzende mit `A5, B5, C5` (Stirnfräsen).

5. Programmierung über Voreilwinkel `LEAD` und Seitwärtswinkel `TILT`
6. Programmierung der Drehachse des Kegels als normierter Vektor über `A6, B6, C6` oder der Zwischenorientierung auf der Kegelmantelfläche über `A7, B7, C7,`
   siehe Kapitel "Orientierungsprogrammierung entlang einer Kegelmantelfläche (ORIPLANE, ORICONxx)".
7. Programmierung der Umorientierung, Richtung und Weglänge des Werkzeugs während der Abhebebewegung über `A8, B8, C8,`
   siehe Kapitel "Glättung des Orientierungsverlaufs (ORIPATHS A8=, B8=, C8=)"

**Hinweis**

In allen Fällen ist die Orientierungsprogrammierung nur zulässig, wenn eine Orientierungstransformation eingeschaltet ist.

Vorteil: Diese Programme sind auf jede Maschinenkinematik portierbar.

## Definition der Werkzeugorientierung über G-Befehl

**Hinweis**

**Maschinenhersteller**

Über Maschinendatum kann zwischen Euler- oder RPY-Winkeln umgeschaltet werden. Bei entsprechenden Maschinendaten Einstellungen ist eine Umschaltung sowohl abhängig als auch unabhängig vom aktiven G-Befehl der Gruppe 50 möglich. Folgende Einstellmöglichkeiten stehen zur Auswahl:

1. Wenn beide Maschinendaten für die Definition der Orientierungsachsen und Definition der Orientierungswinkel über G-Befehl auf Null gesetzt sind:
   Die mit `A2, B2, C2` programmierten Winkel werden **abhängig vom Maschinendatum** Winkeldefinition der Orientierungsprogrammierung entweder als Euler- oder RPY-Winkeln interpretiert.
2. Wenn das Maschinendatum für die Definition der Orientierungsachsen über G-Befehl auf Eins gesetzt ist, erfolgt Umschaltung
   **abhängig** vom aktiven G-Befehl der Gruppe 50:
   Die mit `A2, B2, C2` programmierten Winkel werden gemäß einem der aktiven G-Befehle `ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2, ORIAXPOS` und `ORIPY2` interpretiert. Die mit den Orientierungsachsen programmierten Werte werden entsprechend dem aktiven G-Befehl der Gruppe 50 auch als Orientierungswinkel interpretiert.
3. Wenn das Maschinendatum für die Definition der Orientierungswinkel über G-Befehl auf Eins und das Maschinendatum für die Definition der Orientierungsachsen über G-Befehl auf Null gesetzt ist erfolgt Umschaltung
   **unabhängig** vom aktiven G-Befehl der Gruppe 50:
   Die mit `A2, B2, C2` programmierten Winkel werden gemäß einem der aktiven G-Befehle `ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2 ORIAXPOS` und `ORIPY2` interpretiert. Die mit den Orientierungsachsen programmierten Werte werden unabhängig vom aktiven G-Befehl der Gruppe 50 immer als Rundachspositionen interpretiert.

## Syntax

### Rundachspositionen

```
G1 X<Wert> Y<Wert> Z<Wert> A<Wert> B<Wert> C<Wert>
```

### Eulerwinkel

```
G1 X<Wert> Y<Wert> Z<Wert> A2=<Wert> B2=<Wert> C2=<Wert>
```

### Richtungsvektor

```
G1 X<Wert> Y<Wert> Z<Wert> A3=<Wert> B3=<Wert> C3=<Wert>
```

### Flächennormalenvektor am Satzanfang

```
G1 X<Wert> Y<Wert> Z<Wert> A4=<Wert> B4=<Wert> C4=<Wert>
```

### Flächennormalenvektor am Satzende

```
G1 X<Wert> Y<Wert> Z<Wert> A5=<Wert> B5=<Wert> C5=<Wert>
```

### Voreilwinkel

```
LEAD=<Wert>
```

### Seitwärtswinkel

```
TILT=<Wert>
```

## Bedeutung

| | |
|---|---|
| `G1:` | Linearinterpolation |
| `X, Y, Z:` | Linearachspositionen |
| `A, B, C:` | Rundachspositionen |
| `A2=, B2=, C2=:` | Winkelprogrammierung (Euler- oder RPY-Winkel) |
| `A3=, B3=, C3=:` | Richtungsvektor in X, Y und Z-Koordinaten des WKS |
| `A4=, B4=, C4=:` | Flächennormalenvektor am Satzanfang in X, Y und Z-Koordinaten des WKS |
| `A5=, B5=, C5=:` | Flächennormalenvektors am Satzende in X, Y und Z-Koordinaten des WKS |
| `LEAD= :` | Voreilwinkel [1)] |
| `TILT= :` | Seitwärtswinkel [1)] |
| 1) Die Interpretation der Winkelangaben ist abhängig von der Einstellung in MD21094 $MC_ORI-PATH_MODE | |

## Weitere Informationen

In der Regel werden 5-Achs-Programme von CAD/CAM-Systemen erzeugt und nicht an der Steuerung eingegeben. Deshalb wenden sich die folgenden Erklärungen hauptsächlich an Programmierer von Postprozessoren.

Folgende Befehle zur Orientierungsprogrammierung stehen zur Verfügung:

| Befehl | Bedeutung |
|---|---|
| ORIEULER: | Euler-Winkel mit Drehreihenfolge ZX'Z" |
| ORIRPY: | RPY-Winkel mit Drehreihenfolge XY'Z" |
| ORIRPY2: | RPY-Winkel mit Drehreihenfolge ZY'X" |
| ORIVIRT1: | Virtuelle Orientierungsachsen mit frei definierbarer Drehreihenfolge über:<br>MD21120 $MC_ORIAX_TURN_TAB_1 |
| ORIVIRT2: | Virtuelle Orientierungsachsen mit frei definierbarer Drehreihenfolge über:<br>MD21130 $MC_ORIAX_TURN_TAB_2 |
| ORIAXPOS: | Virtuelle Orientierungsachsen mit Rundachspositionen |

**Hinweis**

Über Maschinendaten können vom Maschinenhersteller verschiedene Varianten definiert werden. Bitte beachten Sie die Angaben des Maschinenherstellers.

### Programmierung in Eulerwinkeln ORIEULER, Drehreihenfolge Z X' Z"

Die bei der Orientierungsprogrammierung ORIEULER mit A2, B2, C2 programmierten Werte werden als Eulerwinkel (in Grad) interpretiert.

Der neue Orientierungsvektor ergibt sich, durch folgende drei Drehungen des ursprünglichen Orientierungsvektors

1. mit der Rundachse A2 um die Koordinatenachse Z
2. mit der Rundachse B2 um die neue Koordinatenachse X'
3. mit der Rundachse C2 um die neue Koordinatenachse Z"

In diesem Fall ist der Wert von C2 (Drehung um neue Z-Achse) bedeutungslos und muss nicht programmiert werden.

### Programmierung in RPY-Winkeln ORIRPY, Drehreihenfolge X Y' Z"

Die bei der Orientierungsprogrammierung ORIRPY mit A2, B2, C2 programmierten Werte werden als RPY-Winkel (in Grad) mit der Drehreihenfolge X Y' Z" interpretiert.

**Hinweis**

Im Gegensatz zur Programmierung mit ORIEULER haben bei ORIRPY alle drei Werte Einfluss auf den Orientierungsvektor.

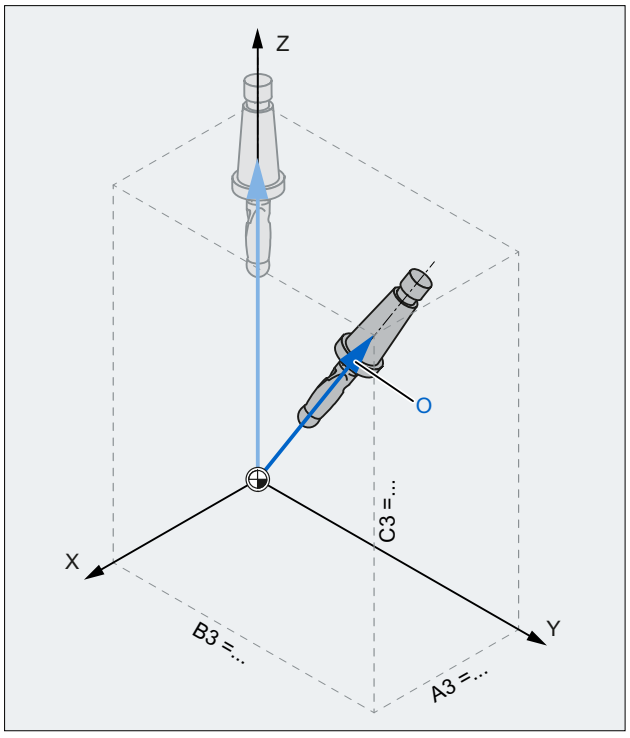
Der neue Orientierungsvektor ergibt sich, durch folgende drei Drehungen des ursprünglichen Orientierungsvektors

1. mit der Rundachse A2 um die Koordinatenachse X
2. mit der Rundachse B2 um die neue Koordinatenachse Y'
3. mit der Rundachse C2 um die neue Koordinatenachse Z"

### Programmierung des Richtungsvektors

Die Komponenten des Richtungsvektors werden mit `A3, B3, C3` programmiert. Der Vektor zeigt in Richtung Werkzeugaufnahme; die Länge des Vektors ist dabei ohne Bedeutung.

Nicht programmierte Vektorkomponenten werden gleich Null gesetzt.



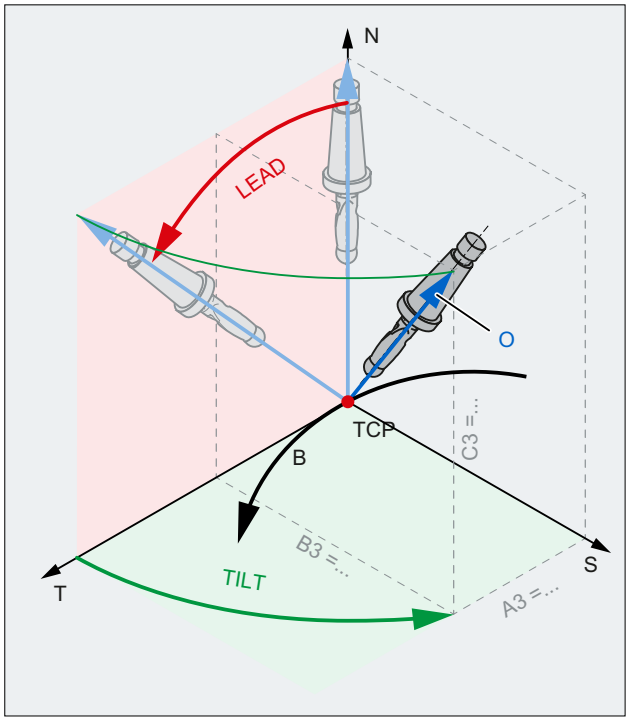

| | |
|---|---|
| X, Y, Z | Koordinatenachsen des WKS |
| A3, B3, C3 | Komponenten des Richtungsvektors |
| O | Orientierungsvektor |

Bild 3-3 Programmierung des Richtungsvektors

### Programmierung der Werkzeugorientierung mit LEAD und TILT

Die resultierende Werkzeugorientierung wird ermittelt aus:

- Bahntangente
- Flächennormalenvektor
  Am Satzanfang `A4, B4, C4` und am Satzende `A5, B5, C5`
- Voreilwinkel `LEAD`
  Winkel in der von Bahntangente und Flächennormalenvektor aufgespannten Ebene
- Seitwärtswinkel `TILT` am Satzende
  Winkel in der Ebene senkrecht zur Bahntangente relativ zum Flächennormalenvektor




| | |
|---|---|
| T | Bahntangente |
| S | Senkrechte zur Bahntangente |
| N | Flächennormale |
| B | Bahn |
| TCP | Tool Center Point |
| O | Orientierungsvektor |

Bild 3-4 Programmierung von LEAD TILT

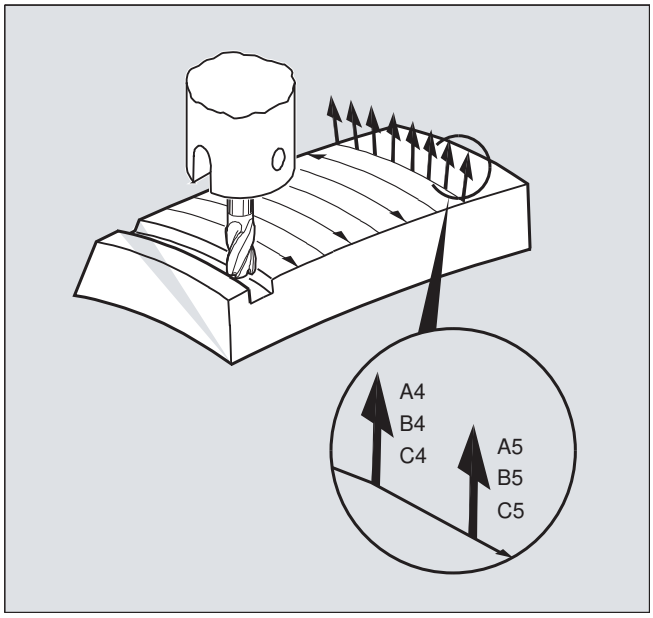
**Hinweis**

**Verhalten bei Innenecken bei 3D-WZK**

Wenn der Satz an einer Innenecke verkürzt wird, wird die programmierte Werkzeugorientierung trotzdem am Satzende eingenommen.

### 3.9.2.5 Stirnfräsen (A4, B4, C4, A5, B5, C5)

Stirnfräsen dient zur Bearbeitung beliebig gekrümmter Oberflächen.




Für diese Art des 3D-Fräsens benötigen Sie die zeilenweise Beschreibung der 3D-Bahnen auf der Werkstückoberfläche.

Die Berechnungen werden unter Berücksichtigung der Werkzeugform und Werkzeugabmessungen üblicherweise im CAM durchgeführt. Die fertig berechneten NC-Sätze werden dann über Postprozessoren in die Steuerung eingelesen.

## Programmierung der Bahnkrümmung

#### Beschreibung der Flächen

Die Beschreibung der Bahnkrümmung erfolgt über Flächennormalenvektoren mit folgenden Komponenten:

`A4, B4, C4` Startvektor am Satzanfang

`A5, B5, C5` Endvektor am Satzende

Steht in einem Satz nur der Startvektor, bleibt der Flächennormalenvektor über den ganzen Satz konstant. Steht in einem Satz nur der Endvektor, so wird vom Endwert des vorherigen Satzes über Großkreisinterpolation zum programmierten Endwert interpoliert.

Sind Start- und Endvektor programmiert, so wird zwischen beiden Richtungen ebenfalls über Großkreisinterpolation interpoliert. Hierdurch lassen sich kontinuierlich glatte Bahnwege erzeugen.

In der Grundstellung zeigen Flächennormalenvektoren unabhängig von der aktiven Ebene G17 bis G19 in Z-Richtung.

Die Länge eines Vektors ist ohne Bedeutung.

Nicht programmierte Vektorkomponenten werden zu Null gesetzt.

Bei aktivem ORIWKS (siehe "Bezug der Orientierungsachsen (ORIWKS, ORIMKS) (Seite 684)") beziehen sich die Flächennormalenvektoren auf den aktiven Frame und werden bei Frame-Drehung mitgedreht.

**Maschinenhersteller**

Der Flächennormalenvektor muss innerhalb eines über Maschinendatum einstellbaren Grenzwertes senkrecht zur Bahntangente stehen, ansonsten wird Alarm ausgegeben.

### 3.9.2.6 Bezug der Orientierungsachsen (ORIWKS, ORIMKS)

Bei Orientierungsprogrammierung im Werkstückkoordinatensystem über

- Euler- bzw. RPY-Winkel oder
- Orientierungsvektor

kann der Verlauf der Drehbewegung über `ORIMKS`/`ORIWKS` eingestellt werden.

**Hinweis**

**Maschinenhersteller**

Die Interpolationsart für die Orientierung wird festgelegt mit dem Maschinendatum:

MD21104 $MC_ORI_IPO_WITH_G_CODE

= FALSE: Bezug sind die G-Befehle ORIWKS und ORIMKS

= TRUE: Bezug sind die G-Befehle der 51. Gruppe (ORIAXES, ORIVECT, ORIPLANE, ...)

## Syntax

```
ORIMKS=...
ORIWKS=...
```

## Bedeutung

| | |
|---|---|
| `ORIMKS`: | Drehung im Maschinenkoordinatensystem |
| `ORIWKS`: | Drehung im Werkstückkoordinatensystem |

**Hinweis**

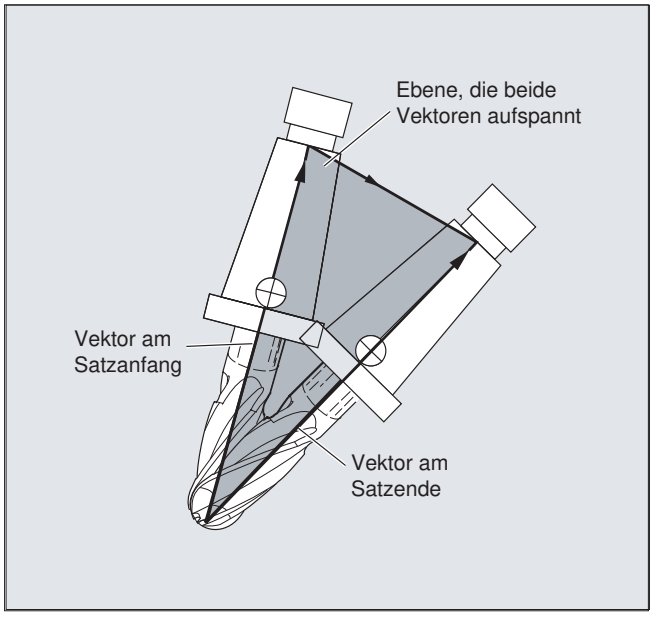
ORIWKS ist Grundeinstellung. Ist bei einem Fünf-Achs-Programm nicht von vornherein klar, auf welcher Maschine es ablaufen soll, so ist grundsätzlich ORIWKS zu wählen. Welche Bewegungen die Maschine tatsächlich ausführt, hängt von der Maschinenkinematik ab.

Mit `ORIMKS` können tatsächliche Maschinenbewegungen programmiert werden, z. B. um Kollisionen mit Vorrichtungen o. ä. zu vermeiden.

## Weitere Informationen

Bei ORIMKS ist die ausgeführte Werkzeugbewegung von der Maschinenkinematik **abhängig**. Bei Orientierungsänderung mit raumfester Werkzeugspitze wird zwischen den Rundachspositionen linear interpoliert.

Bei ORIWKS ist die Werkzeugbewegung von der Maschinenkinematik **unabhängig**. Bei Orientierungsänderung mit raumfester Werkzeugspitze bewegt sich das Werkzeug in der vom Anfangs- und Endvektor aufgespannten Ebene.




### Singuläre Stellungen

**Hinweis**

**ORIWKS**

Orientierungsbewegungen im Bereich der singulären Stellung der Fünf-Achs-Maschine erfordern große Bewegungen der Maschinenachsen. (Beispielsweise sind bei einem Drehschwenkkopf mit C als Drehachse und A als Schwenkachse alle Stellungen mit A=0 singulär.)

**Maschinenhersteller**

Um die Maschinenachsen nicht zu überlasten, senkt die Geschwindigkeitsführung die Bahngeschwindigkeit in der Nähe der singulären Stellen stark ab.

Mit den Maschinendaten

$MC_TRAFO5_NON_POLE_LIMIT

$MC_TRAFO5_POLE_LIMIT

kann die Transformation so parametriert werden, dass Orientierungsbewegungen in der Nähe des Pols durch den Pol gelegt werden und eine zügige Bearbeitung möglich ist.

Singuläre Stellen werden nur mit dem MD $MC_TRAFO5_POLE_LIMIT behandelt.

**Weitere Informationen:** Funktionshandbuch Transformationen

### 3.9.2.7 Programmierung der Orientierungsachsen (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2)

Die Funktion "Orientierungsachsen" beschreibt die Orientierung des Werkzeugs im Raum und wird durch Programmierung der Offsets für die Rundachsen erreicht. Ein weiterer dritter Freiheitsgrad kann durch die zusätzliche Drehung des Werkzeugs um sich selbst erzielt werden. Diese Werkzeugorientierung erfolgt beliebig im Raum über eine dritte Rundachse und erfordert die Sechs-Achs-Transformation. Die Eigendrehung des Werkzeugs um sich selbst wird abhängig von der Interpolationsart der Drehvektoren mit dem Drehwinkel THETA festgelegt (siehe "Drehungen der Werkzeugorientierung (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Seite 695)".

Orientierungsachsen werden über die Achsbezeichner A2, B2, C2 programmiert.

## Syntax

```
N... ORIAXES/ORIVECT            ; Lineare oder Großkreisinterpolation
N... G1 X Y Z A B C

N... ORIPLANE                   ; Orientierungsinterpolation der Ebene

N... ORIEULER/ORIRPY/ORIRPY2    ; Orientierungswinkel Euler-/RPY-Winkel
N... G1 X Y Z A2= B2= C2=       ; Winkelprogrammierung virtueller Achsen

N... ORIVIRT1/ORIVIRT2          ; virtuelle Orientierungsachsen Def. 1/2
N... G1 X Y Z A3= B3= C3=       ; Richtungsvektorprogrammierung
```

**Hinweis**

Für Orientierungsänderungen entlang einer im Raum befindlichen Kegelmantelfläche können weitere Rundachsoffsets der Orientierungsachsen programmiert werden (siehe "Orientierungsprogrammierung entlang einer Kegelmantelfläche (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) (Seite 688)".

## Bedeutung

| | |
|---|---|
| ORIAXES: | Lineare Interpolation der Maschinen- oder Orientierungsachsen |
| ORIVECT: | Großkreisinterpolation (identisch mit ORIPLANE) |
| ORIMKS:<br>ORIWKS: | Drehung im Maschinenkoordinatensystem<br>Drehung im Werkstückkoordinatensystem<br>Beschreibung siehe "Bezug der Orientierungsachsen (ORIWKS, ORIMKS) (Seite 684)". |
| A= B= C=: | Programmierung der Maschinenachsposition |
| ORIEULER: | Orientierungsprogrammierung über Euler-Winkel |

| | |
|---|---|
| ORIRPY: | Orientierungsprogrammierung über RPY-Winkel<br>Die Drehreihenfolge ist XYZ, wobei gilt:<br>• A2 ist der Drehwinkel um X<br>• B2 ist der Drehwinkel um Y<br>• C2 ist der Drehwinkel um Z |
| ORIRPY2: | Orientierungsprogrammierung über RPY-Winkel<br>Die Drehreihenfolge ist ZYX, wobei gilt:<br>• A2 ist der Drehwinkel um Z<br>• B2 ist der Drehwinkel um Y<br>• C2 ist der Drehwinkel um X |
| A2= B2= C2=: | Winkelprogrammierung virtueller Achsen |
| ORIVIRT1/ORIVIRT2: | Orientierungsprogrammierung über virtuelle Orientierungsachsen<br>Definition 1:<br>Festlegung nach MD21120 $MC_ORIAX_TURN_TAB_1<br>Definition 2:<br>Festlegung nach MD21130 $MC_ORIAX_TURN_TAB_2 |
| A3= B3= C3=: | Richtungsvektorprogrammierung der Richtungsachse |

## Weitere Informationen

### Maschinenhersteller

Mit MD21102 $MC_ORI_DEF_WITH_G_CODE wird festgelegt, wie die programmierten Winkel A2, B2, C2 definiert werden:

Definition erfolgt nach MD21100 $MC_ORIENTATION_IS_EULER (Standard) oder Definition erfolgt nach G-Gruppe 50 (ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2).

Mit MD21104 $MC_ORI_IPO_WITH_G_CODE wird festgelegt, welche Interpolationsart wirksam ist: ORIWKS/ORIMKS oder ORIAXES/ORIVECT.

### Betriebsart JOG

Die Orientierungswinkel werden in dieser Betriebsart immer linear interpoliert. Beim kontinuierlichen und inkrementellen Verfahren über Verfahrtasten kann nur eine Orientierungsachse verfahren werden. Über die Handräder können die Orientierungsachsen gleichzeitig verfahren werden.

Für das manuelle Verfahren von Orientierungsachsen wirkt der kanalspezifische Vorschub-Override-Schalter bzw. der Eilgang-Override-Schalter bei Eilgangüberlagerung.

Mit folgenden Maschinendaten ist eine separate Geschwindigkeitsvorgabe möglich:

MD21160 $MC_JOG_VELO_RAPID_GEO

MD21165 $MC_JOG_VELO_GEO

MD21150 $MC_JOG_VELO_RAPID_ORI

MD21155 $MC_JOG_VELO_ORI

**Hinweis**

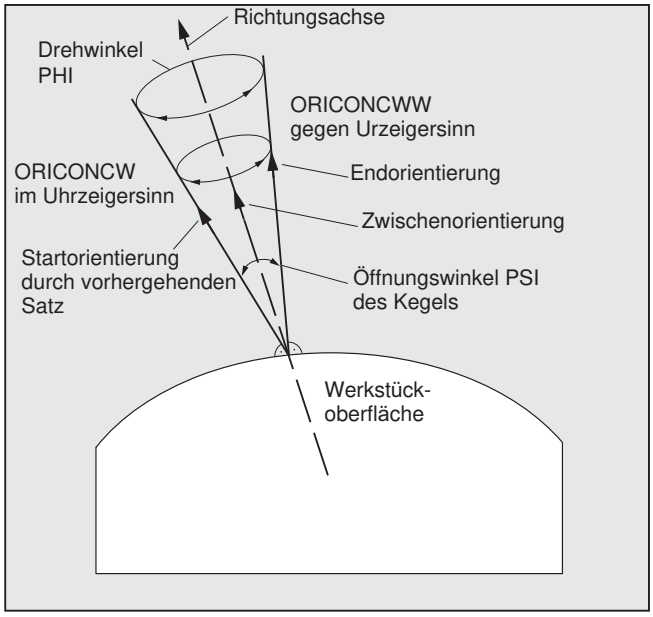
**SINUMERIK 840D sl mit "Transformationspaket Handling"**

Mit der Funktion "Kartesisches manuelles Verfahren" kann im JOG-Betrieb die Translation von Geometrieachsen in den Bezugssystemen MKS, WKS und TCS getrennt voneinander eingestellt werden.

**Weitere Informationen:** Funktionshandbuch Transformationen

### 3.9.2.8 Orientierungsprogrammierung entlang einer Kegelmantelfläche (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO)

Mit der erweiterten Orientierung ist es möglich, Orientierungsänderungen entlang sich einer im Raum befindlichen Kegelmantelfläche auszuführen. Die Interpolation des Orientierungsvektors auf einer Kegelmantelfläche erfolgt mit den modalen Befehlen ORICONxx. Für die Interpolation in einer Ebene kann die Endorientierung mit ORIPLANE programmiert werden. Generell wird die Startorientierung durch die vorhergehenden Sätze festgelegt.



## Programmierung

Die Endorientierung wird entweder durch Angabe der Winkelprogrammierung in Euler- oder RPY-Winkel mit A2, B2, C2 oder durch Programmierung der Rundachspositionen mit A, B, C festgelegt. Für die Orientierungsachsen entlang der Kegelmantelfläche sind weitere Programmierangaben erforderlich:

- Drehachse des Kegels als Vektor mit `A6, B6, C6`
- Öffnungswinkel `PSI` mit den Bezeichner `NUT`
- Zwischenorientierung im Kegelmantel mit `A7, B7, C7`

**Hinweis**

**Programmierung des Richtungsvektor A6, B6, C6 für die Drehachse des Kegels**

Die Programmierung einer Endorientierung ist nicht unbedingt erforderlich. Ist keine Endorientierung angegeben, dann wird ein voller Kegelmantel mit 360 Grad interpoliert.

**Programmierung des Öffnungswinkel des Kegels mit NUT=winkel**

Die Angabe einer Endorientierung ist zwingend erforderlich.

Ein vollständiger Kegelmantel mit 360 Grad kann auf diese Weise nicht interpoliert werden.

**Programmierung der Zwischenorientierung A7, B7, C7 im Kegelmantel**

Die Angabe einer Endorientierung ist zwingend erforderlich. Die Orientierungsänderung und Drehrichtung wird eindeutig durch die drei Vektoren Start-, End- und Zwischenorientierung festgelegt. Alle drei Vektoren müssen hierbei voneinander unterschiedlich sein. Ist die programmierte Zwischenorientierung parallel zur Start- oder Endorientierung, dann wird eine lineare Großkreisinterpolation der Orientierung in der Ebene, die von Start- und Endvektor aufgespannt wird, durchgeführt.

**Erweiterte Orientierungsinterpolation auf einer Kegelmantelfläche**

```
N... ORICONCW oder ORICONCCW
N... A6= B6= C6= A3= B3= C3=
oder
N... ORICONTO
N... G1 X Y Z A6= B6= C6=
oder
N... ORICONIO
N... G1 X Y Z A7= B7= C7=
N... PO[PHI]=(a2, a3, a4, a5)
N... PO[PSI]=(b2, b3, b4, b5)
```

**Interpolation** auf einen Kegelmantel mit
Richtungsvektor im/gegen Uhrzeigersinn des Kegels und Endorientierung oder
tangentialem Übergang und Angabe der Endorientierung
oder
Angabe der Endorientierung und einer Zwischenorientierung im Kegelmantel mit
Polynome für Drehwinkel und
Polynome für Öffnungswinkel

## Parameter

| | |
|---|---|
| `ORIPLANE`: | Interpolation in der Ebene (Großkreisinterpolation) |
| `ORICONCW`: | Interpolation auf einer Kegelmantelfläche im Uhrzeigersinn |
| `ORICONCCW`: | Interpolation auf einer Kegelmantelfläche gegen Uhrzeigersinn |
| `ORICONTO`: | Interpolation auf einer Kegelmantelfläche tangentialer Übergang |

| | |
|---|---|
| A6= B6= C6=: | Programmierung der Drehachse des Kegels (normierter Vektor) |
| NUT=winkel: | Öffnungswinkel des Kegels in Grad |
| NUT=+179: | Verfahrwinkel kleiner oder gleich 180 Grad |
| NUT=-181: | Verfahrwinkel größer oder gleich 180 Grad |
| ORICONIO: | Interpolation auf einer Kegelmantelfläche |
| A7= B7= C7=: | Zwischenorientierung (Programmierung als normierter Vektor) |
| PHI: | Drehwinkel der Orientierung um die Richtungsachse des Kegels |
| PSI: | Öffnungswinkel des Kegels |
| mögliche Polynome<br>PO[PHI]=(a2, a3, a4, a5)<br>PO[PSI]=(b2, b3, b4, b5) | Außer den jeweiligen Winkeln sind auch Polynome maximal 5. Grades programmierbar |

## Beispiel: Unterschiedliche Orientierungsänderungen

| **Programmcode** | **Kommentar** |
|---|---|
| … | |
| N10 G1 X0 Y0 F5000 | |
| N20 TRAORI(1) | ; Orientierungstransformation ein. |
| N30 ORIVECT | ; Werkzeug-Orientierung als Vektor interpolieren. |
| … | ; Werkzeugorientierung in der Ebene. |
| N40 ORIPLANE | ; Großkreisinterpolation auswählen. |
| N50 A3=0 B3=0 C3=1 | |
| N60 A3=0 B3=1 C3=1 | ; Orientierung in der Y/Z-Ebene um 45 Grad gedreht, am Satzende wird die Orientierung (0,1/√2,1/√2)erreicht. |
| … | |
| N70 ORICONCW | ; Orientierungsprogrammierung auf Kegelmantel: |
| N80 A6=0 B6=0 C6=1 A3=0 B3=0 C3=1 | ; Der Orientierungsvektor wird auf einem Kegelmantel mit der Richtung (0,0,1) bis zur Orientierung (1/√2,0,1/√2) im Uhrzeigersinn interpoliert, der Drehwinkel beträgt hierbei 270 Grad. |
| N90 A6=0 B6=0 C6=1 | ; Die Werkzeugorientierung durchläuft eine volle Umdrehung auf demselben Kegelmantel. |

## Weitere Informationen

Sollen Orientierungsänderungen auf einer beliebig im Raum liegenden Kegelmantelfläche beschrieben werden, dann muss der Vektor um den die Werkzeugorientierung gedreht werden soll, bekannt sein. Außerdem müssen die Start- und Endorientierung vorgegeben werden. Die Startorientierung ergibt sich aus den vorhergehenden Satz und die Endorientierung muss entweder programmiert oder durch andere Bedingungen festgelegt werden.

#### Programmierung in der Ebene ORIPLANE entspricht ORIVECT

Die Programmierung der Großkreisinterpolation zusammen mit Winkelpolynomen entspricht der Linear- und Polynominterpolation von Konturen. Die Werkzeugorientierung wird in einer Ebene interpoliert, die von der Start- und Endorientierung aufgespannt wird. Werden zusätzlich Polynome programmiert, dann kann der Orientierungsvektor auch aus der Ebene gekippt werden.

#### Programmierung von Kreisen in einer Ebene G2/G3, CIP und CT

Die erweiterte Orientierung entspricht der Interpolation von Kreisen in einer Ebene. Zu den entsprechenden Programmiermöglichkeiten von Kreisen mit Mittelpunktsangabe oder Radiusangabe wie G2/G3, Kreis über Zwischenpunkt CIP und Tangentialkreise CT siehe Kapitel "Kreisinterpolation (Seite 191)".

#### Orientierungsprogrammierung

#### Interpolation des Orientierungsvektors auf einer Kegelmantelfläche ORICONxx

Für die Interpolation von Orientierungen auf einer Kegelmantelfläche können vier verschiedene Interpolationsarten aus der G-Gruppe 51 ausgewählt werden:

1. Interpolation auf einen Kegelmantel im Uhrzeigersinn ORICONCW mit Angabe der Endorientierung und der Kegelrichtung oder des Öffnungswinkels. Der Richtungsvektor wird mit den Bezeichnern A6, B6, C6 und der Öffnungswinkel des Kegels wird mit dem Bezeichner NUT= Wertebereich im Intervall 0 bis 180 Grad programmiert.
2. Interpolation auf einen Kegelmantel gegen Uhrzeigersinn ORICONCWW mit Angabe der Endorientierung und der Kegelrichtung oder des Öffnungswinkels. Der Richtungsvektor wird mit den Bezeichnern A6, B6, C6 und der Öffnungswinkel des Kegels wird mit dem Bezeichner NUT= Wertebereich im Intervall 0 bis 180 Grad programmiert.
3. Interpolation auf einen Kegelmantel ORICONIO mit Angabe der Endorientierung und einer Zwischenorientierung, die mit den Bezeichnern A7, B7, C7 programmiert wird.
4. Interpolation auf einen Kegelmantel ORICONTO mit tangentialem Übergang und Angabe der Endorientierung. Der Richtungsvektor wird mit den Bezeichnern A6, B6, C6 programmiert.

### 3.9.2.9 Orientierungsvorgabe zweier Kontaktpunke (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=)

#### Programmierung der Orientierungsänderung durch die zweite Raumkurve ORICURVE

Eine weitere Möglichkeit der Programmierung von Orientierungsänderungen besteht darin, außer der Werkzeugspitze entlang einer Raumkurve auch die Bewegung eines zweiten Kontaktpunktes des Werkzeugs mit ORICURVE zu programmieren. Damit können Orientierungsänderungen des Werkzeugs, wie bei der Programmierung des Werkzeugvektors selber, eindeutig festgelegt werden.

#### Maschinenhersteller

Beachten Sie bitte die Hinweise des Maschinenherstellers zu über Maschinendatum einstellbare Achsbezeichner für die Programmierung der 2. Orientierungsbahn des Werkzeugs.

## Programmierung

Bei dieser Interpolationsart können für die beiden Raumkurven Punkte mit G1 bzw. Polynome mit POLY programmiert werden. Kreise und Evolventen sind nicht zulässig. Zusätzlich kann eine Spline-Interpolation mit BSPLINE und die Funktion "Zusammenfassung kurzer Spline-Sätze" aktiviert werden.

**Weitere Informationen** zur Funktion "Zusammenfassung kurzer Spline-Sätze" siehe Funktionshandbuch Basisfunktionen.

Die anderen Splinearten ASPLINE und CSPLINE sowie die Aktivierung eines Kompressors mit COMP... sind nicht zulässig.

Die Bewegung der zwei Kontaktpunkte des Werkzeugs kann bei der Programmierung der Orientierungspolynome für Koordinaten bis maximal 5.Grades vorgegeben werden.

### Erweiterte Orientierungsinterpolation mit zusätzlicher Raumkurve und Polynome für Koordinaten

```
N... ORICURVE
N... PO[XH]=(xe, x2, x3, x4, x5)
N... PO[YH]=(ye, y2, y3, y4, y5)
N... PO[ZH]=(ze, z2, z3, z4, z5)
```

; Angabe der Bewegung des zweiten Kontaktpunkts des Werkzeugs und zusätzliche Polynome der jeweiligen Koordinaten

## Parameter

| | |
|---|---|
| `ORICURVE` | Interpolation der Orientierung mit Vorgabe der Bewegung zweier Kontaktpunkte des Werkzeuges. |
| `XH YH ZH` | Bezeichner der Koordinaten des zweiten Kontaktpunktes des Werkzeugs der zusätzlichen Kontur als Raumkurve |
| `PO[XH]=(xe, x2, x3, x4, x5)`<br>`PO[YH]=(ye, y2, y3, y4, y5)`<br>`PO[ZH]=(ze, z2, z3, z4, z5)` | Mögliche Polynome<br>Außer den jeweiligen Endpunkten sind die Raumkurven zusätzlich mit Polynomen programmierbar. |
| `xe, ye, ze` | Endpunkte der Raumkurve |
| `xi, yi, zi` | Koeffizienten der Polynome maximal 5. Grades |

**Hinweis**

**Bezeichner XH YH ZH für die Programmierung einer 2. Orientierungsbahn**

Die Bezeichner müssen so gewählt werden, dass kein Konflikt mit anderen Bezeichnern der Linearachsen

X Y Z Achsen

und Rundachsen wie

A2 B2 C2 Eulerwinkel bzw. RPY-Winkel

A3 B3 C3 Richtungsvektoren

A4 B4 C4 bzw. A5 B5 C5 Flächennormalenvektoren

A6 B6 C6 Drehvektoren bzw. A7 B7 C7 Zwischenpunktkoordinaten

oder anderen Interpolationsparameter entsteht.

## 3.9.3 Orientierungspolynome (PO[Winkel], PO[Koordinate])

Unabhängig davon, welche Polynominterpolation der G-Gruppe 1 gerade aktiv ist, können zwei verschiedene Typen von Orientierungspolynomen bis maximal 5. Grades bei einer Drei- bis Fünf-Achs-Transformation programmiert werden.

1. Polynome für **Winkel:** Voreilwinkel LEAD, Seitwärtswinkel TILT
   in Bezug auf die Ebene, die von Start- und Endorientierung aufgespannt wird.
2. Polynome für **Koordinaten:** XH, YH, ZH der zweiten Raumkurve für die Werkzeugorientierung eines Bezugspunktes auf dem Werkzeug.

Bei einer Sechs-Achs-Transformation kann zur Werkzeugorientierung zusätzlich die Drehung des Drehvektors THT mit Polynomen bis maximal 5. Grades für Drehungen des Werkzeugs selbst programmiert werden.

### Syntax

Orientierungspolynome vom Typ 1 für **Winkel**

```
N… PO[PHI]=(a2, a3, a4, a5)
N… PO[PSI]=(b2, b3, b4, b5)
```

Drei- bis Fünf-Achs-Transformation

Orientierungspolynome vom Typ 2 für **Koordinaten**

```
N… PO[XH]=(xe, x2, x3, x4, x5)
N… PO[YH]=(ye, y2, y3, y4, y5)
N… PO[ZH]=(ze, z2, z3, z4, z5)
```

Bezeichner für die Koordinaten der zweiten Orientierungsbahn für die Werkzeugorientierung

Zusätzlich kann in beiden Fällen ein Polynom für die **Drehung** bei Sechs-Achs-Transformationen mit

| | |
|---|---|
| `N… PO[THT]=(c2, c3, c4, c5)` | Bahnrelative Interpolation der Drehung |
| oder | |
| `N… PO[THT]=(d2, d3, d4, d5)` | absoluter, relative und tangentiale Interpolation zur Orientierungsänderung |

des Orientierungsvektors programmiert werden. Dies ist dann möglich, wenn die Transformation einen Drehvektor mit einem durch den Drehwinkel THETA programmierbaren und interpolierbaren Offset unterstützt.

## Bedeutung

| | |
|---|---|
| `PO[PHI]` | Winkel in der Ebene zwischen Start- und Endorientierung |
| `PO[PSI]` | Winkel der die Auskippung der Orientierung aus der Ebene zwischen Start- und Endorientierung beschreibt |
| `PO[THT]` | Drehwinkel der durch Drehung des Drehvektors einer der mit THETA programmierten G-Befehle der Gruppe 54 |
| `PHI` | Voreilwinkel LEAD |
| `PSI` | Seitwärtswinkel TILT |
| `THETA` | Drehung um die Werkzeugrichtung in Z |
| `PO[XH]` | X-Koordinate des Bezugspunktes auf dem Werkzeug |
| `PO[YH]` | Y-Koordinate des Bezugspunktes auf dem Werkzeug |
| `PO[ZH]` | Z-Koordinate des Bezugspunktes auf dem Werkzeug |

## Weitere Informationen

Orientierungspolynomen können nicht programmiert werden

- wenn die Splineinterpolationen ASPLINE, BSPLINE, CSPLINE aktiv sind.
  Polynome vom Typ1 für Orientierungswinkel sind für jede Interpolationsart außer Spline d.h. bei Linearinterpolation mit Eilgang G00 bzw. mit Vorschub G01
  bei Polynominterpolation mit POLY und
  bei Kreis- bzw. Evolventeninterpolation mit G02, G03, CIP, CT, INVCW und INCCCW möglich.
  Polynome vom Typ2 für Orientierungskoordinaten sind dagegen nur möglich, wenn Linearinterpolation mit Eilgang G00 bzw. mit Vorschub G01 oder
  Polynominterpolation mit POLY aktiv ist.
- wenn die Orientierung mittels Achsinterpolation ORIAXES interpoliert wird. In diesem Fall können direkt Polynome mit PO[A] und PO[B] für die Orientierungsachsen A und B programmiert werden.

### Orientierungspolynome vom Typ 1 mit ORIVECT, ORIPLANE und ORICONxx

Bei Großkreisinterpolation und Kegelmantelinterpolation mit ORIVECT, ORIPLANE und ORICONxx sind nur Orientierungspolynome vom Typ 1 möglich.

#### Orientierungspolynome vom Typ 2 mit ORICURVE

Ist die Interpolation mit zusätzlicher Raumkurve ORICURVE aktiv, werden die kartesischen Komponenten des Orientierungsvektors interpoliert und es sind nur Orientierungspolynome vom Typ 2 möglich.

## 3.9.4 Drehungen der Werkzeugorientierung (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA)

Soll bei Maschinentypen mit beweglichem Werkzeug auch die Orientierung des Werkzeugs veränderbar sein, so wird jeder Satz mit einer Endorientierung programmiert. Abhängig von der Maschinenkinematik können entweder die Orientierungsrichtung der Orientierungsachsen oder die Drehrichtung des Orientierungsvektors THETA programmiert werden. Für diese Drehvektoren sind verschiedene Interpolationsarten programmierbar:

- ORIROTA: Drehwinkel zu einer absolut vorgegebenen Drehrichtung.
- ORIROTR: Drehwinkel relativ zur Ebene zwischen Start- und Endorientierung.
- ORIROTT: Drehwinkel relativ zur Änderung des Orientierungsvektors.
- ORIROTC: Tangentialer Drehwinkel zur Bahntangente.

### Syntax

Nur wenn die Interpolationsart ORIROTA aktiv ist, kann der Drehwinkel oder der Drehvektor auf die vier möglichen Arten wie folgt programmiert werden:

1. Direkt die Rundachspositionen `A, B, C`
2. Eulerwinkel (in Grad) über `A2, B2, C2`
3. RPY-Winkel (in Grad) über `A2, B2, C2`
4. Richtungsvektor über `A3, B3, C3` (Drehwinkel mittels `THETA=<Wert>`)

Falls ORIOTR oder ORIOTT aktiv sind, kann der Drehwinkel nur noch direkt mit `THETA` programmiert werden.

Eine Drehung kann auch allein in einem Satz programmiert werden, ohne dass eine Orientierungsänderung stattfindet. Dabei haben ORIROTR und ORIROTT keine Bedeutung. In diesem Fall wird der Drehwinkel immer in Bezug zur absoluten Richtung interpretiert (ORIROTA).

| | |
|---|---|
| `N... ORIROTA` | Interpolation des Drehvektors festlegen |
| `N... ORIROTR` | |
| `N... ORIROTT` | |
| `N... ORIROTC` | |
| `N... A3= B3= C3= THETA=<Wert>` | Drehung des Orientierungsvektors festlegen |
| `N... PO[THT]=(`$d_2$`, `$d_3$`, `$d_4$`, `$d_5$`)` | Drehwinkel mit Polynom 5. Grades interpolieren |

## Bedeutung

| | |
|---|---|
| ORIROTA: | Drehwinkel zu einer absolut vorgegebenen Drehrichtung |
| ORIROTR: | Drehwinkel relativ zur Ebene zwischen Start- und Endorientierung |
| ORIROTT: | Drehwinkel als tangentialer Drehvektor zur Orientierungsänderung |
| ORIROTC: | Drehwinkel als tangentialer Drehvektor zur Bahntangente |
| THETA: | Drehung des Orientierungsvektors |
| THETA=<Wert>: | Drehwinkel in Grad, der am Satzende erreicht wird |
| THETA=Θe: | Drehwinkel mit Endwinkel $\Theta_e$ des Drehvektors |
| THETA=AC(…): | Satzweise auf Maßangabe absolut umschalten |
| THETA=AC(…): | Satzweise auf Kettenmaßangabe umschalten |
| Θe: | Endwinkel des Drehvektors sowohl absolut mit G90 als auch relativ mit G91 (Kettenmaßangabe) ist aktiv |
| PO[THT]=(....): | Polynom für den Drehwinkel |

## Beispiel: Drehungen der Orientierungen

| **Programmcode** | **Kommentar** |
|---|---|
| N10 TRAORI | ; Orientierungstransformation aktivieren |
| N20 G1 X0 Y0 Z0 F5000 | ; Orientierung des Werkzeugs |
| N30 A3=0 B3=0 C3=1 THETA=0 | ; in Z-Richtung mit Drehwinkel 0 |
| N40 A3=1 B3=0 C3=0 THETA=90 | ; in X-Richtung und Drehung um 90 Grad |
| N50 A3=0 B3=1 C3=0 PO[THT]=(180,90) | ; Orientierung |
| N60 A3=0 B3=1 C3=0 THETA=IC(-90) | ; in Y-Richtung und Drehung auf 180 Grad |
| N70 ORIROTT | ; bleibt konstant und Drehung auf 90 Grad |
| N80 A3=1 B3=0 C3=0 THETA=30 | ; Drehwinkel relativ zur Orientierungsänderung |
| | ; Drehvektor im Winkel 30 Grad zur X-Y Ebene |

Bei der Interpolation von Satz N40 wird der Drehwinkel vom Startwert 0 Grad zum Endwert 90 Grad linear interpoliert. Im Satz N50 ändert sich der Drehwinkel von 90 Grad auf 180 Grad gemäß der Parabel θ(u) = +90u2. In N60 kann auch eine Drehung ausgeführt werden, ohne dass eine Orientierungsänderung stattfindet.

Bei N80 wird die Werkzeugorientierung von der Y–Richtung in X–Richtung gedreht. Dabei liegt die Orientierungsänderung in der X–Y Ebene und der Drehvektor bildet zu dieser Ebene einen Winkel von 30 Grad.

## Weitere Informationen

### ORIROTA

Der Drehwinkel THETA wird bezüglich einer absolut festgelegten Richtung im Raum interpoliert. Die Grunddrehrichtung erfolgt über Maschinendaten

### ORIROTR

Der Drehwinkel THETA wird relativ zur Ebene, die von der Start- und Endorientierung aufgespannt wird, interpretiert.

**ORIROTT**

Der Drehwinkel `THETA` wird relativ zur Orientierungsänderung interpretiert. Für `THETA=0` wird der Drehvektor tangential zur Orientierungsänderung interpoliert und unterscheidet sich nur dann zu `ORIROTR`, wenn für die Orientierung mindestens ein Polynom für den "Kippwinkel PSI" programmiert wurde. Damit ergibt sich eine Orientierungsänderung, die nicht in der Ebene abläuft. Durch einen zusätzlich programmierten Drehwinkel `THETA` kann dann z. B. der Drehvektor so interpoliert werden, dass er immer einen bestimmten Wert zur Orientierungsänderung bildet.

**ORIROTC**

Der Drehvektor wird relativ zur Bahntangente mit einem durch den Winkel `THETA` programmierbaren Offset interpoliert. Für den Offsetwinkel kann dabei auch ein Polynom `PO[THT]=(c2, c3, c4, c5)` maximal 5. Grades programmiert werden.

## 3.9.5 Bahnrelative Orientierungen

### 3.9.5.1 Orientierungsarten relativ zur Bahn

Mit dieser erweiterten Funktion wird die relative Orientierung nicht nur am Satzende, sondern über den gesamten Bahnverlauf erreicht. Es wird die im Vorgängersatz erreichte Orientierung mittels Großkreisinterpolation in die programmierte Endorientierung überführt. Grundsätzlich gibt es zwei Möglichkeiten die gewünschte Orientierung relativ zur Bahn zu programmieren:

1. Die Werkzeugorientierung als auch die Drehung des Werkzeugs wird mit ORIPATH, ORPATHTS relativ zur Bahn interpoliert.
2. Der Orientierungsvektor wird wie bisher üblich programmiert und interpoliert. Mit ORIROTC wird die Drehung des Orientierungsvektors relativ zur Bahntangente angestellt.

### Syntax

Die Interpolationsart der Orientierung und der Drehung des Werkzeugs wird programmiert mit:

| | |
|---|---|
| `N... ORIPATH` | Bahnrelative Orientierung |
| `N... ORIPATHS` | Bahnrelative Orientierung mit Glättung des Orientierungsverlaufs |
| `N... ORIROTC` | Bahnrelative Interpolation des Drehvektors |

Ein durch eine Ecke im Bahnverlauf hervorgerufener Knick der Orientierung kann mit ORIPATHS geglättet werden. Die Richtung und Weglänge der Abhebebewegung wird durch den Vektor mit den Komponenten A8 = X, B8 = Y und C8 = Z programmiert.

Mit ORIPATH/ORIPATHS können verschiedene Bezüge zur Bahntangente über die drei Winkel

- LEAD = Angabe Vorwärtswinkel bezogen auf die Bahn und Oberfläche
- TILT = Angabe von Seitwärtswinkel bezogen auf die Bahn und Oberfläche
- THETA = Drehwinkel

für den gesamten Bahnverlauf programmiert werden. Zum Drehwinkel THETA können mit PO[THT] = (...) zusätzlich Polynome maximal 5. Grades programmiert werden.

**Hinweis**

**Maschinenhersteller**

Bitte beachten Sie die Angaben des Maschinenherstellers. Über projektierbare Maschinen- und Settingdaten können zur Bahnrelativen Orientierungsart weitere Einstellungen vorgenommen werden.

**Weitere Informationen:** Funktionshandbuch Transformationen

## Bedeutung

Die Interpolation der Winkel LEAD und TILT ist über Maschinedatum unterschiedlich einstellbar:

- Der mit LEAD und TILT programmierte Bezug der Werkzeugorientierung wird über den ganzen Satz hinweg eingehalten.
- Vorwärtswinkel LEAD: Drehung um die Richtung senkrecht zur Tangente und Normalenvektor TILT: Drehung der Orientierung um den Normalenvektor.
- Vorwärtswinkel LEAD: Drehung um die Richtung senkrecht zur Tangente und Normalenvektor Seitwärtswinkel TILT: Drehung der Orientierung um die Richtung der Bahntangente.
- Drehwinkel THETA: Drehung des Werkzeugs um sich selbst mit einer zusätzlichen dritten Rundachse als Orientierungsachse bei Sechs-Achs-Transformation.

**Hinweis**

**Bahnrelative Orientierung zusammen mit OSC, OSS, OSSE, OSD, OST unzulässig**

Die bahnrelative Orientierungsinterpolation ORIPATH bzw. ORIPATHS und ORIOTC kann nicht zusammen mit der Glättung des Orientierungsverlaufs mit einem der G-Befehle aus der Gruppe 34 programmiert werden. Hierfür muss OSOF aktiv sein.

### 3.9.5.2 Bahnrelative Drehung der Werkzeugorientierung (ORIPATH, ORIPATHS, Drehwinkel)

Bei einer Sechs-Achs-Transformation kann zur Werkzeugorientierung beliebig im Raum auch das Werkzeug mit einer dritten Rundachse um sich selbst gedreht werden. Bei bahnrelativer Drehung der Werkzeugorientierung mit ORIPATH bzw. ORIPATHS kann die zusätzliche Drehung über den Drehwinkel THETA programmiert werden. Alternativ hierzu können die Winkel LEAD und TILT durch einen Vektor, der in der Ebene senkrecht zur Werkzeugrichtung liegt, programmiert werden.

**Maschinenhersteller**

Bitte beachten Sie die Angaben des Maschinenherstellers. Über Maschinendatum kann die Interpolation der Winkel LEAD und TILT unterschiedlich eingestellt werden.

## Syntax

### Drehung der Werkzeugorientierung und des Werkzeugs

Die Werkzeugorientierungsart relativ zur Bahn wird mit ORIPATH oder ORIPATHS aktiviert.

| | |
|---|---|
| `N... ORIPATH` | Orientierungsart bezogen auf die Bahn aktivieren |
| `N... ORIPATHS` | Orientierungsart bezogen auf die Bahn mit Glättung des Orientierungsverlaufs aktivieren |

Aktivierung der drei möglichen Winkel mit Drehwirkung:

| | |
|---|---|
| `N... LEAD=` | Winkel für die programmierten Orientierung relativ zum Flächennormalenvektor |
| `N... TILT=` | Winkel für die programmierte Orientierung in der Ebene senkrecht zur Bahntangente relativ zum Flächennormalenvektor |
| `N... THETA=` | Drehwinkel relativ zur Orientierungsänderung um die Werkzeugrichtung der dritten Rundachse |

Die Werte der Winkel am Satzende werden mit `LEAD=Wert`, `TILT=Wert` bzw. `THETA=Wert` programmiert. Zusätzlich zu den konstanten Winkeln können für alle drei Winkel Polynome maximal 5. Grades programmiert werden.

| | |
|---|---|
| `N... PO[PHI]=(a2, a3, a4, a5)` | Polynom für den Voreilwinkel LEAD |
| `N... PO[PSI]=(b2, b3, b4, b5)` | Polynom für den Seitwärtswinkel TILT |
| `N... PO[THT]=(d2, d3, d4, d5)` | Polynom für den Drehwinkel THETA |

Bei der Programmierung können die höheren Polynomkoeffizienten, die Null sind, weggelassen werden. Beispiel `PO[PHI]=a2` ergibt für den Voreilwinkel `LEAD` eine Parabel.

## Bedeutung

### Bahnrelative Werkzeugorientierung

| | |
|---|---|
| `ORIPATH`: | Werkzeugorientierung bezogen auf die Bahn |
| `ORIPATHS`: | Werkzeugorientierung bezogen auf die Bahn, Knick im Orientierungsverlauf wird geglättet |
| `LEAD`: | Winkel relativ zum Flächennormalenvektor, in der von Bahntangente und Flächennormalenvektor aufgespannten Ebene |
| `TILT`: | Drehung der Orientierung um die Z-Richtung bzw. Drehung um die Bahntangente |
| `THETA`: | Drehung um die Werkzeugrichtung nach Z |
| `PO[PHI]`: | Orientierungspolynom für den Voreilwinkel LEAD |
| `PO[PSI]`: | Orientierungspolynom für den Seitwärtswinkel TILT |
| `PO[THT]`: | Orientierungspolynom für den Drehwinkel THETA |

**Hinweis**

**Drehwinkel THETA**

Für die Drehung des Werkzeugs mit dritter Rundachse als Orientierungsachse um sich selbst, ist eine Sechs-Achs-Transformation erforderlich.

### 3.9.5.3 Bahnrelative Interpolation der Werkzeugdrehung (ORIROTC, THETA)

**Interpolation mit Drehvektoren**

Zur mit ORIROTC programmierten Drehung des Werkzeugs relativ zur Bahntangenten kann der Drehvektor auch mit einem durch den Drehwinkel THETA programmierbaren Offset interpoliert werden. Dabei kann für den Offsetwinkel mit PO[THT] ein Polynom bis maximal 5. Grades programmiert werden.

## Syntax

| | |
|---|---|
| `N... ORIROTC` | Drehung des Werkzeugs relativ zur Bahntangente anstellen |
| `N... A3= B3= C3= THETA=Wert` | Drehung des Orientierungsvektors festlegen |
| `N... A3= B3= C3= PO[THT]=(c2, c3, c4, c5)` | Offsetwinkel mit Polynom maximal 5. Grades interpolieren |

Eine Drehung kann auch allein in einem Satz programmiert werden, ohne dass eine Orientierungsänderung stattfindet.

## Bedeutung

**Bahnrelative Interpolation der Drehung des Werkzeugs bei Sechs-Achs-Transformation**

| | |
|---|---|
| `ORIROTC`: | tangentialer Drehvektor zur Bahntangente anstellen |
| `THETA=Wert`: | Drehwinkel in Grad, der am Satzende erreicht wird |
| `THETA=`θe: | Drehwinkel mit Endwinkel $\Theta_e$ des Drehvektors |
| `THETA=AC(…)`: | Satzweise auf Maßangabe absolut umschalten |
| `THETA=IC(…)`: | Satzweise auf Kettenmaßangabe umschalten |
| `PO[THT]=(c2, c3, c4, c5)`: | Offsetwinkel mit Polynom 5. Grades interpolieren |

**Hinweis**

**Interpolation des Drehvektors ORIROTC**

Soll gegen die Orientierungsrichtung des Werkzeugs auch die Drehung des Werkzeugs relativ zur Bahntangente angestellt werden, dann ist dies nur bei einer Sechs-Achs-Transformation möglich.

**Bei aktiven ORIROTC**

Der Drehvektor ORIROTA kann nicht programmiert werden. Im Falle einer Programmierung wird der ALARM 14128 "Absolutprogrammierung der Werkzeugdrehung bei aktivem ORIROTC" ausgegeben.

### Orientierungsrichtung des Werkzeugs bei Drei- bis Fünf-Achs-Transformation

Die Orientierungsrichtung des Werkzeugs kann wie bei der Drei- bis Fünf-Achs-Transformation gewohnt über Eulerwinkel bzw. RPY-Winkel oder des Richtungsvektoren programmiert werden. Auch sind Orientierungsänderungen des Werkzeugs im Raum durch Programmierung der Großkreisinterpolation ORIVECT, der linearen Interpolation der Orientierungsachsen ORIAXES, alle Interpolationen auf einer Kegelmantelfläche ORICONxx sowie der Interpolation zusätzlich zur Raumkurve mit zwei Kontaktpunkten des Werkzeugs ORICURVE möglich.

| | |
|---|---|
| `G....:` | Angabe der Bewegungsart der Rundachsen |
| `X, Y, Z:` | Angabe der Linearachsen |
| `ORIAXES:` | Lineare Interpolation der Maschinen- oder Orientierungsachsen |
| `ORIVECT:` | Großkreisinterpolation (identisch mit ORIPLANE) |
| `ORIMKS:`<br>`ORIWKS:` | Drehung im Maschinenkoordinatensystem<br>Drehung im Werkstückkoordinatensystem<br>Beschreibung siehe Kap. Drehungen der Werkzeugorientierung |
| `A= B= C=:` | Programmierung der Maschinenachsposition |
| `ORIEULER:` | Orientierungsprogrammierung über Euler-Winkel |
| `ORIRPY:` | Orientierungsprogrammierung über RPY-Winkel |
| `A2= B2= C2=:` | Winkelprogrammierung virtueller Achsen |
| `ORIVIRT1:`<br>`ORIVIRT2:` | Orientierungsprogrammierung über virtuelle Orientierungsachsen<br>(Definition 1), Festlegung nach MD $MC_ORIAX_TURN_TAB_1<br>(Definition 2), Festlegung nach MD $MC_ORIAX_TURN_TAB_2 |
| `A3= B3= C3=:` | Richtungsvektorprogrammierung der Richtungsachse |
| `ORIPLANE:` | Interpolation in der Ebene (Großkreisinterpolation) |
| `ORICONCW:` | Interpolation auf einer Kegelmantelfläche im Uhrzeigersinn |
| `ORICONCCW:` | Interpolation auf einer Kegelmantelfläche gegen Uhrzeigersinn |
| `ORICONTO:` | Interpolation auf einer Kegelmantelfläche tangentialer Übergang |
| `A6= B6= C6=:` | Programmierung der Drehachse des Kegels (normierter Vektor) |
| `NUT=winkel` | Öffnungswinkel des Kegels in Grad |
| `NUT=+179` | Verfahrwinkel kleiner oder gleich 180 Grad |
| `NUT=-181` | Verfahrwinkel größer oder gleich 180 Grad |
| `ORICONIO:` | Interpolation auf einer Kegelmantelfläche |

| | |
|---|---|
| `A7= B7= C7=`: | Zwischenorientierung (Programmierung als normierter Vektor) |
| `ORICURVE`<br>`XH YH ZH z.B. mit Polynome PO[XH]=(xe, x2, x3, x4, x5)` | Interpolation der Orientierung mit Vorgabe der Bewegung zweier Kontaktpunkte des Werkzeuges. Außer den jeweiligen Endpunkten sind zusätzliche Raumkurven Polynome programmierbar. |

**Hinweis**

Wird die Werkzeugorientierung mit aktiven ORIAXES über die Orientierungsachsen Interpoliert, dann wird die bahnrelative Anstellung des Drehwinkels nur am Satzende erfüllt.

### 3.9.5.4 Glättung des Orientierungsverlaufs (ORIPATHS A8=, B8=, C8=)

Bei beschleunigungsstetigen Orientierungsänderungen an der Kontur sind Unterbrechungen der Bahnbewegungen, die besonders an einer Ecke der Kontur auftreten können unerwünscht. Der sich hieraus ergebene Knick im Orientierungsverlauf kann durch Einfügen eines eigenen Zwischensatzes geglättet werden. Die Orientierungsänderung erfolgt dann beschleunigungsstetig, wenn während der Umorientierung auch ORIPATHS aktiv ist. In dieser Phase kann eine Abhebebewegung des Werkzeugs durchgeführt werden.

**Maschinenhersteller**

Beachten Sie bitte die Hinweise des Maschinenherstellers zu gegebenenfalls vordefinierten Maschinendaten und Settingdaten mit denen diese Funktion aktiviert wird.

Über Maschinendatum ist einstellbar, wie der Abhebevektor interpretiert wird:

1. Im Werkzeugkoordinatensystem wird die Z-Koordinate durch die Werkzeugrichtung definiert.
2. Im Werkstückkoordinatensystem wird die Z-Koordinate durch die aktive Ebene definiert.

**Weitere Informationen** zur Funktion "Bahnrelative Orientierung" siehe Funktionshandbuch Transformationen.

## Syntax

Für stetige Werkzeugorientierungen bezogen auf die gesamte Bahn sind an einer Ecke der Kontur weitere Programmierangaben erforderlich. Die Richtung und die Weglänge dieser Bewegung wird durch den Vektor mit den Komponenten A8 = X, B8 = Y, C8 = Z programmiert:

```
N... ORIPATHS A8=X B8=Y C8=Z
```

## Bedeutung

| | |
|---|---|
| `ORIPATHS`: | Werkzeugorientierung bezogen auf die Bahn, ein Knick im Orientierungsverlauf wird geglättet. |
| `A8= B8= C8=`: | Vektorkomponenten für Richtung und Weglänge |
| `X, Y, Z`: | Abhebebewegung in Werkzeugrichtung |

**Hinweis**

**Programmierung des Richtungsvektors A8, B8, C8**

Ist die Länge dieses Vektors gleich Null erfolgt keine Abhebebewegung.

**ORIPATHS**

Die bahnbezogene Werkzeugorientierung wird mit ORIPATHS aktiv. Anderenfalls wird die Orientierung mittels linearer Großkreisinterpolation von der Start- zur Endorientierung überführt.

## 3.9.6 Komprimierung der Orientierung (COMPON, COMPCURV, COMPCAD, COMPSURF)

NC-Programme, in denen eine Orientierungstransformation (TRAORI) aktiv ist und Werkzeugorientierungen (egal welcher Art) programmiert sind, können unter Einhaltung von vorgegeben Toleranzen komprimiert werden.

### Programmierung

**Werkzeugorientierung**

Falls eine Orientierungstransformation (TRAORI) aktiv ist, kann bei 5-Achs Maschinen die Werkzeugorientierung folgendermaßen (kinematikunabhängig) programmiert werden:

- Programmierung des Richtungs**vektors** über:
  `A3=<...> B3=<...> C3=<...>`
- Programmierung der Euler**winkel** bzw. RPY-**Winkel** über:
  `A2=<...> B2=<...> C2=<...>`

**Drehung des Werkzeugs**

Bei **6-Achs** Maschinen kann zusätzlich zur Werkzeugorientierung noch die Drehung des Werkzeugs programmiert werden.

Die Programmierung des Drehwinkels erfolgt mit:

`THETA=<...>`

Siehe " Drehungen der Werkzeugorientierung (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Seite 695) ".

**Hinweis**

NC-Sätze, in denen zusätzlich eine Drehung programmiert ist, sind nur dann komprimierbar, falls sich der Drehwinkel **linear** ändert. D. h., für den Drehwinkel darf kein Polynom mit PO[THT] = (...) programmiert sein.

### Allgemeine Form eines komprimierbaren NC-Satzes

Die allgemeine Form eines komprimierbaren NC-Satzes kann daher wie folgt aussehen:

```
N... X=<...> Y=<...> Z=<...> A3=<...> B3=<...> C3=<...> THETA=<...> F=<...>
```

bzw.

```
N... X=<...> Y=<...> Z=<...> A2=<...> B2=<...> C2=<...> THETA=<...> F=<...>
```

**Hinweis**

Die Positionswerte können direkt (z. B. X90) oder indirekt über Parameterzuweisungen (z. B. X=R1*(R2+R3)) angegeben werden.

### Programmierung der Werkzeugorientierung durch Rundachspositionen

Die Werkzeugorientierung kann auch durch Rundachspositionen angegeben sein, z. B. in der Form:

```
N... X=<...> Y=<...> Z=<...> A=<...> B=<...> C=<...> THETA=<...> F=<...>
```

In diesem Fall wird die Komprimierung auf zwei unterschiedliche Arten durchgeführt, abhängig davon, ob eine Großkreisinterpolation durchgeführt wird oder nicht. Wenn keine Großkreisinterpolation stattfindet, dann wird die komprimierte Orientierungsänderung durch axiale Polynome für die Rundachsen in üblicher Weise dargestellt.

### Konturgenauigkeit

Abhängig vom eingestellten Kompressionsmodus (MD20482 $MC_COMPRESSOR_MODE) werden für die Geometrieachsen und Orientierungsachsen bei der Komprimierung entweder die projektierten achsspezifischen Toleranzen (MD33100 $MA_COMPRESS_POS_TOL) oder die folgenden über Settingdaten einstellbaren kanalspezifischen Toleranzen wirksam:

SD42475 $SC_COMPRESS_CONTUR_TOL (Maximale Konturabweichung)

SD42476 $SC_COMPRESS_ORI_TOL (Maximale Winkelabweichung für die Werkzeugorientierung)

SD42477 $SC_COMPRESS_ORI_ROT_TOL (Maximale Winkelabweichung für den Drehwinkel des Werkzeugs) (nur bei 6-Achs Maschinen verfügbar)

### Aktivierung / Deaktivierung

Kompressor-Funktionen werden eingeschaltet durch die modalen G-Befehle COMPON, COMPCURV, COMPCAD bzw. COMPSURF.

Beendet wird die Kompressor-Funktion mit COMPOF.

Siehe " NC-Satz-Kompression ein-/ausschalten (COMPON, COMPCURV, COMPCAD, COMPSURF, COMPOF) (Seite 613) ".

**Hinweis**

Die Orientierungsbewegung wird nur komprimiert bei aktiver Großkreisinterpolation (d. h., die Änderung der Werkzeugorientierung erfolgt in der Ebene, die von Start- und Endorientierung aufgespannt wird).

Eine Großkreisinterpolation wird unter den folgenden Bedingungen durchgeführt:

- MD21104 $MC_ORI_IPO_WITH_G_CODE = 0,
  ORIWKS ist aktiv und
  Orientierung ist mittels Vektoren programmiert (mit A3, B3, C3 bzw. A2, B2, C2).
- MD21104 $MC_ORI_IPO_WITH_G_CODE = 1 und
  ORIVECT bzw. ORIPLANE ist aktiv.
  Die Werkzeugorientierung kann entweder als Richtungsvektor oder mit Rundachspositionen programmiert sein. Ist einer der G-Befehle ORICONxx oder ORICURVE aktiv oder sind Polynome für die Orientierungswinkel (PO[PHI] und PO[PSI]) programmiert, wird keine Großkreisinterpolation durchgeführt.

## Beispiel

Im nachfolgenden Programmbeispiel wird ein Kreis, der durch einen Polygonzug angenähert ist, komprimiert. Die Werkzeugorientierung bewegt sich dabei synchron dazu auf einem Kegelmantel. Obwohl die aufeinander folgenden programmierten Orientierungsänderungen unstetig verlaufen, generiert die Kompressor-Funktion einen glatten Verlauf der Orientierung.

```
Programmierung                          Kommentar
DEF INT ANZAHL=60
DEF REAL RADIUS=20
DEF INT COUNTER
DEF REAL WINKEL
N10 G1 X0 Y0 F5000 G64

$SC_COMPRESS_CONTUR_TOL=0.05            ; Maximale Abweichung der Kontur = 0.05 mm
$SC_COMPRESS_ORI_TOL=5                  ; Maximale Abweichung der Orientierung
                                        = 5 Grad


TRAORI
COMPCURV                                ; Es wird ein Kreis gefahren, der aus Polygo-
                                        nen gebildet wird. Die Orientierung bewegt
                                        sich dabei auf einem Kegel um die Z-Achse
                                        mit einem Öffnungswinkel von 45 Grad.
N100 X0 Y0 A3=0 B3=-1 C3=1
N110 FOR COUNTER=0 TO ANZAHL
N120 WINKEL=360*COUNTER/ANZAHL
N130 X=RADIUS*cos(WINKEL) Y=RADIUS*sin(WINKEL)
A3=sin(WINKEL) B3=-cos(WINKEL) C3=1
N140 ENDFOR
```

## 3.9.7 Komprimierung der Orientierung (COMPCAD, COMPSURF)

NC-Programme, in denen eine Orientierungstransformation (TRAORI) aktiv ist und Werkzeugorientierungen (egal welcher Art) programmiert sind, können unter Einhaltung von vorgegeben Toleranzen komprimiert werden.

### Programmierung

#### Werkzeugorientierung

Falls eine Orientierungstransformation (TRAORI) aktiv ist, kann bei 5-Achs Maschinen die Werkzeugorientierung folgendermaßen (kinematikunabhängig) programmiert werden:

- Programmierung des Richtungs**vektors** über:
  `A3=<...> B3=<...> C3=<...>`
- Programmierung der Euler**winkel** bzw. RPY-**Winkel** über:
  `A2=<...> B2=<...> C2=<...>`

#### Drehung des Werkzeugs

Bei **6-Achs** Maschinen kann zusätzlich zur Werkzeugorientierung noch die Drehung des Werkzeugs programmiert werden.

Die Programmierung des Drehwinkels erfolgt mit:

`THETA=<...>`

Siehe " Drehungen der Werkzeugorientierung (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Seite 695) ".

**Hinweis**

NC-Sätze, in denen zusätzlich eine Drehung programmiert ist, sind nur dann komprimierbar, falls sich der Drehwinkel **linear** ändert. D. h., für den Drehwinkel darf kein Polynom mit PO[THT] = (...) programmiert sein.

#### Allgemeine Form eines komprimierbaren NC-Satzes

Die allgemeine Form eines komprimierbaren NC-Satzes kann daher wie folgt aussehen:

```
N... X=<...> Y=<...> Z=<...> A3=<...> B3=<...> C3=<...> THETA=<...> F=<...>
```

bzw.

```
N... X=<...> Y=<...> Z=<...> A2=<...> B2=<...> C2=<...> THETA=<...> F=<...>
```

**Hinweis**

Die Positionswerte können direkt (z. B. X90) oder indirekt über Parameterzuweisungen (z. B. X=R1*(R2+R3)) angegeben werden.

#### Programmierung der Werkzeugorientierung durch Rundachspositionen

Die Werkzeugorientierung kann auch durch Rundachspositionen angegeben sein, z. B. in der Form:

```
N... X=<...> Y=<...> Z=<...> A=<...> B=<...> C=<...> THETA=<...> F=<...>
```

In diesem Fall wird die Komprimierung auf zwei unterschiedliche Arten durchgeführt, abhängig davon, ob eine Großkreisinterpolation durchgeführt wird oder nicht. Wenn keine Großkreisinterpolation stattfindet, dann wird die komprimierte Orientierungsänderung durch axiale Polynome für die Rundachsen in üblicher Weise dargestellt.

### Konturgenauigkeit

Abhängig vom eingestellten Kompressionsmodus (MD20482 $MC_COMPRESSOR_MODE) werden für die Geometrieachsen und Orientierungsachsen bei der Komprimierung entweder die projektierten achsspezifischen Toleranzen (MD33100 $MA_COMPRESS_POS_TOL) oder die folgenden über Settingdaten einstellbaren kanalspezifischen Toleranzen wirksam:

SD42475 $SC_COMPRESS_CONTUR_TOL (Maximale Konturabweichung)

SD42476 $SC_COMPRESS_ORI_TOL (Maximale Winkelabweichung für die Werkzeugorientierung)

SD42477 $SC_COMPRESS_ORI_ROT_TOL (Maximale Winkelabweichung für den Drehwinkel des Werkzeugs) (nur bei 6-Achs Maschinen verfügbar)

### Aktivierung / Deaktivierung

Kompressor-Funktionen werden eingeschaltet durch die modalen G-Befehle COMPCAD und COMPSURF.

Beendet wird die Kompressor-Funktion mit COMPOF.

Siehe " AUTOHOTSPOT ".

---

### Hinweis

Die Orientierungsbewegung wird nur komprimiert bei aktiver Großkreisinterpolation (d. h., die Änderung der Werkzeugorientierung erfolgt in der Ebene, die von Start- und Endorientierung aufgespannt wird).

Eine Großkreisinterpolation wird unter den folgenden Bedingungen durchgeführt:

- MD21104 $MC_ORI_IPO_WITH_G_CODE = 0,
  ORIWKS ist aktiv und
  Orientierung ist mittels Vektoren programmiert (mit A3, B3, C3 bzw. A2, B2, C2).
- MD21104 $MC_ORI_IPO_WITH_G_CODE = 1 und
  ORIVECT bzw. ORIPLANE ist aktiv.
  Die Werkzeugorientierung kann entweder als Richtungsvektor oder mit Rundachspositionen programmiert sein. Ist einer der G-Befehle ORICONxx oder ORICURVE aktiv oder sind Polynome für die Orientierungswinkel (PO[PHI] und PO[PSI]) programmiert, wird keine Großkreisinterpolation durchgeführt.

---

### Beispiel

Im nachfolgenden Programmbeispiel wird ein Kreis, der durch einen Polygonzug angenähert ist, komprimiert. Die Werkzeugorientierung bewegt sich dabei synchron dazu auf einem Kegelmantel. Obwohl die aufeinander folgenden programmierten Orientierungsänderungen unstetig verlaufen, generiert die Kompressor-Funktion einen glatten Verlauf der Orientierung.

```
Programmierung                          Kommentar
DEF INT ANZAHL=60
DEF REAL RADIUS=20
DEF INT COUNTER
DEF REAL WINKEL
N10 G1 X0 Y0 F5000 G64

$SC_COMPRESS_CONTUR_TOL=0.05            ; Maximale Abweichung der Kontur = 0.05 mm
$SC_COMPRESS_ORI_TOL=5                  ; Maximale Abweichung der Orientierung
                                        = 5 Grad


TRAORI
COMPCAD                                 ; Es wird ein Kreis gefahren, der aus Polygo-
                                        nen gebildet wird. Die Orientierung bewegt
                                        sich dabei auf einem Kegel um die Z-Achse
                                        mit einem Öffnungswinkel von 45 Grad.
N100 X0 Y0 A3=0 B3=-1 C3=1
N110 FOR COUNTER=0 TO ANZAHL
N120 WINKEL=360*COUNTER/ANZAHL
N130 X=RADIUS*cos(WINKEL) Y=RADIUS*sin(WINKEL)
A3=sin(WINKEL) B3=-cos(WINKEL) C3=1
N140 ENDFOR
```

## 3.9.8 Glättung des Orientierungsverlaufs ein-/ausschalten (ORISON, ORISOF)

Die "Glättung des Orientierungsverlaufs" wird im Teileprogramm durch die Befehle der G-Gruppe 61 ein-/ausgeschaltet. Die Befehle sind modal wirksam.

### Voraussetzungen

- System mit 5/6-Achs-Transformation.
- Kompressor-Funktion COMPCAD ist aktiv.

### Syntax

```
ORISON
...
ORISOF
```

## Bedeutung

| | |
|---|---|
| ORISON: | Glättung des Orientierungsverlaufs einschalten |
| ORISOF: | Glättung des Orientierungsverlaufs ausschalten |

## Beispiel

```
Programmcode                    Kommentar
...
TRAORI()                        ; Einschalten der Orientierungstransformation.
COMPCAD                         ; Einschalten der Kompressor-Funktion COMPCAD.
ORISON                          ; Einschalten der Orientierungsglättung.
$SC_ORISON_TOL=1.0              ; Maximale Winkelabweichung der Werkzeugorientierung
                                = 1,0 Grad.
G91
X10 A3=1 B3=0 C3=1
X10 A3=-1 B3=0 C3=1
X10 A3=1 B3=0 C3=1
X10 A3=-1 B3=0 C3=1
X10 A3=1 B3=0 C3=1
X10 A3=-1 B3=0 C3=1
X10 A3=1 B3=0 C3=1
X10 A3=-1 B3=0 C3=1
X10 A3=1 B3=0 C3=1
X10 A3=-1 B3=0 C3=1
...
ORISOF                          ; Ausschalten der Orientierungsglättung.
...
```

Die Orientierung wird um 90 Grad in der XZ–Ebene von -45 bis +45 Grad geschwenkt. Durch die Glättung des Orientierungsverlaufs erreicht die Orientierung nicht mehr die maximalen Winkelwerte von -45 bzw. +45 Grad.

# 3.9.9 Kinematische Transformation

## 3.9.9.1 Stirnseitentransformation einschalten (TRANSMIT)

Die Stirnseitentransformation (TRANSMIT) wird im Teileprogramm oder Synchronaktion über die Anweisung `TRANSMIT` eingeschaltet.

## Syntax

```
TRANSMIT
TRANSMIT(<n>)
```

## Bedeutung

| | |
|---|---|
| TRANSMIT: | TRANSMIT mit erstem TRANSMIT-Datensatz einschalten |
| TRANSMIT(n): | TRANSMIT mit <n>-tem TRANSMIT-Datensatz einschalten |

**Hinweis**

Eine im Kanal aktive Transformation TRANSMIT wird ausgeschaltet durch:

- Transformation ausschalten: TRAFOOF
- Einschalten einer anderen Transformation: z. B. TRACYL, TRAANG, TRAORI

### 3.9.9.2 Zylindermanteltransformation einschalten (TRACYL)

Die Zylindermanteltransformation (TRACYL) wird im Teileprogramm oder Synchronaktion über die Anweisung TRACYL eingeschaltet.

## Syntax

```
TRACYL(<d>)
TRACYL(<d>,<n>)
TRACYL(<d>,<n>,<k>)
```

## Bedeutung

| | | |
|---|---|---|
| TRACYL(<d>): | TRACYL mit erstem TRACYL-Datensatz und Arbeitsdurchmesser <d> einschalten | |
| TRACYL (<d>,<n>): | TRACYL mit <n>-tem TRACYL-Datensatz und Arbeitsdurchmesser <d> einschalten | |
| <d>: | Bezugs- bzw. Arbeitsdurchmesser<br>Der Wert muss größer 1 sein. | |
| <n>: | TRACYL-Datensatznummer (optional) | |
| | Wertebereich: | 1, 2 |
| <k>: | Der Parameter <k> ist nur relevant für Transformationstyp 514 | |
| | k = 0: | ohne Nutwandkorrektur |
| | k = 1: | mit Nutwandkorrektur |
| | Wird der Parameter nicht angegeben, wirkt die parametrierte Grundstellung:<br>$MC_TRACYL_DEFAULT_MODE_<n><br>mit <n> = TRACYL-Datensatznummer | |

**Hinweis**

Eine im Kanal aktive Transformation TRACYL wird ausgeschaltet durch:

- Transformation ausschalten: `TRAFOOF`
- Einschalten einer anderen Transformation: z. B. `TRAANG`, `TRANSMIT`, `TRAORI`

## Beispiel

| Programmcode | Kommentar |
|---|---|
| ... | |
| N40 TRACYL(40.) | ; TRACYL mit erstem TRACYL-Datensatz und Arbeitsdurchmesser 40 mm einschalten. |
| ... | |

## Weitere Informationen

### Programmstruktur

Ein Teileprogramm zum Fräsen einer Nut mit TRACYL-Transformation 513 (TRACYL mit Nutwandkorrektur) besteht in der Regel aus folgenden Schritten:

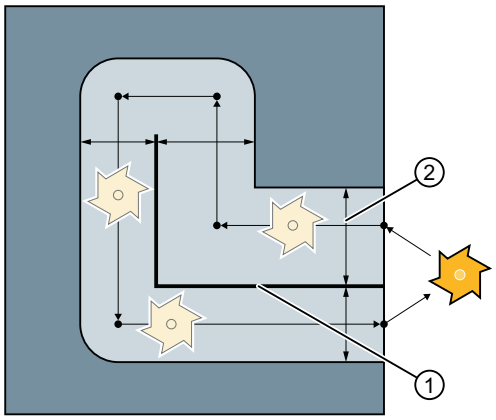
1. Werkzeug anwählen.
2. TRACYL anwählen.
3. Passende Koordinatenverschiebung (FRAME) anwählen.
4. Positionieren.
5. OFFN programmieren.
6. WRK anwählen.
7. Anfahrsatz (Einfahren der WRK und Anfahren der Nutwand).
8. Kontur der Nutmittenlinie.
9. WRK abwählen.
10. Abfahrsatz (Ausfahren der WRK und Wegfahren von der Nutwand).
11. Positionieren.
12. TRAFOOF.
13. Ursprüngliche Koordinatenverschiebung (FRAME) wieder anwählen.

### Kontur-Offset (OFFN)

Um mit der TRACYL-Transformation 513 Nuten zu fräsen, wird im Teileprogramm die Mittenlinie der Nut und über die Adresse OFFN **die halbe Nutbreite** programmiert.

Um eine Beschädigung der Nutwand zu vermeiden, wird OFFN erst mit angewählter Werkzeugradiuskorrektur wirksam.

Eine Änderung von OFFN innerhalb des Teileprogramms ist möglich. Damit kann die Nutmittenlinie aus der Mitte verschoben werden:




① OFFN
② Programmierte Bahn

**Hinweis**

OFFN sollte mindestens so groß wie der Werkzeugradius sein, um eine Beschädigung der gegenüberliegenden Nutwand auszuschließen.

**Hinweis**

OFFN mit TRACYL wirkt sich anders aus als ohne TRACYL. Da OFFN auch ohne TRACYL bei aktiver WRK eingerechnet wird, sollte OFFN nach TRAFOOF wieder zu Null gesetzt werden.

| ACHTUNG |
| --- |
| **Wirkung von OFFN ist vom Transformationstyp abhängig**<br>Bei der TRACYL-Transformation 513 (TRACYL mit Nutwandkorrektur) wird für OFFN die halbe Nutbreite programmiert.<br>Bei der TRACYL-Transformation 512 (TRACYL ohne Nutwandkorrektur) wirkt der Wert von OFFN dagegen als Aufmass zur WRK. |

### Werkzeugradiuskorrektur (WRK)

Bei der TRACYL-Transformation 513 wird die WRK nicht relativ zur Nutwand, sondern zur programmierten Mitte der Nut eingerechnet. Damit das Werkzeug links von der Nutwand fährt, ist statt G41 die Anweisung G42 zu programmieren oder der Wert von OFFN mit negativem Vorzeichen anzugeben.

### Werkzeugdurchmesser

Mit TRACYL und einem Werkzeug, dessen Durchmesser kleiner als die Nutbreite ist, wird nicht dieselbe Nutwandgeometrie erzeugt, wie mit einem Werkzeug, dessen Durchmesser gleich der Nutbreite ist. Zur Verbesserung der Genauigkeit wird empfohlen, den Werkzeugdurchmesser nur wenig kleiner als die Nutbreite zu wählen.

### Achsnutzung

**Hinweis**

Folgende Achsen können nicht als Positionierachse bzw. Pendelachse verwendet werden:

- die Geometrieachse in Umfangsrichtung der Zylindermantelfläche (Y-Achse)
- die zusätzliche Linearachse bei Nutwandkorrektur (Z-Achse)

## 3.9.9.3 Schiefwinkeltransformation mit programmierbarem Winkel einschalten (TRAANG)

Die Schiefwinkeltransformation mit programmierbarem Winkel wird im Teileprogramm oder Synchronaktion über die Anweisung `TRAANG` eingeschaltet.

### Syntax

```
TRAANG
TRAANG()
TRAANG(,<n>)
TRAANG(<α>)
TRAANG(<α>,<n>)
```

### Bedeutung

| | | |
|---|---|---|
| `TRAANG`:<br>`TRAANG()`: | TRAANG mit erstem TRAANG-Datensatz und zuletzt gültigem Winkel <α> einschalten | |
| `TRAANG(,<n>)`: | TRAANG mit <n>-tem TRAANG-Datensatz und zuletzt gültigem Winkel <α> einschalten | |
| `TRAANG(<α>)`: | TRAANG mit erstem TRAANG-Datensatz und Winkel <α> einschalten | |
| `TRAANG(<α>,<n>)`: | TRAANG mit <n>-tem TRAANG-Datensatz und Winkel <α> einschalten | |
| <α>: | Winkel der schrägstehenden Achse (optional) | |
| | Wertebereich: | -90° < α < + 90° |
| | Wird kein Winkel angegeben, wirkt die im Maschinendatum parametrierte Grundstellung:<br>MD2xxxx $MC_TRAANG_ANGLE_<n> | |
| `<n>`: | TRAANG-Datensatznummer (optional) | |
| | Wertebereich: | 1, 2 |

**Hinweis**

Eine im Kanal aktive Schiefwinkeltransformation `TRAANG` wird ausgeschaltet durch:

- Transformation ausschalten: `TRAFOOF`
- Einschalten einer anderen Transformation: z. B. `TRACYL`, `TRANSMIT`, `TRAORI`

## Beispiel

| Programmcode | Kommentar |
|---|---|
| N20 TRAANG(45) | ; TRAANG mit erstem TRAANG-Datensatz und Winkel 45° einschalten. |

### 3.9.9.4 Schräges Einstechen an Schleifmaschinen (G5, G7)

Die G-Befehle G7 und G5 dienen zur Vereinfachung der Programmierung für das schräge Einstechen an Schleifmaschinen mit Transformation "Schräge Achse (TRAANG)", so dass beim Einstechen ausschließlich die schräge Achse verfährt.

Zu Programmieren ist dabei nur die gewünschte Endposition der Einstechbewegung in X und Z. Die zugehörige Startposition wird von der NC bei G7, ausgehend von der aktuellen Position der X-Achse, der programmierten Endposition und dem Winkel α der schrägen Achse, berechnet und angefahren.

Die Startposition ergibt sich aus dem Schnittpunkt der beiden Geraden:

- Gerade parallel zur Z-Achse im Abstand der aktuellen Position der X-Achse
- Gerade parallel zur schrägen Achse durch die programmierte Endposition

Mit dem nachfolgenden G5 wird die schräge Achse auf die programmierte Endposition verfahren.

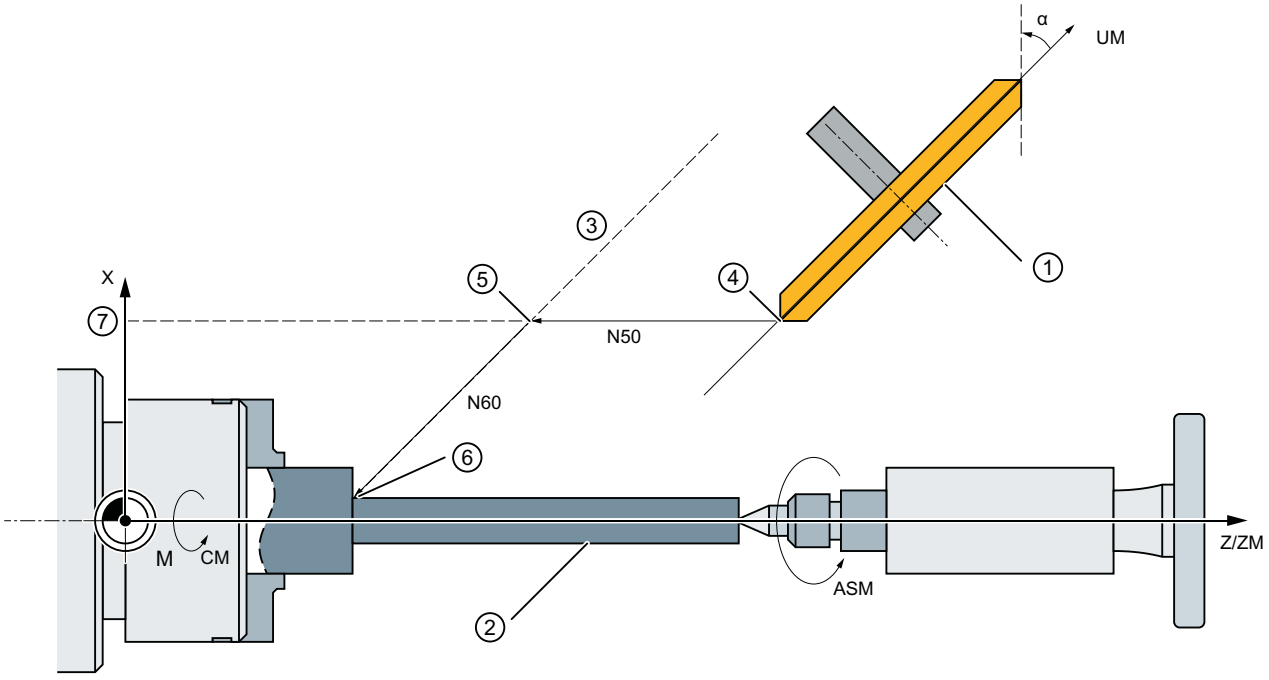
## Syntax

```
G7 <Endpos_X> <Endpos_Z>
G5 <Endpos_X>
```

## Bedeutung

| | |
|---|---|
| G7: | Startposition zum schrägen Einstechen berechnen und anfahren |
| G5: | Verfahren der schrägen Achse auf die programmierte Endposition |
| <Endpos_X>: | Endposition der X-Achse |
| <Endpos_Z>: | Endposition der Z-Achse |

## Beispiel




① Schleifscheibe
② Werkstück
③ Parallele zur schrägen Achse durch die programmierte Endposition
④ Ausgangsposition
⑤ Einstechen: Startposition
⑥ Einstechen: Endposition
⑦ Parallele zur Z-Achse im Abstand der aktuellen Position der X-Achse
X Geometrieachse
Z Geometrieachse
ZM Maschinenachse
UM Maschinenachse

Bild 3-5 Schräge Achse programmieren

| Programmcode | Kommentar |
|---|---|
| N... G18 | ; XZ-Ebene anwählen. |
| N40 TRAANG(45.0) | ; Transformation TRAANG einschalten, Winkel = 45°. |
| N50 G7 X40 Z70 F4000 | ; Startposition berechnen und anfahren. |
| N60 G5 X40 F100 | ; Schräge Achse auf Endposition verfahren. |
| N70 ... | |

## 3.9.10 Verkettete Transformation einschalten (TRACON)

Eine projektierte verkettete Transformation wird im Teileprogramm oder Synchronaktion über die Anweisung TRACON eingeschaltet.

## Syntax

```
TRACON(<Trafo_Nr>,<Par_1>,...,<Par_n>,<Par_n+1>)
...
TRAFOOF
```

## Bedeutung

| | | | |
|---|---|---|---|
| TRACON: | Verkettete Transformation einschalten<br>Eine zuvor aktivierte andere Transformation wird durch TRACON() implizit ausgeschaltet. | | |
| <Trafo_Nr>: | Nummer der verketteten Transformation | | |
| | Typ: | INT | |
| | Wertebereich: | 0 ... 2 | |
| | Wert: | 0, 1 | erste/einzige verkettete Transformation |
| | | 2 | zweite verkettete Transformation |
| | | keine Angabe | gleichbedeutend mit 0 bzw. 1 |
| | | **Hinweis:**<br>Werte ungleich 0, 1, 2 erzeugen einen Fehleralarm. | |
| <Par_1>, ..., <Par_n>, <Par_n+1>: | Parameter für die miteinander verketteten Transformationen | | |
| | <Par_1>, ..., <Par_n> | Parameter der ersten Transformation in der Kette<br>Die tatsächliche Anzahl ist vom Transformationstyp abhängig:<br>• TRAORI: 2 Parameter<br>• TRACYL: 1 bis 2 Parameter | |
| | <Par_n+1> | Parameter der zweiten Transformation in der Kette (TRAANG)<br>≙ Winkel der schrägen Achse | |
| | Bei nicht gesetzten Parametern werden die Voreinstellungen oder die zuletzt benutzten Parameter wirksam. Durch Kommasetzung muss dafür gesorgt werden, dass die angegeben Parameter in der Reihenfolge ausgewertet werden, in der sie erwartet werden, wenn für vorher stehende Parameter Voreinstellungen wirken sollen. Insbesondere muss bei Angabe mindestens eines Parameters vor diesem ein Komma stehen, auch wenn die Angabe von <Trafo_Nr> nicht notwendig ist, also beispielsweise TRACON( , 3.7). | | |
| TRAFOOF: | Die zuletzt eingeschaltete (verkettete) Transformation ausschalten | | |

## Beispiel

| **Programmcode** | **Kommentar** |
|---|---|
| ... | |
| N230 TRACON(1,45.) | ; Erste verkettete Transformation einschalten. |

| Programmcode | Kommentar |
|---|---|
| | ; Die vorher aktive Transformation wird automatisch abgewählt. |
| | ; Der Winkel für die schräge Achse ist 45°. |
| ... | |
| N330 TRACON(2,40.) | ; Zweite verkettete Transformation einschalten. |
| | ; Der Winkel für die schräge Achse ist 40°. |
| ... | |
| N380 TRAFOOF | ; Zweite verkettete Transformation ausschalten. |
| ... | |

## 3.9.11 Kartesisches PTP-Fahren

### 3.9.11.1 Kartesisches PTP-Fahren ein-/ausschalten (PTP, PTPG0, PTPWOC, CP)

Das kartesische Punkt-zu-Punkt- bzw. PTP-Fahren wird im NC-Programm mit den Befehlen der G-Gruppe 49 ein-/ausgeschaltet.

Die Befehle sind modal wirksam. Voreinstellung ist das Verfahren mit einer kartesischen Bahnbewegung (CP).

Anders als bei CP wird bei aktivem PTP-Fahren nur noch eine Transformation des kartesischen Zielpunkts durchgeführt und die Maschinenachsen werden synchron verfahren.

Damit der kartesische Zielpunkt eindeutig in Maschinenachswerte umgerechnet werden kann, sind zusätzlich zu den Positions- und Winkelangaben noch Informationen notwendig, die die Achsstellungen kennzeichnen. Diese Angaben erfolgen über die einstellbaren Adressen STAT (Seite 718) und TU (Seite 722).

#### Voraussetzung

Transformation TRAORI, TRANSMIT, RCTRA oder ROBX ist aktiv.

#### Syntax

```
PTP / PTPG0 / PTPWOC
...
CP
```

## Bedeutung

| | |
|---|---|
| PTP: | Punkt-zu-Punkt-Bewegung PTP einschalten<br>Die programmierte kartesische Position in G0- und G1-Sätzen wird mit einer Synchronachsbewegung angefahren. |
| PTPG0: | Punkt-zu-Punkt-Bewegung PTPG0 einschalten<br>Nur in G0-Sätzen wird die programmierte kartesische Position mit einer Synchronachsbewegung angefahren. In G1-Sätzen wird auf die Bahnbewegung CP umgeschaltet. |
| PTPWOC: | Punkt-zu-Punkt-Bewegung PTPWOC einschalten (nur möglich bei aktiver Orientierungstransformation)<br>Wie PTP, allerdings ohne Ausgleichsbewegungen, die durch die Bewegungen von Rundachsen und Orientierungsachsen verursacht sind. |
| CP: | Punkt-zu-Punkt-Bewegung ausschalten und Bahnbewegung CP einschalten<br>Mit CP wird eine kartesische Bahnbewegung ausgeführt. |

**Hinweis**

**PTPWOC**

In Kombination mit einer RCTRA- oder ROBX-Transformation ist die Verwendung von PTPWOC nicht sinnvoll!

## Beispiele

Siehe:

- Beispiel 1: PTP-Fahren eines 6-Achs-Roboters mit ROBX-Transformation (Seite 725)
- Beispiel 2: PTP-Fahren bei generischer 5-Achs-Transformation (Seite 726)
- Beispiel 3: PTPG0 und TRANSMIT (Seite 727)

### 3.9.11.2 Stellung der Gelenke angeben (STAT)

Die Positionsangabe mit kartesischen Koordinaten und die Angabe der Werkzeugorientierung reichen nicht aus, um die Maschinenstellung eindeutig festzulegen, da bei gleicher Werkzeugorientierung mehrere Gelenkstellungen möglich sind. Je nachdem, um welche Kinematik es sich handelt, existieren bis zu 8 unterschiedliche Gelenkstellungen. Diese unterschiedlichen Gelenkstellungen sind transformationsspezifisch.

Um Mehrdeutigkeiten zu vermeiden, wird die Stellung der Gelenke unter der Adresse STAT angegeben.

**Hinweis**

Die Steuerung berücksichtigt programmierte STAT-Werte nur bei PTP-Bewegungen. Bei CP-Bewegungen werden sie ignoriert, da beim Verfahren mit aktiver Transformation ein Stellungswechsel normalerweise nicht möglich ist. Beim Verfahren mit aktivem CP wird die Stellung für den Zielpunkt vom Startpunkt übernommen.

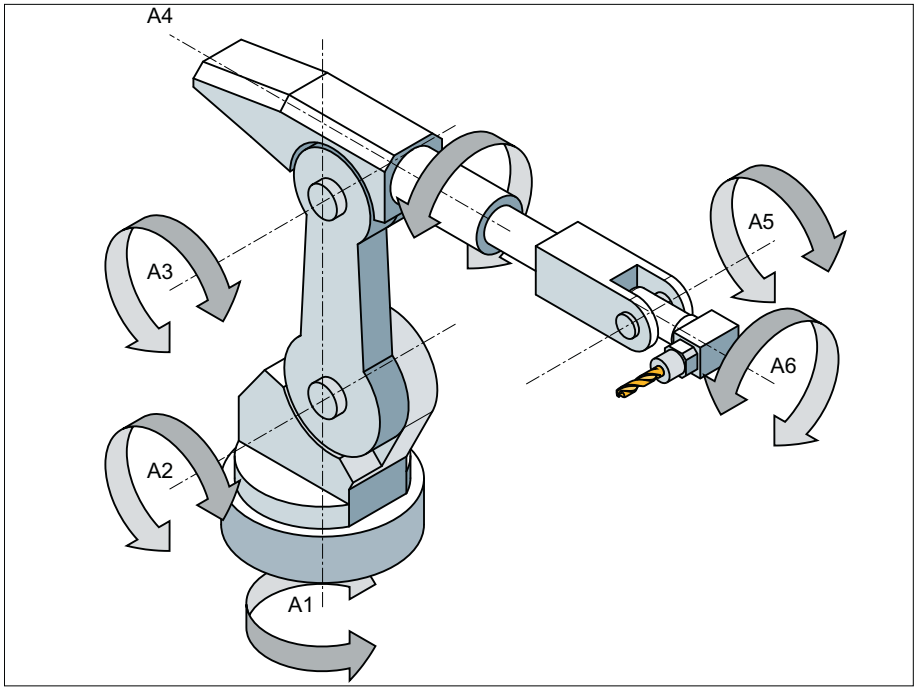
## Syntax

```
STAT=<Wert>
```

## Bedeutung

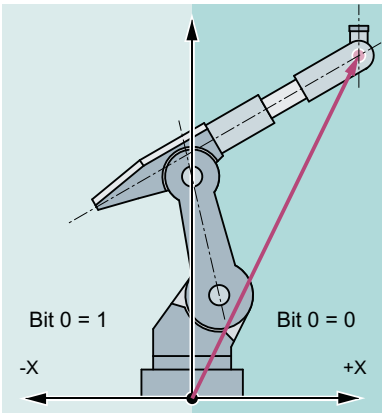
| | |
|---|---|
| STAT: | Einstellbare Adresse zur Angabe der Gelenkstellungen |
| <Wert>: | Binär- oder Dezimalwert<br>Enthält für jede der möglichen Stellungen ein Bit. Die Bedeutung der Bits wird von der jeweiligen Transformation festgelegt. |

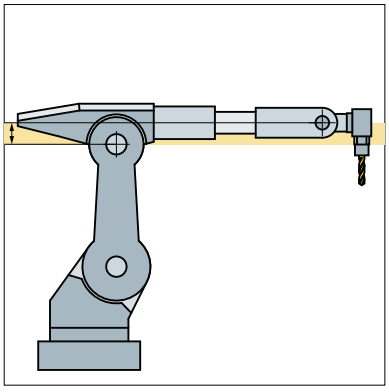
Die Anwendung von STAT soll am Beispiel eines 6-achsigen Knickarmroboters mit Frässpindel veranschaulicht werden. Die kinematische Transformation soll über die Robotertransformation ROBX erfolgen (Voraussetzung: Compile-Zyklus "RMCC/ROBX Transformation Robotik erweitert" ist geladen und aktiv).




Die Achsen A1, A2 und A3 sind die Hauptachsen des Knickarmroboters. Mit den Hauptachsen werden die Achsen A4, A5 und A6, die als Kopf- oder Handachsen bezeichnet werden, im Arbeitsraum positioniert. Durch die zusätzlichen Bewegungsmöglichkeiten der Handachsen kann die Frässpindel im Raum so orientiert werden, wie es für die Bearbeitungsaufgabe erforderlich ist. Bei gleicher Werkzeugorientierung sind dabei verschiedene Gelenkstellungen möglich.

Die Auswahl der für die Bearbeitung zu verwendenden Gelenkstellungen erfolgt durch Programmierung von Bit 0 ... 2 der einstellbaren Adresse STAT:

<table>
<tr><td rowspan="3">Bit 0</td><td colspan="2">Position des Schnittpunkts der Handachsen (A4, A5, A6)</td><td rowspan="3"> 

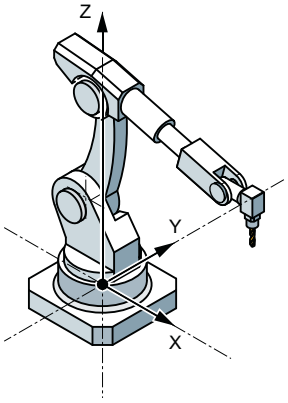
Beispiel: Der Schnittpunkt der Handachsen befindet sich im Grundbereich</td></tr>
<tr><td>= 0</td><td>Grundbereich (Shoulder Right)<br>Der Roboter befindet sich im Grundbereich, wenn der X-Wert des Schnittpunkts der Handachsen, bezogen auf das A1-Koordinatensystem, positiv ist.</td></tr>
<tr><td>= 1</td><td>Überkopfbereich (Shoulder Left)<br>Der Roboter befindet sich im Überkopfbereich, wenn der X-Wert des Schnittpunkts der Handachsen, bezogen auf das A1-Koordinatensystem, negativ ist.</td></tr>
<tr><td rowspan="4">Bit 1</td><td colspan="3">Position von Achse 3<br>Der Winkel, bei dem sich der Wert von Bit 1 ändert, ist abhängig vom Robotertyp. Für Roboter, deren Achsen 3 und 4 sich schneiden, gilt:</td></tr>
<tr><td>= 0</td><td colspan="2">A3 < 0° (Elbow Down)</td></tr>
<tr><td>= 1</td><td colspan="2">A3 ≥ 0° (Elbow Up)</td></tr>
<tr><td colspan="2">Hinweis:<br>Bei Robotern mit einem Offset zwischen Achse 3 und 4 ist der Winkel, bei dem sich der Wert von Bit 1 ändert, von der Größe dieses Offsets abhängig.</td><td>  <br>Offset zwischen A3 und A4</td></tr>
<tr><td rowspan="3">Bit 2</td><td colspan="3">Position von Achse 5</td></tr>
<tr><td>= 0</td><td colspan="2">A5 ≥ 0° (No Handflip)</td></tr>
<tr><td>= 1</td><td colspan="2">A5 < 0° (Handflip)</td></tr>
</table>

Programmbeispiel:

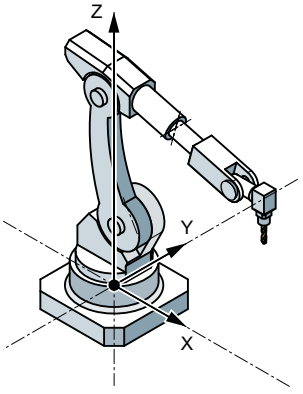
| **Programmcode** | **Kommentar** |
|---|---|
| ... | |
| N14 T="T8MILLD20" D1 | ; $TC_DP3[1,1]=132.95 |

| **Programmcode** | **Kommentar** |
|---|---|
| N16 ORIMKS | |
| N17 G1 PTP X1665.67 Y0 Z1377.405 A=0 B=0 C=0 **STAT=...** F2000 | ; Der STAT-Wert bestimmt die Gelenkstellungen (s. u.). |
| ... | |

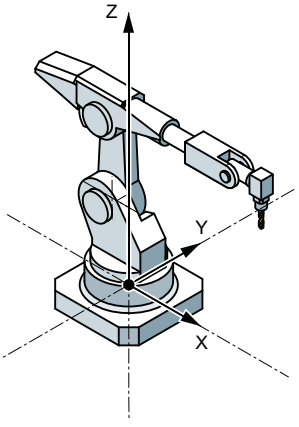
STAT=1 ('B001') → Shoulder Left
→ Elbow Down
→ No Handflip



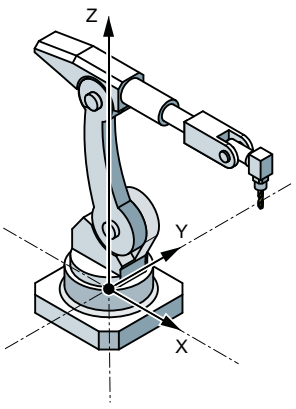

STAT=2 ('B010') → Shoulder Right
→ Elbow Up
→ No Handflip




STAT=5 ('B101') → Shoulder Left
→ Elbow Down
→ Handflip



`STAT=6 ('B110')` → Shoulder Right
→ Elbow Up
→ Handflip




**TRANSMIT**

Bei TRANSMIT wird die Adresse STAT benutzt, um die Mehrdeutigkeit hinsichtlich des Pols aufzulösen.

Wenn die Rundachse um 180º drehen muss bzw. die Kontur bei CP durch den Pol führen würde, gilt:

| | | |
|---|---|---|
| Bit 0 | Nur relevant bei $MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 1 bzw. 2: | |
| | = 0 | Rundachse verfährt um +180º bzw. dreht im Uhrzeigersinn. |
| | = 1 | Rundachse verfährt um -180º bzw. dreht gegen den Uhrzeigersinn. |
| Bit 1 | Nur relevant bei $MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 0: | |
| | = 0 | Es wird durch den Pol gefahren. Die Rundachse dreht nicht. |
| | = 1 | Es wird um den Pol gedreht. Dabei ist Bit 0 von STAT relevant. |

### 3.9.11.3 Vorzeichen der Achswinkel angeben (TU)

Um bei rotatorischen Achsen auch Achswinkel, die größer +180° oder kleiner -180° sind, ohne besondere Verfahrstrategie (z. B. Zwischenpunkte) anfahren zu können, müssen unter der einstellbaren Adresse TU die Vorzeichen der Achswinkel angegeben werden.

**Hinweis**

Die Steuerung berücksichtigt programmierte TU-Werte nur bei PTP-Bewegungen. Bei CP-Bewegungen werden sie ignoriert.

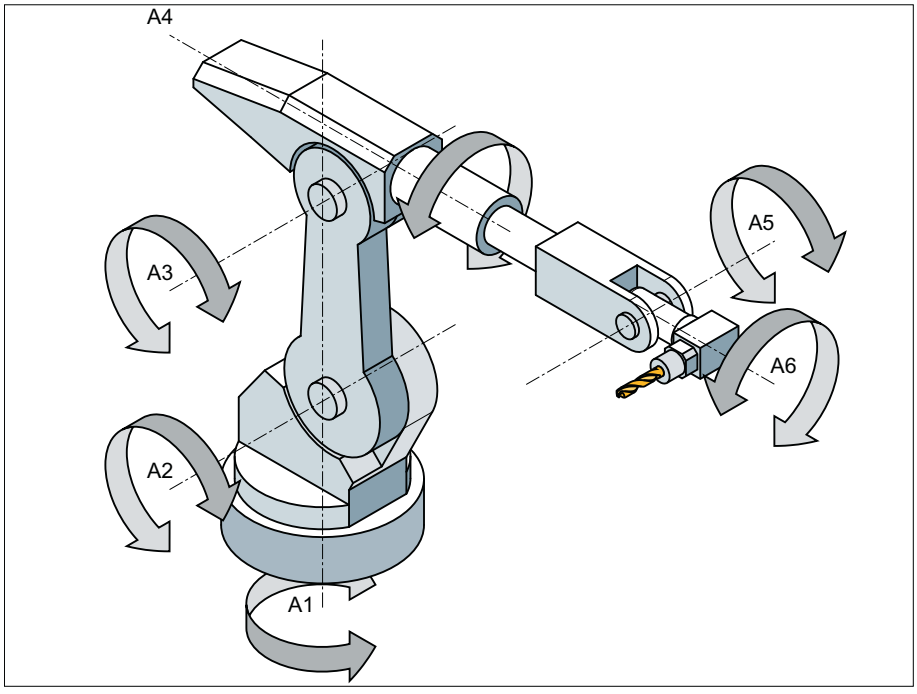
## Syntax

```
TU=<Wert>
```

## Bedeutung

| | | | | |
|---|---|---|---|---|
| TU: | Einstellbare Adresse zur Angabe der Achswinkelvorzeichen | | | |
| <Wert>: | Binär- oder Dezimalwert<br>Für jede Achse, die in die Transformation eingeht, gibt es ein Bit, das das Vorzeichen des Achswinkels (θ) und damit die Verfahrrichtung anzeigt. | | | |
| | Bit | = 0 | Achswinkelvorzeichen: + | Achswinkelbereich: 0° ≤ θ < 360° |
| | | = 1 | Achswinkelvorzeichen: - | Achswinkelbereich: - 360° < θ < 0° |

Beispiel: 6-achsiger Knickarmroboter




| Bit | Bedeutung | Wert | Achswinkel-vorzeichen | Achswinkel |
|---|---|---|---|---|
| Bit 0 [1)] | Vorzeichen für den Achswinkel von A1 | = 0 | + | ≥ 0° |
| | | = 1 | - | < 0° |
| Bit 1 [1)] | Vorzeichen für den Achswinkel von A2 | = 0 | + | ≥ 0° |
| | | = 1 | - | < 0° |
| Bit 2 [1)] | Vorzeichen für den Achswinkel von A3 | = 0 | + | ≥ 0° |
| | | = 1 | - | < 0° |
| Bit 3 [1)] | Vorzeichen für den Achswinkel von A4 | = 0 | + | ≥ 0° |
| | | = 1 | - | < 0° |
| Bit 4 [1)] | Vorzeichen für den Achswinkel von A5 | = 0 | + | ≥ 0° |
| | | = 1 | - | < 0° |

| Bit | Bedeutung | Wert | Achswinkel-vorzeichen | Achswinkel |
|---|---|---|---|---|
| Bit 5 [1] | Vorzeichen für den Achswinkel von A6 | = 0 | + | ≥ 0° |
| | | = 1 | - | < 0° |

[1] Die tatsächlichen TU-Bitnummern ergeben sich aus den Kanalachsnummern der Roboterachsen! Im Beispiel sind die Roboterachsen (A1 bis A6) die ersten sechs Achsen im Kanal, folglich werden die TU-Bits 0 ... 5 verwendet. Bei einer anderen Kanalachszuordnung der Roboterachsen würden sich die TU-Bitnummern der Roboterachsen entsprechend ändern (z. B.: Roboterachsen sind die 3. bis 8. Kanalachse, d. h. es werden die TU-Bits 2 ... 7 für die Roboterachsen verwendet).

TU=19 (entspricht TU='B010011) würde also bedeuten:

| Bit | Wert | | Achswinkel |
|---|---|---|---|
| 0 | = 1 | ⇒ | $\theta_{A1} < 0°$ |
| 1 | = 1 | ⇒ | $\theta_{A2} < 0°$ |
| 2 | = 0 | ⇒ | $\theta_{A3} \geq 0°$ |
| 3 | = 0 | ⇒ | $\theta_{A4} \geq 0°$ |
| 4 | = 1 | ⇒ | $\theta_{A5} < 0°$ |
| 5 | = 0 | ⇒ | $\theta_{A6} \geq 0°$ |

**Hinweis**

Bei Achsen mit einem Verfahrbereich > ±360° wird immer auf kürzestem Weg verfahren, da die Achsstellung durch die TU-Information nicht eindeutig bestimmbar ist.

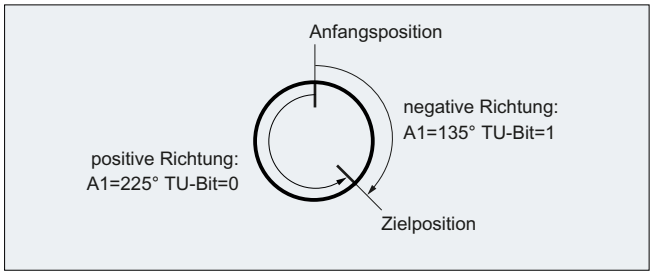
Wird bei einer Position kein TU programmiert, wird in Abhängigkeit von MD30455 $MA_MISC_FUNCTION_MASK der kürzere oder längere Weg verfahren.

**TRANSMIT**

Beim PTP-Fahren bei aktivem TRANSMIT ist die Adresse TU ohne Bedeutung!

## Beispiel

Die in der folgenden Abbildung angegebene Rundachsposition kann in negativer oder in positiver Richtung angefahren werden. Unter der Adresse A1 wird die Winkelposition programmiert. Erst durch die TU-Angabe wird die Verfahrrichtung eindeutig.




### 3.9.11.4 Beispiel 1: PTP-Fahren eines 6-Achs-Roboters mit ROBX-Transformation

Im folgenden Anwendungsbeispiel werden das kartesische PTP-Fahren und die damit im Zusammenhang stehenden NC-Befehle exemplarisch gezeigt.

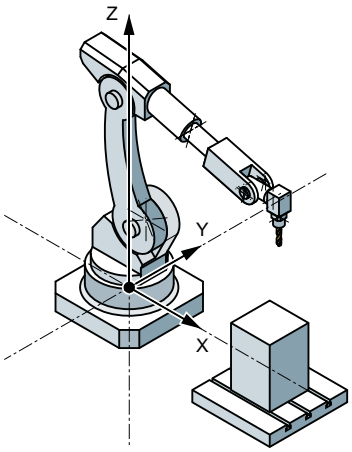



Bild 3-6 6-achsiger Knickarmroboter mit Frässpindel

```
N1 G90
N2 T="T8MILLD20" D1 M6
N3 TRAORI
;$P_UIFR[1]=CTRANS(X,1500,Y,0,Z,400):CROT(X,0,Y,0,Z,-90)
N4 G54
N5 M3 S20000
N6 ORIWKS
N7 ORIVIRT1
N8 CYCLE832(0.01,_FINISH,1)
;HOME
N9 TRAFOOF
N10 G0 RA1=0.0000 RA2=-90.0000 RA3=90.0000 A=0.0000 B=90.0000 C=0.0000
N11 TRAORI
```

```
N12 G54
N13 G0 PTP X1369.2426 Y956.7528 Z502.5517 A=135.5761 B=-33.2223
C=161.1435 STAT='B010' TU='B001011'
N14 G0 X1355.1242 Y1014.9394 Z424.9695 A=135.8491 B=-33.1439
C=160.9941 STAT='B010' TU='B001011'
N15 G1 CP X1354.8361 Y1016.1269 Z423.3862 A=136.0635 B=-33.0819 C=160.8770
F1000
N16 G1 X1336.4283 Y1016.1269 Z426.6311 A=136.0484 B=-32.2151 C=160.9643
F2000
N17 G1 X1317.9831 Y1016.1269 Z429.6730 A=136.0175 B=-31.3394 C=161.0655
;HOME
N18 TRAFOOF
N19 G0 RA1=0.0000 RA2=-90.0000 RA3=90.0000 A=0.0000 B=90.0000 C=0.0000
N20 M30
```

### 3.9.11.5 Beispiel 2: PTP-Fahren bei generischer 5-Achs-Transformation

Annahme: Es liegt eine rechtwinklige CA-Kinematik zu Grunde.

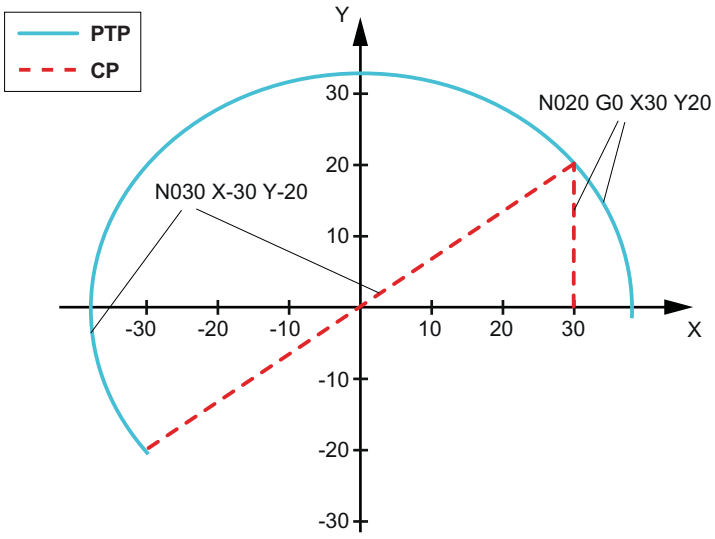
| **Programmcode** | **Kommentar** |
|---|---|
| **TRAORI** | ; Transformation CA-Kinematik ein |
| **PTP** | ; PTP-Fahren einschalten |
| N10 A3=0 B3=0 C3=1 | ; Rundachspositionen C=0 A=0 |
| N20 A3=1 B3=0 C3=1 | ; Rundachspositionen C=90 A=45 |
| N30 A3=1 B3=0 C3=0 | ; Rundachspositionen C=90 A=90 |
| N40 A3=1 B3=0 C3=1 **STAT=1** | ; Rundachspositionen C=270 A=–45 |

Eindeutige Anfahrstellung der Rundachsposition auswählen:

Im Satz N40 fahren dabei die Rundachsen durch die Programmierung von **STAT=1** den längeren Weg von ihrem Startpunkt (C=90, A=90) zum Endpunkt (C=270, A=–45). Mit **STAT=0** würden die Rundachsen dagegen den kürzeren Weg zum Endpunkt (C=90, A=45) fahren.

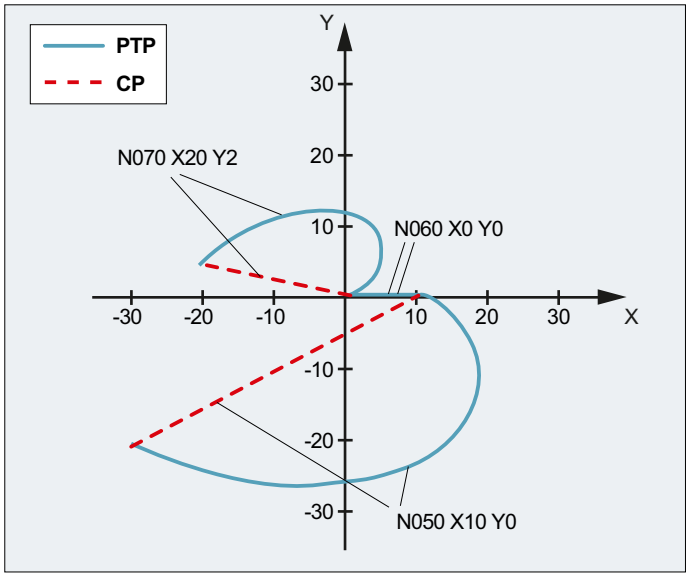
### 3.9.11.6 Beispiel 3: PTPG0 und TRANSMIT

## Umfahren des Pols mit PTPG0 und TRANSMIT




| **Programmcode** | **Kommentar** |
|---|---|
| N001 G0 X30 Z0 F10000 T1 D1 G90 | ; Ausgangsstellung Absolutmaß |
| N002 SPOS=0 | |
| N003 **TRANSMIT** | ; Transformation TRANSMIT |
| N010 **PTPG0** | ; Zu jedem G0-Satz automatisch PTP und danach wieder CP. |
| N020 G0 X30 Y20 | |
| N030 X-30 Y-20 | |
| N120 G1 X30 Y20 | |
| N110 X30 Y0 | |
| M30 | |

### Herausfahren aus dem Pol mit PTPG0 und TRANSMIT




| **Programmierung** | **Kommentar** |
|---|---|
| N001 G0 X90 Z0 F10000 T1 D1 G90 | ; Ausgangsstellung |
| N002 SPOS=0 | |
| N003 **TRANSMIT** | ; Transformation TRANSMIT |
| N010 **PTPG0** | ; Zu jedem G0-Satz automatisch PTP und danach wieder CP. |
| N020 G0 X90 Y60 | |
| N030 X-90 Y-60 | |
| N040 X-30 Y-20 | |
| N050 X10 Y0 | |
| N060 X0 Y0 | |
| N070 X-20 Y2 | |
| N170 G1 X0 Y0 | |
| N160 X10 Y0 | |
| N150 X-30 Y-20 | |
| M30 | |

## 3.9.12 Randbedingungen bei der Anwahl einer Transformation

### Funktion

Die Anwahl von Transformationen ist über Teileprogramm bzw. MDA möglich. Dabei ist zu beachten:

- Ein Bewegungszwischensatz wird nicht eingefügt (Fasen/Radien).
- Eine Spline-Satzfolge muss abgeschlossen sein; wenn nicht, erscheint eine Meldung.

- Werkzeugfeinkorrektur muss abgewählt sein (FTOCOF); wenn nicht, erscheint eine Meldung.
- Werkzeugradiuskorrektur muss abgewählt sein (G40); wenn nicht, erscheint eine Meldung.
- Eine aktivierte Werkzeuglängenkorrektur wird von der Steuerung in die Transformation übernommen.
- Der vor der Transformation wirksame aktuelle Frame wird von der Steuerung abgewählt.
- Eine aktive Arbeitsfeldbegrenzung wird für die von der Transformation betroffenen Achsen von der Steuerung abgewählt (entspricht WALIMOF).
- Schutzbereichsüberwachung wird abgewählt.
- Bahnsteuerbetrieb und Überschleifen werden unterbrochen.
- Alle in dem Maschinendatum angegebenen Achsen müssen satzbezogen synchronisiert sein.
- Getauschte Achsen werden zurückgetauscht; wenn nicht, erscheint eine Meldung.
- Bei abhängigen Achsen wird eine Meldung ausgegeben.

#### Werkzeugwechsel

Ein Werkzeugwechsel ist nur bei abgewählter Werkzeugradiuskorrektur zulässig.

Ein Wechsel der Werkzeuglängenkorrektur und eine An-/Abwahl der Werkzeugradiuskorrektur darf nicht im selben Satz programmiert sein.

#### Framewechsel

Alle Anweisungen, die sich nur auf das Basis-Koordinatensystem beziehen, sind erlaubt (FRAME, Werkzeugradiuskorrektur). Ein Framewechsel bei G91 (Kettenmaß) wird aber – anders als bei inaktiver Transformation – nicht gesondert behandelt. Das zu fahrende Inkrement wird im Werkstück-Koordinatensystem des neuen Frames ausgewertet – unabhängig davon, welches Frame im Vorgängersatz wirkte.

#### Ausschlüsse

Von der Transformation betroffene Achsen können nicht verwendet werden:

- als Preset-Achse (Alarm),
- für das Fixpunktanfahren (Alarm),
- zum Referenzieren (Alarm).

## 3.9.13 Transformation abwählen (TRAFOOF)

Mit der vordefinierten Prozedur TRAFOOF werden alle aktiven Transformationen und Frames ausgeschaltet.

**Hinweis**

Für die Abwahl der Transformation gelten dieselben Randbedingungen (Seite 728) wie für die Anwahl.

Danach benötigte Frames müssen durch erneute Programmierung aktiv geschaltet werden.

### Syntax

```
...
TRAFOOF
```

### Bedeutung

| | |
|---|---|
| `TRAFOOF`: | Alle aktiven Transformationen/Frames ausschalten |

# 3.10 Kinematische Ketten

## 3.10.1 Löschen von Komponenten (DELOBJ)

Die Funktion `DELOBJ()` "löscht" Komponenten durch Zurücksetzen der zugeordneten Systemvariablen auf ihren Defaultwert:

- Elemente von kinematischen Ketten
- Schutzbereiche, Schutzbereichselemente und Kollisionspaare
- Transformationsdaten

### Syntax

```
[<RetVal>=] DELOBJ(<CompType>[,,,<NoAlarm>)])
[<RetVal>=] DELOBJ(<CompType>,<Index1>[,,<NoAlarm>])
[<RetVal>=] DELOBJ(<CompType>[,<Index1>][,<Index2>][,<NoAlarm>])
```

## Bedeutung

| | | |
|---|---|---|
| DELOBJ: | Löschen von Elementen von kinematischen Ketten, Schutzbereichen, Schutzbereichselementen, Kollisionspaaren und Transformationsdaten | |
| <CompType>: | Typ der zu löschenden Komponente | |
| | Datentyp: | STRING |
| | **Wert**: "KIN_CHAIN_ELEM"<br>**Bedeutung**: Systemvariablen aller kinematischen Elemente: $NK_... | |
| | **Wert**: "KIN_CHAIN_SWITCH"<br>**Bedeutung**: Systemvariable $NK_SWITCH[<i>] | |
| | **Wert**: "KIN_CHAIN_ALL"<br>**Bedeutung**: Alle kinematischen Elemente und Schalter.<br>Gleichbedeutend mit dem sukzessiven Aufruf von DELOBJ mit "KIN_CHAIN_ELEM" und "KIN_CHAIN_SWITCH" | |
| | **Wert**: "PROT_AREA"<br>**Bedeutung**: Systemvariablen der Schutzbereiche:<br>• $NP_PROT_NAME<br>• $NP_CHAIN_NAME<br>• $NP_CHAIN_ELEM<br>• $NP_1ST_PROT | |
| | **Wert**: "PROT_AREA_ELEM"<br>**Bedeutung**: Systemvariablen der Schutzbereichselemente von Maschinenschutzbereichen und/oder automatischen Werkzeugschutzbereichen:<br>• $NP_NAME<br>• $NP_NEXT<br>• $NP_NEXTP<br>• $NP_COLOR<br>• $NP_D_LEVEL<br>• $NP_USAGE<br>• $NP_TYPE<br>• $NP_FILENAME<br>• $NP_PARA<br>• $NP_OFF<br>• $NP_DIR<br>• $NP_ANG | |
| | **Wert**: "PROT_AREA_COLL_PAIRS"<br>**Bedeutung**: Systemvariablen der Kollisionspaare:<br>• $NP_COLL_PAIR<br>• $NP_SAFETY_DIST | |
| | **Wert**: "PROT_AREA_ALL"<br>**Bedeutung**: Alle Schutzbereiche, Schutzbereichselemente und Kollisionspaare (Systemvariablen $NP_... )<br>Gleichbedeutend mit dem sukzessiven Aufruf von DELOBJ mit "PROT_AREA", "PROT_AREA_ELEM" und "PROT_AREA_COLL_PAIRS" | |
| | **Wert**: "TRAFO_DATA"<br>**Bedeutung**: Systemvariablen aller Transformationen $NT_... | |

| | | |
|---|---|---|
| <Index1>: | Index der ersten zu löschenden Komponente (**optional**) | |
| | Datentyp: | INT |
| | Defaultwert: | -1 |
| | Wertebereich: | -1 ≤ x ≤ (maximale Anzahl projektierter Komponenten -1) |
| | **Wert** | **Bedeutung** |
| | 0, 1, 2, .... | Index der zu löschenden Komponente. |
| | -1 | Alle Komponenten des angegebenen Typs werden gelöscht. <Index2> wird nicht ausgewertet. |
| <Index2>: | Index der letzten zu löschenden Komponenten (**optional**)<br>Ist <Index2> nicht programmiert, werden nur die Systemvariablen der in <Index1> referenzierten Komponente gelöscht. | |
| | Datentyp: | INT |
| | Defaultwert: | Nur die Systemvariablen der in <Index1> referenzierten Komponente werden gelöscht. |
| | Wertebereich: | <Index1> < x ≤ (max. Anzahl projektierter Komponenten -1) |
| <NoAlarm>: | Alarmunterdrückung (**optional**) | |
| | Datentyp: | BOOL |
| | Defaultwert: | FALSE |
| | **Wert** | **Bedeutung** |
| | FALSE | Im Fehlerfall (<RetVal> < 0) wird die Programmabarbeitung angehalten und ein Alarm angezeigt. |
| | TRUE | Im Fehlerfall wird die Programmabarbeitung nicht angehalten und es wird kein Alarm angezeigt.<br>Anwendungsfall: Anwenderspezifische Reaktion entsprechend Rückgabewert |
| <RetVal>: | Rückgabewert der Funktion | |
| | Datentyp: | INT |
| | Wertebereich: | 0, -1, -2, ... -7 |
| | **Wert** | **Bedeutung** |
| | 0 | Kein Fehler aufgetreten. |
| | -1 | Aufruf der Funktion ohne Parameter. Mindestens der Parameter <CompType> muss angegeben werden. |
| | -2 | <CompType> bezeichnet unbekannte Komponente |
| | -3 | <Index1> ist kleiner als -1 |
| | -4 | <Index1> ist größer als die projektierte Anzahl Komponenten |
| | -5 | <Index1> hat beim Löschen einer Komponenten**gruppe** einen Wert ungleich -1 |
| | -6 | <Index2> ist kleiner als <Index1> |
| | -7 | <Index2> ist größer als die projektierte Anzahl Komponenten |

## 3.10.2 Indexermittlung per Namen (NAMETOINT)

In Systemvariablenfeldern vom Typ STRING sind anwenderspezifische Namen eingetragen. Anhand des Bezeichners der Systemvariablen und des Namens, ermittelt die Funktion `NAMETOINT()` den zum Namen gehörenden Indexwert, unter dem er im Systemvariablenfeld abgelegt ist.

### Syntax

```
<RetVal> = NAMETOINT(<SysVar>,<Name>[,<NoAlarm>])
```

### Bedeutung

| | | |
|---|---|---|
| `NAMETOINT`: | Ermittlung des Systemvariablenindex | |
| `<SysVar>`: | Name des Systemvariablenfeldes vom Typ STRING | |
| | Datentyp: | STRING |
| | Wertebereich: | Namen aller Systemvariablenfelder der NC vom Typ STRING |
| `<Name>`: | Zeichenkette bzw. Name, zu dem der Systemvariablenindex ermittelt werden soll. | |
| | Datentyp: | STRING |
| `<NoAlarm>`: | Alarmunterdrückung (**optional**) | |
| | Datentyp: | BOOL |
| | Defaultwert: | FALSE |
| | **Wert** | **Bedeutung** |
| | TRUE | Im Fehlerfall wird die Programmabarbeitung nicht angehalten und es wird kein Alarm angezeigt.<br>Anwendungsfall: Anwenderspezifische Reaktion entsprechend Rückgabewert |
| | FALSE | Im Fehlerfall (<RetVal> < 0) wird die Programmabarbeitung angehalten und ein Alarm angezeigt. |
| | | |
| `<RetVal>`: | Systemvariablenindex oder Fehlermeldung | |
| | Datentyp: | INT |
| | Wertebereich: | -1 ≤ x ≤ (max. Anzahl projektierter Komponenten -1) |
| | **Wert** | **Bedeutung** |
| | ≥ 0 | Der gesuchte Name wurde unter dem angegebenen Systemvariablenindex gefunden. |
| | -1 | Der gesuchte Name wurde nicht gefunden bzw. es ist ein Fehler aufgetreten. |

### Beispiel

| **Programmcode** | **Kommentar** |
|---|---|
| `DEF INT INDEX` | |
| `$NP_PROT_NAME[27] = "Abdeckung"` | |
| `...` | |
| `INDEX = NAMETOINT("$NP_PROT_NAME","Abdeckung")` | `; INDEX == 27` |

# 3.11 Kollisionsvermeidung mit kinematischen Ketten

**Hinweis**

**Schutzbereiche**

Die in den nachfolgenden Kapiteln genannten Schutzbereiche beziehen sich auf die Funktion "Geometrische Maschinenmodellierung".

**Informationen** zu dieser Funktion siehe Funktionshandbuch Überwachen und Kompensieren.

## 3.11.1 Prüfen auf Kollisionspaar (COLLPAIR)

Die Funktion `COLLPAIR(...)` ermittelt, ob zwei Schutzbereiche ein Kollisionspaar bilden.

### Syntax

```
[<RetVal> =] COLLPAIR(<Name_1>,<Name_2>[,<NoAlarm>)])
```

### Bedeutung

| | | | |
|---|---|---|---|
| `COLLPAIR`: | Prüfen auf Zugehörigkeit zu einem Kollisionspaar | | |
| `<RetVal>`: | Rückgabewert der Funktion | | |
| | Datentyp: | INT | |
| | Wert: | ≥ 0 | Die beiden Schutzbereiche bilden ein Kollisionspaar.<br>Rückgabewert == Kollisionspaarindex m (siehe $NP_COLL_PAIR) |
| | | -1 | Es wurden entweder weniger als zwei Strings angegeben, oder mindestens einer der beiden ist der Null-String. |
| | | -2 | Der im ersten Parameter angegebene Schutzbereich wurde nicht gefunden. |
| | | -3 | Der im zweiten Parameter angegebene Schutzbereich wurde nicht gefunden. |
| | | -4 | Keiner der beiden angegebenen Schutzbereiche wurde gefunden. |
| | | -5 | Beide angegebenen Schutzbereiche wurden gefunden, aber nicht gemeinsam in einem Kollisionspaar. |
| `<Name_1>`: | Name des ersten Schutzbereichs | | |
| | Datentyp: | STRING | |
| | Wertebereich: | Parametrierte Schutzbereichsnamen | |
| `<Name_2>`: | Name des zweiten Schutzbereichs | | |
| | Datentyp: | STRING | |
| | Wertebereich: | Parametrierte Schutzbereichsnamen | |

| | | | |
|---|---|---|---|
| <NoAlarm>: | Alarmunterdrückung **(optional)** | | |
| | Datentyp: | BOOL | |
| | Wert: | FALSE (Default) | Im Fehlerfall (<RetVal> < 0) wird die Programmabarbeitung angehalten und ein Alarm angezeigt. |
| | | TRUE | Im Fehlerfall wird die Programmabarbeitung nicht angehalten und es wird kein Alarm angezeigt.<br>Anwendungsfall: Anwenderspezifische Reaktion entsprechend Rückgabewert |

## 3.11.2 Neuberechnung des Maschinenmodells der Kollisionsvermeidung anfordern (PROTA)

Werden Systemvariable der kinematischen Kette $NK_..., der geometrischen Maschinenmodellierung oder der Kollisionsvermeidung $NP_... im Teileprogramm geschrieben, muss anschließend die Prozedur PROTA aufgerufen werden, damit die Änderung im NC-internen Maschinenmodell der Kollisionsvermeidung wirksam wird.

### Syntax

```
PROTA[(<Par>)]
```

### Bedeutung

| | | | |
|---|---|---|---|
| PROTA: | Neuberechnung des Maschinenmodells der Kollisionsvermeidung anfordern | | |
| | • Löst Vorlaufstopp aus.<br>• Muss alleine im Satz stehen. | | |
| <Par>: | Parameter (**optional**) | | |
| | Datentyp: | STRING | |
| | Wert: | --- | Ohne Parameter.<br>Es wird eine Neuberechnung des Maschinenmodells durchgeführt. Die Stati der Schutzbereiche bleiben erhalten. |
| | | "R" | Es wird eine Neuberechnung des Maschinenmodells durchgeführt. Die Schutzbereiche werden in ihren Initialisierungsstatus entsprechend $NP_INIT_STAT versetzt. |

### Randbedingungen

#### Simulation

Die Prozedur PROTA darf in Teileprogrammen nicht im Zusammenhang mit der Simulation (simNC) verwendet werden.

Beispiel: Vermeidung des Aufrufs von `PROTA`, während die Simulation aktiv ist.

```
Programmcode                    Kommentar
...
IF $P_SIM == FALSE              ; IF Simulation nicht aktiv
   PROTA                        ; THEN Kollisionsmodell neu berechnen
ENDIF                           ; ENDIF
...
```

### Siehe auch

Schutzbereichszustand setzen (PROTS) (Seite 737)

## 3.11.3 Schutzbereichszustand setzen (PROTS)

Die Prozedur `PROTS(...)` setzt den Status von Schutzbereichen auf den angegebenen Wert.

### Syntax

```
PROTS(<State>[,<Name_1>,...,<Name_n>])
```

### Bedeutung

| | | | |
|---|---|---|---|
| `PROTS`: | Status von Schutzbereichen setzen | | |
| | • Muss alleine im Satz stehen. | | |
| `<State>`: | Status, auf den die angegebenen Schutzbereiche gesetzt werden sollen | | |
| | Datentyp: | CHAR | |
| | Wert: | "A"oder "a" | Status: Aktiv |
| | | "I"oder "i" | Status: Inaktiv |
| | | "P"oder "p" | Status: Voraktiviert bzw. PLC-gesteuert [1] |
| | | "R"oder "r" | Status: NC-interner Wert des Initialisierungs-status [2] |
| `<Name_1> ... <Name_n>`: | Name eines oder mehrerer Schutzbereiche, die auf den angegebenen Status gesetzt werden sollen (**optional**)<br>Ist kein Name angegeben, wird der angegebenen Status für alle definierten Schutzbereiche gesetzt. | | |
| | Datentyp: | STRING | |
| | Wertebereich: | Parametrierte Schutzbereichsnamen | |
| | **Hinweis**<br>Die maximale Anzahl von Schutzbereichen, die als Parameter angegeben werden können, ist nur abhängig von der maximal möglichen Anzahl von Zeichen pro Programmzeile. | | |

[1] Die Aktivierung / Deaktivierung erfolgt über: DB10.DBX234.0 - DBX241.7

[2] Der Status wird auf den NC-internen Wert des Initialisierungsstatus gesetzt, d. h. auf den Wert, den die Systemvariable $NP_INIT_STAT zum Zeitpunkt des letzten Aufrufs von PROTA(...) (Seite 736) hatte.

## 3.11.4 Abstandsbestimmung zweier Schutzbereiche (PROTD)

Die Funktion `PROTD(...)` berechnet den Abstand von zwei Schutzbereichen.

Funktionseigenschaften:

- Die Abstandsberechnung erfolgt unabhängig vom Status der Schutzbereiche (aktiviert, deaktiviert, voraktiviert).
- Zur Abstandsberechnung zweier Schutzbereiche werden nur solche Schutzbereichselemente herangezogen, welche mit $NP_USAGE = "C" oder "A" gekennzeichnet sind. Schutzbereichselemente des Schutzbereichs, die mit $NP_USAGE = "V" gekennzeichnet sind, werden nicht betrachtet.
- Schutzbereiche, bei denen alle Schutzbereichselemente des Schutzbereichs mit $NP_USAGE = "V" gekennzeichnet sind, können nicht zur Abstandsberechnung herangezogen werden.
- Die Abstandsberechnung erfolgt mit den am Ende des Vorgängersatzes gültigen Positionen.
- Überlagerungen, die im Hauptlauf eingerechnet werden, z. B. DRF-Verschiebung oder externe Nullpunktverschiebungen, gehen mit den zum **Interpretationszeitpunkt** der Funktion gültigen Werten in die Abstandsberechnung ein.

**Hinweis**

**Synchronisation**

Bei Anwendung der Funktion `PROTD(...)` liegt es ausschließlich in der Verantwortung des Anwenders, Haupt- und Vorlauf gegebenenfalls mittels Vorlaufstopp `STOPRE` zu synchronisieren.

**Kollision**

Liegt zwischen den angegebenen Schutzbereichen eine Kollision vor, liefert die Funktion einen Abstand von 0,0. Eine Kollision liegt vor, wenn sich die beiden Schutzbereiche berühren oder durchdringen.

Der Sicherheitsabstand für die Kollisionsprüfung (MD10622 $MN_COLLISION_SAFETY_DIST) wird bei der Abstandsberechnung nicht berücksichtigt.

### Syntax

```
[<RetVal> =] PROTD([<Name_1>],[<Name_2>],VAR <Vector>[,<System>])
```

### Bedeutung

| | | |
|---|---|---|
| `PROTD`: | Abstand der beiden angegebenen Schutzbereiche berechnen | |
| | • Muss alleine im Satz stehen. | |
| `<RetVal>`: | Rückgabewert der Funktion: Absolutwert des Abstandes der beiden Schutzbereiche oder 0,0 bei Kollision (siehe oben: Absatz Kollision) | |
| | Datentyp: | REAL |
| | Wertebereich: | 0,0 ≤ x ≤ +max. REAL-Wert |

| | | | |
|---|---|---|---|
| <Name_1>, <Name_2>: | Namen der beiden Schutzbereiche, deren Abstand zueinander berechnet werden soll (**optional**) | | |
| | Datentyp: | STRING | |
| | Wertebereich: | Parametrierte Schutzbereichnamen | |
| | Defaultwert: | "" (Leerstring)<br>Sind keine Schutzbereiche angegeben, berechnet die Funktion den aktuell kleinsten Abstand aus allen im Kollisionsmodell enthaltenen aktivierten und voraktivierten Schutzbereichen. | |
| <Vector>: | Rückgabewert: 3-dimensionaler Abstandsvektor von Schutzbereich <Name_2> zu Schutzbereich <Name_1> mit:<br>• <Vector>[0]: X-Koordinate im Weltkoordinatensystem<br>• <Vector>[1]: Y-Koordinate im Weltkoordinatensystem<br>• <Vector>[2]: Z-Koordinate im Weltkoordinatensystem<br>Bei Kollision: <Vector> == Nullvektor | | |
| | Datentyp: | VAR REAL [3] | |
| | Wertebereich: | <Vector> [n]: 0,0 ≤ x ≤ ±max. REAL-Wert | |
| <System>: | Maßsystem (inch / metrisch) für Abstand und Abstandsvektor (**optional**) | | |
| | Datentyp: | BOOL | |
| | Wert: | FALSE (Default) | Maßsystem entsprechend dem aktuell aktiven G-Befehl aus G-Gruppe 13 (G70, G71, G700, G710). |
| | | TRUE | Maßsystem entsprechend des eingestellten Grundsystems:<br>MD52806 $MN_ISO_SCALING_SYSTEM |

# 3.12 Transformationen mit kinematischen Ketten

## 3.12.1 Transformation einschalten (TRAFOON)

Eine mit kinematischen Ketten definierte Transformation wird mit der vordefinierten Prozedur TRAFOON aktiviert. Der Aufruf muss alleine in einem Satz stehen.

**Hinweis**

Alternativ kann eine mit kinematischen Ketten definierte Transformation auch über konventionelle Sprachbefehle wie z. B. TRAORI oder TRANSMIT aktiviert werden. Dazu muss in der Systemvariablen $NT_TRAFO_INDEX ein entsprechender Wert ungleich null eingetragen sein.

Weitere Informationen zu $NT_TRAFO_INDEX siehe "Listenhandbuch Systemvariablen".

### Syntax

```
TRAFOON(<Trafoname>,<Diameter>,<k>)
```

### Bedeutung

| | | | |
|---|---|---|---|
| TRAFOON: | Prozedur zur Aktivierung einer mit kinematischen Ketten definierten Transformation | | |
| <Trafoname>: | Name des Transformationsdatensatzes | | |
| | Datentyp: | STRING | |
| | Wertebereich: | Alle über $NT_NAME definierte Namen von Transformationsdatensätzen | |
| | **Hinweis:**<br>Der Name des Transformationsdatensatzes muss eindeutig sein. Er darf in $NT_NAME nur einmal vorkommen. | | |
| <Diameter>: | Bezugs- bzw. Arbeitsdurchmesser (nur TRACYL) | | |
| | Datentyp: | REAL | |
| | Der Wert muss > 1 sein. | | |
| <k>: | Definiert die Verwendung der Nutwandkorrektur (nur TRACYL). | | |
| | Datentyp: | BOOL | |
| | Wert: | FALSE | ohne Nutwandkorrektur |
| | | TRUE | mit Nutwandkorrektur |
| | Entspricht dem TRACYL-Transformationstyp 514 (Nutwandkorrektur programmierbar). Ist <k> nicht angegeben, wirkt die parametrierte Einstellung von Bit 10 in $NT_CNTRL[<n>]. | | |

### Beispiel

| **Programmcode** | **Kommentar** |
|---|---|
| `TRAFOON["Trans_1"]` | `Aktiviert die Transformation mit dem Namen Trans_1.` |

## 3.12.2 Orientierungstransformation nach Maschinenvermessung modifizieren (CORRTRAFO)

Bei Maschinen mit Orientierungstransformationen, die mittels kinematischer Ketten definiert wurden, kann der Anwender die vordefinierte Funktion CORRTRAFO verwenden, um nach einer Maschinenvermessung die Offsetvektoren oder die Richtungsvektoren der Orientierungsachsen im kinematischen Modell der Maschine zu modifizieren.

**Hinweis**

Die mit der Funktion CORRTRAFO eingeschriebenen Korrekturwerte sind nicht sofort in der Transformation wirksam. Erst nach Transformationsabwahl, NEWCONF und Transformationsanwahl sind die Korrekturwerte wirksam.

### Syntax

```
<Corr_Status> = CORRTRAFO(<Corr_Vect>, <Corr_Index>, <Corr_Mode>,
[ <No_Alarm>])
```

## Bedeutung

| | |
|---|---|
| CORRTRAFO: | Funktionsaufruf |

| | | | |
|---|---|---|---|
| <Corr_Status>: | Rückgabewert der Funktion | | |
| | Datentyp: | INT | |
| | Werte: | 0 | Die Funktion wurde ohne Fehler ausgeführt. |
| | | 1 | Es ist keine Transformation aktiv. |
| | | 2 | Die aktuell aktive Transformation ist keine Orientierungstransformation. |
| | | 3 | Die aktive Orientierungstransformation wurde nicht mit kinematischen Ketten definiert. |
| | | 10 | Der Aufrufparameter <Corr_Index> ist negativ. |
| | | 11 | Der Aufrufparameter <Corr_Mode> ist negativ. |
| | | 12 | Ungültiger Verweis auf einen Abschnitt einer Teilkette (1er-Stelle von <Corr_Index>). Der Wert darf nicht größer sein als die Zahl der Orientierungsachsen in der Teilkette. |
| | | 13 | Ungültiger Verweis auf die Orientierungsachse einer Teilkette (1er-Stelle von <Corr_Index>). Der Wert muss kleiner sein als die Zahl der Orientierungsachsen in der Teilkette. |
| | | 14 | Ungültiger Verweis auf eine Teilkette (10er-Stelle von <Corr_Index>). Es sind nur die Werte 0 und 1 zulässig (Verweis auf Part- bzw. Tool-Kette). Diese Fehlernummer tritt auch auf, wenn die Teilkette, auf die <Corr_Index> verweist, nicht existiert. |
| | | 15 | Im Abschnitt, auf den mit Parameter <Corr_Index> verwiesen wird, ist kein Korrekturelement definiert ($NT_CORR_ELEM_P bzw. $NT_CORR_ELEM_T). |
| | | 20 | Ungültiger Korrekturmodus (1er-Stelle von <Corr_Mode>). Es sind nur die Werte 0 und 1 zulässig. |
| | | 21 | Ungültiger Korrekturmodus (10er- und/oder 100er-Stelle von <Corr_Mode>). Beim Schreiben einer Achsrichtung darf nur die 1er-Stelle ungleich null sein. |
| | | 30 | Die 100er-Stelle von <Corr_Mode> ist ungültig. Es sind nur die Werte 0 und 1 zulässig. |
| | | 31 | Die 1000er-Stelle von <Corr_Mode> ist ungültig. Es sind nur die Werte 0 und 1 zulässig. |
| | | 40 | Der Richtungsvektor, der als Achsrichtung übernommen werden soll, ist der Nullvektor. Dieser Fehler kann nur auftreten, wenn die 1000er-Stelle von <Corr_Mode> gleich 0 ist. Ist die 1000er-Stelle dieses Parameters gleich 1 (Überwachung der maximalen Korrektur deaktiviert), kann auch der Nullvektor geschrieben werden. |
| | | 41 | Bei der Korrektur eines Offsetvektors ist die Abweichung gegenüber dem aktuellen Werte in mindestens einer Koordinate größer als der durch das Settingdatum SD41610 $SN_CORR_TRAFO_LIN_MAX vorgegebene Maximalwert. Der Parameter <Corr_Vect> wird durch einen Fehlervektor überschrieben. Das gilt auch dann, wenn die Bearbeitung mit Alarm abgebrochen wird (siehe Parameter <No_Alarm>).<br>Der Fehlervektor enthält in den Komponenten, deren Korrekturwert das zulässige Limit überschritten hat, die vorzeichenrichtige Differenz zwischen dem ermittelten Korrekturwert und dem Limit. |

| | | | |
|---|---|---|---|
| | | | Der Inhalt der Komponenten, die ihr Limit nicht überschritten haben, ist null. |
| | | 42 | Bei der Korrektur eines Richtungsvektors ist die Winkelabweichung gegenüber der aktuellen Richtung größer als der durch das Settingdatum SD41611 $SN_CORR_TRAFO_DIR_MAX vorgegebene Maximalwert. |
| | | 43 | Der Versuch, eine Systemvariable zu beschreiben, wurde wegen fehlender Schreibrechte abgewiesen. |
| <Corr_Vect>: | Korrekturvektor<br>Der Inhalt des Korrekturvektors wird durch die nachfolgenden Parameter **<**Corr_Index**>** und <Corr_Mode> definiert.<br>Ist <Corr_Status> = 41, wird der Inhalt des Vektors überschrieben (s. o.). | | |
| | Datentyp: | REAL | |
| **<**Corr_Index**>**: | Abschnitt, dessen Korrekturelement modifiziert werden soll /<br>Index der Orientierungsachse, deren Richtungsvektor modifiziert werden soll | | |
| | Datentyp: | INT | |
| | Der Parameter <Corr_Index> ist dezimalcodiert (1er- bis 10er-Stelle): | | |
| | 1er-Stelle: | Enthält den Index des Abschnitts bzw. der Orientierungsachse in der Teilkette. | |
| | 10er-Stelle: | Verweist auf die Teilkette. | |
| | | **0**x | Werkstück-Kette |
| | | **1**x | Werkzeug-Kette |

| | | | |
|---|---|---|---|
| <Corr_Mode>: | Korrekturmodus | | |
| | Datentyp: | INT | |
| | Der Parameter <Corr_Mode> ist dezimalcodiert (1er- bis 1000er-Stelle): | | |
| | 1er-Stelle: | Bestimmt, welches Element korrigiert werden soll. | |
| | | xxx**0** | Korrektur eines linearen Offsetvektors |
| | | xxx**1** | Korrektur des Richtungsvektors einer Orientierungsachse |
| | 10er-Stelle: | Bestimmt, wie das Korrekturelement, auf das der Inhalt von <Corr_Index> verweist, modifiziert werden soll. | |
| | | xx**0**x | Der Korrekturvektor wird unmittelbar in das Korrekturelement geschrieben.<br>Diese Variante kann dazu verwendet werden, das Korrekturelement unmittelbar zu beschreiben, ohne dass der Index <n> des betreffenden Systemdatums ($NK_OFF_DIR[<n>, ...]) bekannt sein muss. |
| | | xx**1**x | Wie 0, jedoch mit dem Unterschied, dass der übergebene Korrekturwert in Weltkoordinaten interpretiert wird.<br>Ein Unterschied zwischen den Varianten 0 und 1 kann immer dann entstehen, wenn die kinematische Kette in Grundstellung (Positionen aller Orientierungsachsen gleich 0) weitere Drehungen enthält. |
| | | xx**2**x | Wie 1, jedoch mit dem Unterschied, dass sich der Korrekturwert auf den gesamten Abschnitt bezieht, d. h., in das Korrekturelement wird ein solcher Wert eingetragen, dass der gesamte Abschnitt die durch den Korrekturwert definierte Länge erreicht. |
| | | **Hinweis:**<br>Die Werte 1 und 2 sind beim Beschreiben des Richtungsvektors einer Orientierungsachse nicht erlaubt. | |
| | 100er-Stelle: | Bestimmt, wie der Inhalt des Parameters <Corr_Vect> zu interpretieren ist. | |
| | | x**0**xx | Der übergebene Korrekturvektor <Corr_Vect> enthält die gesamte neue Länge des Korrekturelements bzw. des Abschnitts, auf den <Corr_Index> in Verbindung mit der 10er-Stelle von <Corr_Mode> verweist (absolute Korrektur). |
| | | x**1**xx | Der übergebene Korrekturvektor <Corr_Vect> enthält nur die Differenz gegenüber der aktuellen Länge des Korrekturelements bzw. des Abschnitts, auf den <Corr_Index> in Verbindung mit der 10er-Stelle von <Corr_Mode> verweist (inkrementelle Korrektur). |
| | | **Hinweis:**<br>Bei der Korrektur des Richtungsvektors einer Orientierungsachse muss der Inhalt der 100er-Stelle 0 sein. | |
| | 1000er-Stelle: | Bestimmt, ob die Korrektur durch folgenden Maximalwert begrenzt werden soll:<br>• SD41610 $SN_CORR_TRAFO_LIN_MAX<br>bzw.<br>• SD41611 $SN_CORR_TRAFO_DIR_MAX | |
| | | **0**xxx | Überwachung der maximalen Korrektur ist aktiv. |
| | | **1**xxx | Überwachung der maximalen Korrektur ist nicht aktiv. |

| | | | |
|---|---|---|---|
| <No_Alarm>: | Verhalten im Fehlerfall (Rückgabewert > 0) (**optional**) | | |
| | Datentyp: | BOOL | |
| | Wert: | FALSE (Default) | Im Fehlerfall wird die Programmabarbeitung angehalten und Alarm 14103 angezeigt. |
| | | TRUE | Im Fehlerfall wird die Programmabarbeitung nicht angehalten und es wird kein Alarm angezeigt.<br>Anwendungsfall: Anwenderspezifische Reaktion entsprechend Rückgabewert |

**Hinweis**

Falls beim Aufruf der Funktion ein Fehler auftritt, wird entweder ein Alarm ausgegeben oder eine Fehlernummer zurückgegeben (siehe Parameter <No_Alarm>), sodass der Anwender selbst geeignet auf den Fehlerzustand reagieren kann. Die Fehlerursache wird durch einen Alarmparameter näher bezeichnet. Eine an Stelle eines Alarms zurückgegebene Fehlernummer ist identisch mit dem Alarmparameter.

## Weitere Informationen zu CORRTRAFO

Die kinematische Struktur einer Maschine mit Orientierungstransformation wird durch eine oder zwei kinematische Ketten (Teilketten) beschrieben, die vom Nullpunkt des Weltkoordinatensystems ausgehen. Eine der beiden Ketten, die **Werkzeug-Kette**, endet am Bezugspunkt des Werkzeugs, die andere, die **Werkstück-Kette** im Nullpunkt des Basiskoordinatensystems.

Die Funktion CORRTRAFO schreibt Hebelarmlängen und Achsrichtungen bei Maschinen mit Orientierungstransformation in spezielle Korrekturelemente. Eine kinematische Kette wird unter anderem mit Elementen vom Typ OFFSET beschrieben, die über $NK_TYPE definiert werden.

### CORRTRAFO arbeitet mit Abschnitten

Die beiden Teilketten können sich jeweils in maximal vier Abschnitte aufteilen:

- Abschnitt 1 beginnt am Startpunkt der Kette und endet an der ersten Orientierungsachse.
- Abschnitt 2 ist der Abschnitt zwischen Orientierungsachse 1 und Orientierungsachse 2.
- Abschnitt 3 ist der Abschnitt zwischen Orientierungsachse 2 und Orientierungsachse 3.
- Abschnitt 4 ist der Abschnitt zwischen Orientierungsachse 3 und dem Ende der Werkzeug- bzw. Werkstück-Kette.

Die Abschnitte können jeweils konstante Kettenelemente des Typs OFFSET oder ROT_CONST enthalten.

Die folgende Grafik zeigt eine Orientierungstransformation mit 2 Orientierungsachsen.

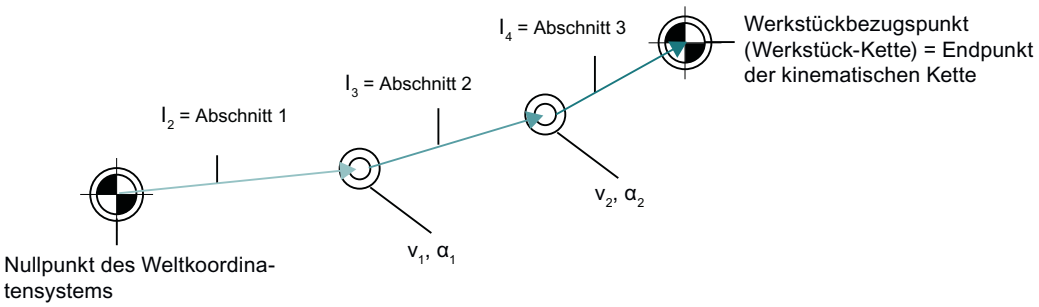



Bild 3-7 CORRTRAFO Beispiel

Die Abschnitte sind eindeutig definiert: Durchläuft man die kinematische Teil-Kette von ihrem Startpunkt bis zum Endpunkt, so hat der erste Abschnitt den Index 0, der nächste den Index 1 usw. Der Index des letzten Abschnitts ist dann immer gleich der Zahl der Orientierungsachsen.

### Korrekturelemente

Auf jeweils ein konstantes kinematisches Kettenelement (Kettenelement des Typs $NK_TYPE[<n>] = "OFFSET") in jedem dieser Abschnitte kann mit der Systemvariablen $NT_CORR_ELEM_T[<n>, 0 ... 3] bzw. $NT_CORR_ELEM_P[<n>, 0 ... 3] verwiesen werden. In diese so bezeichneten Elemente werden mithilfe der Funktion CORRTRAFO die Korrekturwerte geschrieben, die bei der Maschinenvermessung ermittelt wurden.

### Beispiel mit Transformations-Index = 1:

- $NT_CORR_ELEM_T[1,0] = "C_AXIS_OFFSET"; Offset der C-Achse (Orientierungsachse 1) in Abschnitt 1 ist als Korrekturelement definiert.
- $NT_CORR_ELEM_T[1,1] = "B_AXIS_OFFSET"; Offset der B-Achse (Orientierungsachse 2) in Abschnitt 2 ist als Korrekturelement definiert.
- $NT_CORR_ELEM_T[1,2] = "BASE_TOOL_OFFSET"; Offset von der B-Achse zum Werkzeugbezugspunkt in Abschnitt 3 ist als Korrekturelement definiert.

Die Reihenfolge der Verweise in $NT_CORR_ELEM_T/P[<n>, 0 ... 3] muss mit den oben beschriebenen Abschnitten korrespondieren, d. h. in $NT_CORR_ELEM_T/P [<n>, 0] kann nur ein Kettenelement stehen, das vor der ersten Orientierungsachse liegt usw..

Die Funktion CORRTRAFO schreibt in die so definierten Korrekturelemente die Werte, die über die Vermessung der Maschine bestimmt wurden. Die Modifizierung der Korrekturwerte wird in CORRTRAFO über den Parameter <Corr_Mode> definiert.

### Kette schließen

Sind Bit 7 oder Bit 8 in der Systemvariablen $NT_CNTRL[<n>] gesetzt, werden am Ende der Werkstück-Kette (Bit 7) bzw. vor dem Startpunkt der Werkzeug-Kette (Bit 8) intern automatisch zusätzliche konstante Kettenelemente eingefügt, die eine Verbindung vom Endpunkt der Kette zum Maschinennullpunkt herstellen ("Kette schließen").

Diese automatisch eingefügten Elemente können von extern nicht beschrieben, sondern nur gelesen werden (siehe dazu die Systemvariablen $AC_TRAFO_CORR_ELEM_P/T).

### Punkt zum Schließen der Werkzeug-Kette

Ist die Systemvariable $NT_CLOSE_CHAIN_T nicht leer, wird die Werkzeug-Kette nicht am Endpunkt der Kette, sondern am Endpunkt des bezeichneten Kettenelements geschlossen. Weitere Kettenelemente, die hinter diesem Punkt liegen, führen dann beim Aktivieren der Transformation zu einer entsprechenden Nullpunktverschiebung.

### Index einer Orientierungsachse

Außer den konstanten Offsets zwischen den Orientierungsachsen können mit der Funktion CORRTRAFO auch die Richtungsvektoren der Orientierungsachsen beschrieben werden. Unter dem Index einer Orientierungsachse wird hier der Index verstanden, der sich ergibt, wenn man die kinematische Teilkette vom Ursprung bis zu ihrem Ende durchläuft, wobei die Zählung bei null beginnt. Der Index einer Orientierungsachse ist deshalb immer gleich dem Index des davorliegenden Abschnitts.

Der Index einer Orientierungsachse kann auch mit der Systemvariablen $AC_TRAFO_ORIAX_LOC ermittelt werden.

### Maximal zulässige Veränderung eines Kettenelements

Die maximal zulässige Veränderung eines Kettenelements kann durch die beiden Settingdaten SD41610 $SN_CORR_TRAFO_LIN_MAX für Offsetvektoren und SD41611 $SN_CORR_TRAFO_DIR_MAX für die Richtungsvektoren von Orientierungsachsen begrenzt werden. SD41610 $SN_CORR_TRAFO_LIN_MAX gibt an, um welchen Betrag jede einzelne Vektorkomponente gegenüber ihrem Bezugswert maximal verändert werden darf. SD41611 $SN_CORR_TRAFO_DIR_MAX gibt an, um welchen Winkel sich die Richtung des Achsvektors maximal gegenüber ihrem Bezugswert verändern darf. Der Bezugswert ist dabei immer der entsprechende Wert, der in der beim Aufruf von CORRTRAFO aktiven Transformation wirksam ist. D. h., dass eventuell nach der Aktivierung der Transformation veränderte Inhalte der Kinematikdaten in der Datenhaltung keinen Einfluss auf die Wirkungsweise der Funktion CORRTRAFO haben.

# 3.13 Werkzeugkorrekturen

## 3.13.1 Korrekturspeicher

### Aufbau des Korrekturspeichers

Jedes Datenfeld ist mit einer T- und D-Nummer aufrufbar und enthält neben den geometrischen Angaben für das Werkzeug noch weitere Einträge, z. B. den Werkzeugtyp.

### Anwender-Schneidendaten

Über Maschinendatum können Anwender-Schneidendaten konfiguriert werden. Bitte beachten Sie die Angaben des Maschinenherstellers.

## Werkzeugparameter

**Hinweis**

**Einzelne Werte im Korrekturspeicher**

Die einzelnen Werte des Korrekturspeichers P1 bis P25 sind über Systemvariable vom Programm les- und schreibbar. Alle übrigen Parameter sind reserviert.

Die Werkzeugparameter $TC_DP6 bis $TC_DP8, $TC_DP10 und $TC_DP11 sowie $TC_DP15 bis $TC_DP17, $TC_DP19 und $TC_DP20 haben abhängig vom Werkzeugtyp eine andere Bedeutung.

| Werkzeugparameter Nummer (DP) | Bedeutung der Systemvariablen | Bemerkung |
|---|---|---|
| $TC_DP1 | Werkzeugtyp | Übersicht siehe Liste |
| $TC_DP2 | Schneidenlage | nur für Drehwerkzeuge |
| **Geometrie** | **Längenkorrektur** | |
| $TC_DP3 | Länge 1 | Verrechnung nach |
| $TC_DP4 | Länge 2 | Typ und Ebene |
| $TC_DP5 | Länge 3 | |
| **Geometrie** | **Radius** | |
| $TC_DP6 [1)]<br>$TC_DP6 [2)] | Radius 1 / Länge 1<br>Durchmesser d | Fräs-/Dreh-/Schleifwerkz.<br>Nutsäge |
| $TC_DP7 [1)]<br>$TC_DP7 [2)] | Länge 2 / Eckenradius kegelige Fräser<br>Nutbreite b Eckenradius | Fräswerkzeuge<br>Nutsäge |
| $TC_DP8 [1)]<br>$TC_DP8 [2)] | Verrundungsradius 1 für Fräswerkzeuge<br>Überstand k | Fräswerkzeuge<br>Nutsäge |
| $TC_DP9 [1) 3)] | Verrundungsradius 2 | reserviert |
| $TC_DP10 [1)] | Winkel 1 Stirnseite des Werkzeugs | kegelige Fräswerkzeuge |
| $TC_DP11 [1)] | Winkel 2 Werkzeug-Längsachse | kegelige Fräswerkzeuge |
| **Verschleiß** | **Längen- und Radiuskorrektur** | |

| Werkzeugparameter Nummer (DP) | Bedeutung der Systemvariablen | Bemerkung |
|---|---|---|
| $TC_DP12 | Länge 1 | |
| $TC_DP13 | Länge 2 | |
| $TC_DP14 | Länge 3 | |
| $TC_DP15 [1]<br>$TC_DP15 [2] | Radius 1 / Länge 1<br>Durchmesser d | Fräs-/Dreh-/Schleifwerkz.<br>Nutsäge |
| $TC_DP16 [1]<br>$TC_DP16 [3] | Länge 2 / Eckenradius kegelige Fräser Nutbreite b<br>Eckenradius | Fräswerkzeuge<br>Nutsäge |
| $TC_DP17 [1]<br>$TC_DP17 [2] | Verrundungsradius 1 für Fräswerkzeuge<br>Überstand k | Fräsen / 3D Stirnfräsen<br>Nutsäge |
| $TC_DP18 [1] [3] | Verrundungsradius 2 | reserviert |
| $TC_DP19[1] | Winkel 1 Stirnseite des Werkzeugs | kegelige Fräswerkzeuge |
| $TC_DP20[1] | Winkel 2 Werkzeug-Längsachse | kegelige Fräswerkzeuge |
| **Basismaß/Adapter** | **Längenkorrekturen** | |
| $TC_DP21 | Länge 1 | |
| $TC_DP22 | Länge 2 | |
| $TC_DP23 | Länge 3 | |
| **Technologie** | | |
| $TC_DP24 | Freiwinkel | nur für Drehwerkzeuge |
| $TC_DP25 | | reserviert |

[1] Gilt auch bei Fräswerkzeugen für das 3D-Stirnfräsen

[2] Bei Nutsäge Werkzeugtyp

[3] reserviert

**Anmerkungen**

Für die geometrischen Größen (z. B. Länge 1 oder Radius) bestehen mehrere Eintragskomponenten. Diese werden zu einer resultierenden Größe additiv verrechnet (z. B. Gesamtlänge 1, Gesamtradius), die dann zur Wirkung kommt.

Nicht benötigte Korrekturen sind mit dem Wert Null zu belegen.

## Werkzeugparameter $TC-DP1 bis $TC-DP23 mit Konturwerkzeugen

**Hinweis**

Die Werkzeugparameter, die in der Tabelle nicht aufgeführt sind wie z. B. $TC_DP7, werden nicht ausgewertet, d. h. ihr Inhalt ist bedeutungslos.

| Werkzeugparameter Nummer (DP) | Bedeutung | Schneiden Dn | | Bemerkung |
|---|---|---|---|---|
| $TC_DP1 | Werkzeugtyp | | | 400 bis 599 |
| $TC_DP2 | Schneidenlage | | | |
| **Geometrie** | **Längenkorrektur** | | | |

| Werkzeugparameter Nummer (DP) | Bedeutung | Schneiden Dn | | Bemerkung |
|---|---|---|---|---|
| $TC_DP3 | Länge 1 | | | |
| $TC_DP4 | Länge 2 | | | |
| $TC_DP5 | Länge 3 | | | |
| **Geometrie** | **Radius** | | | |
| $TC_DP6 | Radius | | | |
| **Geometrie** | **Grenzwinkel** | | | |
| $TC_DP10 | minimaler Grenzwinkel | | | |
| $TC_DP11 | maximaler Grenzwinkel | | | |
| **Verschleiß** | **Längen- und Radiuskorrektur** | | | |
| $TC_DP12 | Verschleiß Länge 1 | | | |
| $TC_DP13 | Verschleiß Länge 2 | | | |
| $TC_DP14 | Verschleiß Länge 3 | | | |
| $TC_DP15 | Verschleiß Radius | | | |
| **Verschleiß** | **Grenzwinkel** | | | |
| $TC_DP19 | Verschleiß min. Grenzwinkel | | | |
| $TC_DP20 | Verschleiß max. Grenzwinkel | | | |
| **Basismaß/Adapter** | **Längenkorrekturen** | | | |
| $TC_DP21 | Länge 1 | | | |
| $TC_DP22 | Länge 2 | | | |
| $TC_DP23 | Länge 3 | | | |

#### Grundwert und Verschleißwert

Die resultierenden Größen ergeben sich jeweils als Summe aus Grundwert und Verschleißwert (z. B. $TC_DP6 + $TC_DP15 für den Radius). Zur Werkzeuglänge der ersten Schneide wird außerdem noch das Basismaß ($TC_DP21 – $TC_DP23) addiert. Zusätzlich wirken auf diese Werkzeuglänge alle anderen Größen, die auch bei einem herkömmlichen Werkzeug die effektive Werkzeuglänge beeinflussen können (Adapter, orientierbarer Werkzeugträger, Settingdaten).

#### Grenzwinkel 1 und 2

Die Grenzwinkel 1 bzw. 2 beziehen sich jeweils auf den Vektor vom Schneidenmittelpunkt zum Schneidenbezugspunkt und werden im Gegenuhrzeigersinn gezählt.

## 3.13.2 Additive Korrekturen

### 3.13.2.1 Additive Korrekturen anwählen (DL)

Additive Korrekturen können als in der Bearbeitung programmierbare Prozesskorrekturen betrachtet werden. Sie beziehen sich auf die geometrischen Daten einer Schneide und sind somit Bestandteil der Werkzeugschneidendaten.

Die Daten einer additiven Korrektur werden über eine DL-Nummer angesprochen (DL: Location dependent; Korrekturen bezüglich des jeweiligen Einsatzorts) und über die Bedienoberfläche eingegeben.

## Anwendung

Durch additive Korrekturen können einsatzortbedingte Maßfehler ausgeglichen werden.

## Syntax

```
DL=<Nummer>
```

## Bedeutung

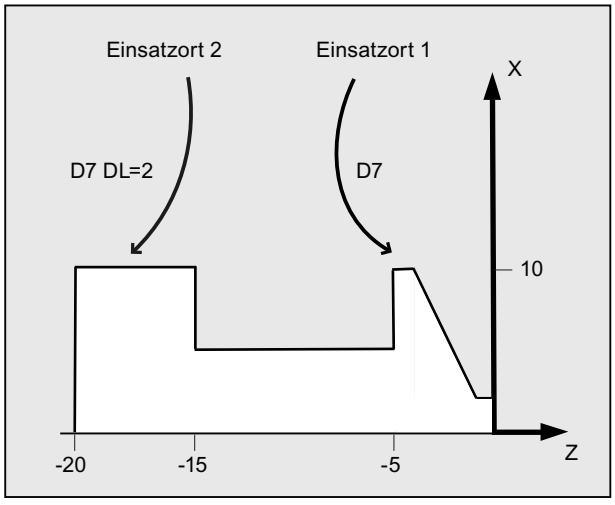
| | |
|---|---|
| DL: | Befehl zur Aktivierung einer additiven Korrektur |
| <Nummer>: | Über den Parameter <Nummer> wird der zu aktivierende additive Werkzeugkorrekturdatensatz angegeben |

### Hinweis

Die Festlegung von Anzahl und Aktivierung der additiven Korrekturen erfolgt über Maschinendaten (→ Angaben des Maschinenherstellers beachten!).

## Beispiel

Die gleiche Schneide wird für 2 Lagersitze verwendet:




| **Programmcode** | **Kommentar** |
|---|---|
| N110 T7 D7 | ; Der Revolver wird auf Platz 7 positioniert. D7 und DL=1 werden aktiviert und im nächsten Satz herausgefahren. |
| N120 G0 X10 Z1 | |
| N130 G1 Z-6 | |

| **Programmcode** | **Kommentar** |
|---|---|
| N140 G0 DL=2 Z-14 | ; Additiv zu D7 wird DL=2 aktiviert und im nächsten Satz herausgefahren. |
| N150 G1 Z-21 | |
| N160 G0 X200 Z200 | ; Werkzeugwechselpunkt anfahren. |
| ... | |

### 3.13.2.2 Verschleiß- und Einrichtewerte festlegen ($TC_SCPxy[t,d], $TC_ECPxy[t,d])

Verschleiß- und Einrichtewerte können über Systemvariablen gelesen und geschrieben werden. Dabei orientiert sich die Logik an der Logik der entsprechenden Systemvariablen für Werkzeuge und Schneiden.

## Systemvariablen

| | |
|---|---|
| $TC_SCPxy[<t>,<d>]: | Verschleißwerte, die über xy dem jeweiligen Geometrieparameter zugeordnet sind, wobei x die Nummer des Verschleißwerts entspricht und y den Bezug zum Geometrieparameter herstellt. |
| $TC_ECPxy[<t>,<d>]: | Einrichtewerte, die über xy dem jeweiligen Geometrieparameter zugeordnet sind, wobei x die Nummer des Einrichtewerts entspricht und y den Bezug zum Geometrieparameter herstellt. |
| <t>: T-Nummer des Werkzeugs<br><d>: D-Nummer der Schneide des Werkzeugs | |

**Hinweis**

Die festgelegten Verschleiß- und Einrichtewerte werden zu den Geometrieparametern und den übrigen Korrekturparametern (D-Nummer) addiert.

## Beispiel

Der Verschleißwert der Länge 1 wird für die Schneide <d> des Werkzeugs <t> auf den Wert 1.0 festgelegt.

Parameter: $TC_DP**3** (Länge 1, bei Drehwerkzeugen)

Verschleißwerte: $TC_SCP1**3** bis $TC_SCP6**3**

Einrichtewerte: $TC_ECP1**3** bis $TC_ECP6**3**

$TC_SCP4**3** [<t>,<d>] = 1.0

### 3.13.2.3 Additive Korrekturen löschen (DELDL)

Mit dem Befehl `DELDL` werden additive Korrekturen für die Schneide eines Werkzeugs gelöscht (Freigabe von Speicher). Dabei werden sowohl die festgelegten Verschleißwerte als auch die Einrichtewerte gelöscht.

### Syntax

```
DELDL[<t>,<d>]
DELDL[<t>]
DELDL
<Status>=DELDL[<t>,<d>]
```

### Bedeutung

| | | |
|---|---|---|
| DELDL: | Befehl zum Löschen additiver Korrekturen | |
| <t>: | T-Nummer des Werkzeugs | |
| <d>: | D-Nummer der Schneide des Werkzeugs | |
| DELDL[<t>,<d>]: | Es werden alle additiven Korrekturen der Schneide <d> des Werkzeugs <t> gelöscht. | |
| DELDL[<t>]: | Es werden alle additiven Korrekturen aller Schneiden des Werkzeugs <t> gelöscht. | |
| DELDL: | Es werden alle additiven Korrekturen aller Schneiden aller Werkzeuge der TO-Einheit gelöscht (für den Kanal, in dem der Befehl programmiert wird). | |
| <Status>: | Lösch-Status | |
| | Wert: | Bedeutung: |
| | 0 | Das Löschen wurde erfolgreich durchgeführt. |
| | - | Das Löschen wurde nicht durchgeführt (wenn die Parametrierung genau eine Schneide bezeichnet), oder das Löschen erfolgte nicht vollständig (wenn die Parametrierung mehrere Schneiden bezeichnet). |

**Hinweis**

Verschleiß- und Einrichtewerte aktiver Werkzeuge können nicht gelöscht werden (verhält sich analog zum Löschverhalten von D bzw. Werkzeugdaten).

## 3.13.3 Werkzeugkorrektur - Sonderbehandlung

Mit den Settingdaten SD42900 bis SD42960 lässt sich die Bewertung der Vorzeichen für Werkzeuglänge und Verschleiß steuern.

Das gilt ebenfalls für das Verhalten der Verschleißkomponenten beim Spiegeln von Geometrieachsen oder beim Wechsel der Bearbeitungsebene und auch zur Temperaturkompensation in Werkzeugrichtung.

### Verschleißwerte

Wenn im Folgenden auf Verschleißwerte Bezug genommen wird, ist darunter jeweils die Summe aus den eigentlichen Verschleißwerten ($TC_DP12 bis $TC_DP20) und den Summenkorrekturen mit den Verschleißwerten ($SCPX3 bis $SCPX11) und Einrichtewerten ($ECPX3 bis $ECPX11) zu verstehen.

**Informationen** zu den Summenkorrekturen siehe Funktionshandbuch Werkzeugverwaltung.

## Settingdaten

| | |
|---|---|
| SD42900 $SC_MIRROR_TOOL_LENGTH | Spiegeln von Werkzeuglängenkomponenten und Komponenten des Basismaßes. |
| SD42910 $SC_MIRROR_TOOL_WEAR | Spiegeln von Verschleißwerten der Werkzeuglängenkomponenten. |
| SD42920 $SC_WEAR_SIGN_CUTPOS | Vorzeichenbewertung der Verschleißkomponenten in Abhängigkeit von der Schneidenlage. |
| SD42930 $SC_WEAR_SIGN | Invertiert die Vorzeichen der Verschleißmaße. |
| SD42935 $SC_WEAR_TRANSFORM | Transformation der Verschleißwerte. |
| SD42940 $SC_TOOL_LENGTH_CONST | Zuordnung der Werkzeuglängenkomponenten zu den Geometrieachsen. |
| SD42950 $SC_TOOL_LENGTH_TYPE | Zuordnung der Werkzeuglängenkomponenten unabhängig vom Werkzeugtyp. |
| SD42960 $SC_TOOL_TEMP_COMP | Temperaturkompensationswert in Werkzeugrichtung. Ist auch bei vorhandener Werkzeugorientierung wirksam. |

## Weitere Informationen

### Wirksamwerden der veränderten Settingdaten

Die Neubewertung von Werkzeugkomponenten bei einer Änderung der beschriebenen Settingdaten wird erst wirksam, wenn das nächste Mal eine Werkzeugschneide angewählt wird. Ist ein Werkzeug bereits aktiv und die Bewertung der Daten dieses Werkzeugs soll verändert wirksam werden, muss dieses Werkzeug erneut angewählt werden.

Entsprechendes gilt für den Fall, dass sich die resultierende Werkzeuglänge ändert, weil der Spiegelungszustand einer Achse geändert wurde. Das Werkzeug muss nach dem Spiegelbefehl erneut angewählt werden, damit die geänderten Werkzeuglängenkomponenten wirksam werden.

### Orientierbare Werkzeugträger und neue Settingdaten

Die Settingdaten SD42900 bis SD42940 wirken nicht auf die Komponenten eines eventuell aktiven orientierbaren Werkzeugträgers. Ein Werkzeug geht jedoch immer mit seiner gesamten resultierenden Länge (Werkzeuglänge + Verschleiß + Basismaß) in die Berechnung mit einem orientierbaren Werkzeugträger ein. Bei der Berechnung der resultierenden Gesamtlänge werden alle Änderungen berücksichtigt, die durch die Settingdaten verursacht wurden; d.h. Vektoren des orientierbaren Werkzeugträgers sind unabhängig von der Bearbeitungsebene.

### Hinweis

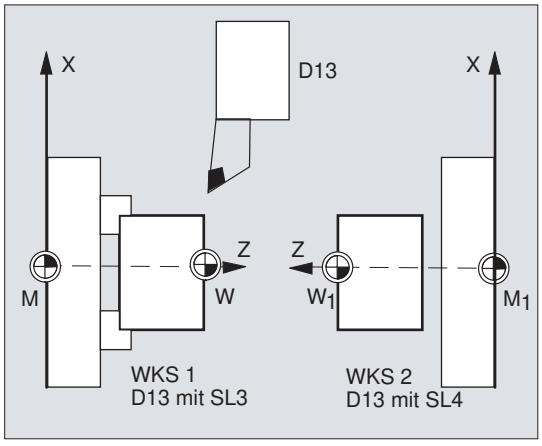
Häufig wird es beim Einsatz orientierbarer Werkzeugträger sinnvoll sein, alle Werkzeuge für ein nicht gespiegeltes Grundsystem zu definieren, auch diejenigen, die nur bei Spiegelbearbeitung verwendet werden. Bei Bearbeitung mit gespiegelten Achsen wird dann der Werkzeugträger so gedreht, dass die tatsächliche Lage des Werkzeugs richtig beschrieben wird. Alle Werkzeuglängenkomponenten wirken dann automatisch in der richtigen Richtung, so dass sich eine Steuerung der Bewertung einzelner Komponenten über Settingdaten abhängig vom Spiegelungszustand einzelner Achsen erübrigt.

**Weitere Anwendungsmöglichkeiten**

Die Verwendung der Funktionalität orientierbarer Werkzeugträger kann auch dann sinnvoll sein, wenn an der Maschine physikalisch keine Möglichkeit vorgesehen ist, Werkzeuge zu drehen, Werkzeuge aber mit verschiedenen Orientierungen fest installiert sind. Die Werkzeugvermaßung kann dann einheitlich in einer Grundorientierung vorgenommen werden, und die für die Bearbeitung relevanten Maße ergeben sich durch Drehungen eines virtuellen Werkzeugträgers.

### 3.13.3.1 Werkzeuglängen spiegeln

Mit gesetzten Settingdaten SD42900 $SC_MIRROR_TOOL_LENGTH und SD42910 $SC_MIRROR_TOOL_WEAR ungleich Null können Sie Werkzeuglängenkomponenten und Komponenten der Basismaße mit Verschleißwerten deren zugehörigen Achsen spiegeln.




**SD42900 $SC_MIRROR_TOOL_LENGTH**

Settingdatum **ungleich** Null:

Es werden die Werkzeuglängenkomponenten ($TC_DP3, $TC_DP4 und $TC_DP5) und die Komponenten der Basismaße ($TC_DP21, $TC_DP22 und $TC_DP23), deren zugehörige Achsen gespiegelt sind, ebenfalls gespiegelt - durch Vorzeicheninvertierung.

Die Verschleißwerte werden **nicht** mitgespiegelt. Sollen diese ebenfalls gespiegelt werden, muss das Settingdatum SD42910 $SC_MIRROR_TOOL_WEAR gesetzt sein.

**SD42910 $SC_MIRROR_TOOL_WEAR**

Settingdatum **ungleich** Null:

Es werden die Verschleißwerte der Werkzeuglängenkomponenten, deren zugehörige Achsen gespiegelt sind, ebenfalls gespiegelt - durch Vorzeicheninvertierung.

### 3.13.3.2 Vorzeichenbewertung Verschleiß

Mit gesetzten Settingdaten SD42920 $SC_WEAR_SIGN_CUTPOS und SD42930 $SC_WEAR_SIGN ungleich Null können Sie die Vorzeichenbewertung der Verschleißkomponenten invertieren.

**SD42920 $SC_WEAR_SIGN_CUTPOS**

Settingdatum **ungleich** Null:

Bei Werkzeugen mit relevanter Schneidenlage (Dreh- und Schleifwerkzeuge, Werkzeugtypen 400) hängt die Vorzeichenbewertung der Verschleißkomponenten in der Bearbeitungsebene von der Schneidenlage ab. Bei Werkzeugtypen ohne relevanter Schneidenlage ist dieses Settingdatum bedeutungslos.

In folgender Tabelle sind die Maße durch ein X gekennzeichnet, deren Vorzeichen über das SD42920 (ungleich 0) invertiert wird:

| Schneidenlage | Länge 1 | Länge 2 |
|---|---|---|
| 1 | | |
| 2 | | X |
| 3 | X | X |
| 4 | X | |
| 5 | | |
| 6 | | |
| 7 | | X |
| 8 | X | |
| 9 | | |

**Hinweis**

Die Vorzeichenbewertung durch SD42920 und SD42910 sind voneinander unabhängig. Wenn z. B. das Vorzeichen einer Maßangabe durch beide Settingdaten geändert wird, bleibt das resultierende Vorzeichen unverändert.

**SD42930 $SC_WEAR_SIGN**

Settingdatum **ungleich** Null:

Das Vorzeichen aller Verschleißmaße wird invertiert. Es wirkt sowohl auf die Werkzeuglänge als auch auf die übrigen Größen wie Werkzeugradius, Verrundungsradius usw.

Wird ein positives Verschleißmaß eingegeben, wird somit das Werkzeug "kürzer" und "dünner", siehe Kapitel "Werkzeugkorrektur, Sonderbehandlung", Wirksamwerden der veränderten Settingdaten".

### 3.13.3.3 Koordinatensystem der aktiven Bearbeitung (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)

Abhängig von der Kinematik der Maschine oder vom Vorhandensein eines orientierbaren Werkzeugträgers werden die in einem dieser Koordinatensysteme gemessenen Verschleißwerte in ein geeignetes Koordinatensystem überführt bzw. transformiert.

### Koordinatensysteme der aktiven Bearbeitung

Aus den folgenden Koordinatensystemen können Offsets der Werkzeuglänge hervorgehen, welche die Werkzeuglängenkomponente Verschleiß über den entsprechenden G-Befehl der Gruppe 56 in ein aktives Werkzeug einrechnet:

- Maschinenkoordinatensystem (MKS)
- Basiskoordinatensystem (BKS)
- Werkstückkoordinatensystem (WKS)
- Werkzeugkoordinatensystem (TCS)
- Werkzeugkoordinatensystem der kinematischen Transformation (KCS)

## Syntax

```
TOWSTD
TOWMCS
TOWWCS
TOWBCS
TOWTCS
TOWKCS
```

## Bedeutung

| | |
|---|---|
| TOWSTD: | Grundstellungswert für Korrekturen in der Werkzeuglänge Verschleißwert |
| TOWMCS: | Korrekturen in der Werkzeuglänge im MKS |
| TOWWCS: | Korrekturen in der Werkzeuglänge im WKS |
| TOWBCS: | Korrekturen in der Werkzeuglänge im BKS |
| TOWTCS: | Korrekturen der Werkzeuglänge am Werkzeugträgerbezugspunkt (orientierbarer Werkzeugträger) |
| TOWKCS: | Korrekturen der Werkzeuglänge des Werkzeugkopfs (kinematischer Transformation) |

## Weitere Informationen

### Unterscheidungsmerkmale

In der folgenden Tabelle sind die wichtigsten Unterscheidungsmerkmale dargestellt:

| G-Befehl | Verschleißwert | Aktiver orientierbarer Werkzeugträger |
|---|---|---|
| TOWSTD | Grundstellungswert, Werkzeuglänge | Verschleißwerte unterliegen der Drehung. |
| TOWMCS | Verschleißwert Im MKS. TOWMCS ist mit TOWSTD identisch, wenn kein orientierbarer Werkzeugträger aktiv ist. | Es dreht nur der Vektor der resultierenden Werkzeuglänge ohne Berücksichtigung des Verschleißes. |
| TOWWCS | Der Verschleißwert wird Im WKS auf das MKS umgerechnet. | Der Werkzeugvektor wird ohne Berücksichtigung des Verschleißes wie bei TOWMCS berechnet. |

| | | |
|---|---|---|
| TOWBCS | Der Verschleißwert wird Im BKS auf das MKS umgerechnet. | Der Werkzeugvektor wird ohne Berücksichtigung des Verschleißes wie bei TOWMCS berechnet. |
| TOWTCS | Der Verschleißwert wird Im Werkzeugkoordinatensystem auf das MKS umgerechnet. | Der Werkzeugvektor wird ohne Berücksichtigung des Verschleißes wie bei TOWMCS berechnet. |

TOWWCS, TOWBCS, TOWTCS: Der Verschleißvektor wird zum Werkzeugvektor addiert.

#### Lineare Transformation

Die Werkzeuglänge ist im MKS nur sinnvoll definierbar, wenn das MKS aus dem BKS durch eine lineare Transformation hervorgeht.

#### Nicht lineare Transformation

Ist z. B. mit TRANSMIT eine nicht lineare Transformation aktiv, dann wird bei Angabe des MKS als gewünschtes Koordinatensystem automatisch das BKS verwendet.

#### Keine kinematische Transformation und kein orientierbarer Werkzeugträger

Ist weder eine kinematische Transformation noch ein orientierbarer Werkzeugträger aktiv, dann fallen bis auf das WKS alle weiteren vier Koordinatensysteme zusammen. Damit unterscheidet sich nur das WKS von den übrigen. Da ausschließlich Werkzeuglängen zu bewerten sind, haben Translationen zwischen den Koordinatensystemen keine Bedeutung.

#### Einrechnung der Verschleißwerte

Das Settingdatum **SD42935 $SC_WEAR_TRANSFORM** legt fest, welche der drei Verschleißkomponenten:

- Verschleiß
- Summenkorrekturen fein
- Summenkorrekturen grob

einer Drehung durch eine Adaptertransformation oder einen orientierbaren Werkzeugträger unterworfen werden soll, wenn einer der folgenden G-Befehle aktiv ist:

- TOWSTD
  Grundstellung. Für Korrekturen in der Werkzeuglänge.
- TOWMCS
  Verschleißwerte im Maschinenkoordinatensystem (MKS).
- TOWWCS
  Verschleißwerte im Werkstückkoordinatensystem (WKS).
- TOWBCS
  Verschleißwerte im Basiskoordinatensystem (BKS).

- TOWTCS
  Verschleißwerte im Werkzeugkoordinatensystem an der Werkzeughalteraufnahme (T Werkzeugträgerbezug).
- TOWKCS
  Verschleißwerte im Koordinatensystem des Werkzeugkopfs bei kinetischer Transformation.

**Hinweis**

Die Bewertung der einzelnen Verschleißkomponenten (Zuordnung zu den Geometrieachsen, Vorzeichenbewertung) wird durch folgende Faktoren beeinflusst:

- Aktive Ebene
- Adaptertransformation
- Settingdaten:
  - SD42910 $SC_MIRROR_TOOL_WEAR
  - SD42920 $SC_WEAR_SIGN_CUTPOS
  - SD42930 $SC_WEAR_SIGN
  - SD42940 $SC_TOOL_LENGTH_CONST
  - SD42950 $SC_TOOL_LENGTH_TYPE

### 3.13.3.4 Werkzeuglänge und Ebenenwechsel

Mit gesetzten Settingdaten SD42940 $SC_TOOL_LENGTH_CONST ungleich Null können Sie Werkzeuglängenkomponenten wie Länge, Verschleiß und Basismaß zu den Geometrieachsen für Dreh- und Schleifwerkzeuge bei einen Ebenenwechsel zuordnen.

**SD42940 $SC_TOOL_LENGTH_CONST**

Settingdatum **ungleich** Null:

Die Zuordnung der Werkzeuglängenkomponenten (Länge, Verschleiß und Basismaß) zu den Geometrieachsen beim Wechsel der Bearbeitungsebene (`G17` - `G19`) wird nicht verändert.

Folgende Tabelle zeigt die Zuordnung der Werkzeuglängenkomponenten zu den Geometrieachsen für Dreh- und Schleifwerkzeuge (WZ-Typ 400 bis 599):

| Inhalt | Länge 1 | Länge 2 | Länge 3 |
|---|---|---|---|
| 17 | Y | X | Z |
| *) | X | Z | Y |
| 19 | Z | Y | X |
| -17 | X | Y | Z |
| -18 | Z | X | Y |
| -19 | Y | Z | X |

*) Jeder Wert ungleich 0, der nicht gleich einem der sechs aufgeführten Werte ist, wird wie der Wert 18 bewertet.

Folgende Tabelle zeigt die Zuordnung der Werkzeuglängenkomponenten zu den Geometrieachsen für alle anderen Werkzeuge (WZ-Typ < 400 bzw. > 599):

| Bearbeitungsebene | Länge 1 | Länge 2 | Länge 3 |
|---|---|---|---|
| *) | Z | Y | X |

| | | | |
|---|---|---|---|
| 18 | Y | X | Z |
| 19 | X | Z | Y |
| -17 | Z | X | Y |
| -18 | Y | Z | X |
| -19 | X | Y | Z |

*) Jeder Wert ungleich 0, der nicht gleich einem der sechs aufgeführten Werte ist, wird wie der Wert 17 bewertet.

**Hinweis**

Bei der Darstellung in den Tabellen wird davon ausgegangen, dass die Geometrieachsen bis 3 mit X, Y, Z bezeichnet werden. Für die Zuordnung einer Korrektur zu einer Achse ist nicht der Achsbezeichner, sondern die Achsreihenfolge maßgebend.

## 3.13.4 Online-Werkzeugkorrektur

### 3.13.4.1 Polynom-Funktion definieren (FCTDEF)

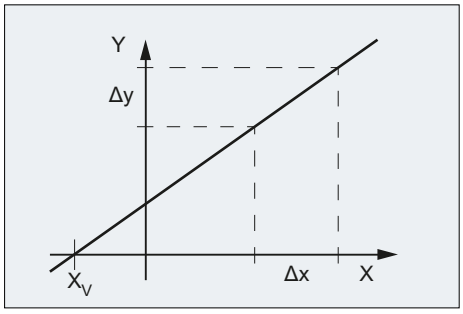
Bestimmte Abrichtstrategien (z. B. Abrichtrolle) zeichnen sich dadurch aus, dass die Schleifscheibe kontinuierlich (linear) mit der Zustellung der Abrichtrolle am Radius abnimmt. Hierfür benötigt man eine lineare Funktion zwischen der Zustellung der Abrichtrolle und dem Schreiben des Verschleißwerts der jeweiligen Länge. Die Definition der linearen Funktion erfolgt über die vordefinierte Prozedur FCTDEF(...) für Polynom-Funktionen bis maximal dritten Grades.

**Geradengleichung**

$y = f(x) = a_0 + a_1 * x_1$

$a_1$: Steigung der Geraden, mit $a_1 = \Delta x / \Delta y$

$a_0$: Verschiebung der Geraden entlang der X-Achse, mit $a_0 = -a1 * X_V$




**Syntax**

```
FCTDEF(<Func>,<LLimit>,<ULimit>,<a0>,<a1>,<a2>,<a3>)
```

## Bedeutung

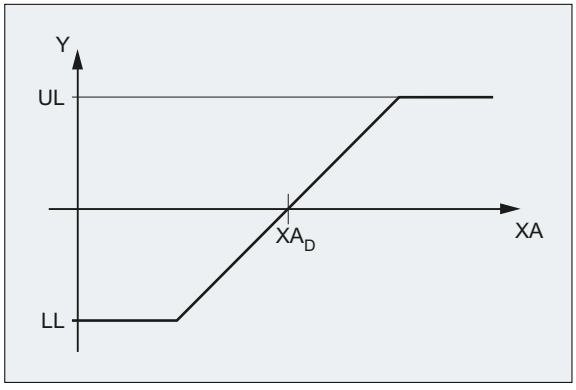
| | | |
|---|---|---|
| `FCTDEF(...)`: | Definition einer Polynom-Funktion für PUTFTOCF(...):<br>$y = f(x) = a_0 + a_1*x + a_2*x^2 + a_3*x^3$ | |
| **<**`Func`**>**: | Funktionsnummer | |
| | Datentyp: | INT |
| | Wertebereich: | 1, 2, 3 |
| `<LLimit>`: | Unterer Begrenzungswert | |
| | Datentyp: | REAL |
| `<ULimit>`: | Oberer Begrenzungswert | |
| | Datentyp: | REAL |
| `<a0>,<a1>,<a2>,<a3>`: | Koeffizienten der Polynom-Funktion | |
| | Datentyp: | REAL |

## Beispiel

### Festlegungen

- Funktionsnummer: 1
- Unterer und Oberer Begrenzungswert: -100, 100
- Steigung der Kennlinie: $a_1 = 1$
- Der Arbeitspunkt soll in der Mitte der Kennlinie liegen. Die Kennlinie muss dazu anhand der Sollposition der Achse XA im WKS zum Zeitpunkt der Funktionsdefinition im NC-Programm in negativer Y-Richtung verschoben werden: $a_0 = -a_1 * XA_D = -1 * \$AA_IW$
- $a_2 = a_3 = 0$

### Kennlinie




UL Oberer Begrenzungswert

LL Unterer Begrenzungswert

$XA_D$ Sollwert der Achse XA zum Zeitpunkt der Funktionsdefinition im NC-Programm

### Programmierung

| Programmcode | Kommentar |
|---|---|
| FCTDEF(1,-100,100,-$AA_IW[XA],1) | ; Funktionsdefinition |

## 3.13.4.2 Online-Werkzeugkorrektur schreiben, kontinuierlich (PUTFTOCF)

Mit der vordefinierten Prozedur PUTFTOCF(...) wird eine Online-Werkzeugkorrektur über eine zuvor mit FCTDEF(...) (Seite 761) definierte Polynom-Funktion vorgenommen.

**Hinweis**

Die Online-Werkzeugkorrektur kann auch über eine Synchronaktion erfolgen.

Weitere Informationen siehe Funktionshandbuch Synchronaktionen.

### Syntax

```
PUTFTOCF(<Func>,<RefVal>,<ToolPar>,<Chan>,<Sp>)
```

### Bedeutung

| | | |
|---|---|---|
| PUTFTOCF(...): | Online-WZK schreiben, kontinuierlich satzweise anhand der mit FCTDEF(...) definierten Polynom-Funktion | |
| <Func>: | Funktionsnummer, festgelegt bei der Funktionsdefinition mit FCTDEF(...) | |
| | Datentyp: | INT |
| | Wertebereich: | 1, 2, 3 |
| <RefVal>: | Bezugswert, von dem die Korrektur abgeleitet werden soll (z. B. Sollwert einer Achse) | |
| | Datentyp: | VAR REAL |
| <ToolPar>: | Nummer des Verschleißparameters (Länge 1, 2 oder 3), in dem der Korrekturwert verrechnet werden soll | |
| | Datentyp: | INT |
| <Chan>: | Nummer des Kanals, in dem die Online-WZK wirksam werden soll<br>**Hinweis**:<br>Nur erforderlich, wenn die Korrektur nicht im aktiven Kanal wirksam werden soll. | |
| | Datentyp: | INT |
| <Sp>: | Nummer der Spindel, für die die Online-WZK wirksam werden soll<br>**Hinweis:**<br>Nur erforderlich, wenn statt dem aktiven, im Einsatz befindlichen Werkzeug eine nicht aktive Schleifscheibe korrigiert werden soll. | |
| | Datentyp: | INT |

### 3.13.4.3 Online-Werkzeugkorrektur schreiben, diskret (PUTFTOC)

#### Funktion

Mit der vordefinierten Prozedur PUTFTOC(...) wird eine Online-Werkzeugkorrektur mit einem festen Korrekturwert vorgenommen.

#### Syntax

```
PUTFTOC(<CorrVal>,<ToolPar>,<Chan>,<Sp>)
```

#### Bedeutung

| | | |
|---|---|---|
| `PUTFTOC(...)`: | Online-WZK schreiben | |
| `<CorrVal>`: | Korrekturwert, der zum Verschleißparameter addiert wird | |
| | Datentyp: | REAL |
| `<ToolPar>`: | Nummer des Verschleißparameters (Länge 1, 2 oder 3), in dem der Korrekturwert verrechnet werden soll | |
| | Datentyp: | INT |
| `<Chan>`: | Nummer des Kanals, in dem die Online-WZK wirksam werden soll<br>**Hinweis:**<br>Nur erforderlich, wenn die Korrektur nicht im aktiven Kanal wirksam werden soll. | |
| | Datentyp: | INT |
| `<Sp>`: | Nummer der Spindel, für die die Online-WZK wirksam werden soll<br>**Hinweis:**<br>Nur erforderlich, wenn statt dem aktiven, im Einsatz befindlichen Werkzeug eine nicht aktive Schleifscheibe korrigiert werden soll. | |
| | Datentyp: | INT |

### 3.13.4.4 Online-Werkzeugkorrektur ein-/ausschalten (FTOCON/FTOCOF)

Mit den G-Befehlen FTOCON und FTOCOF wird die Online-Werkzeugkorrektur ein- bzw. ausgeschaltet.

#### Syntax

```
FTOCON
...
FTOCOF
```

### Bedeutung

| | |
|---|---|
| FTOCON: | Online-Werkzeugkorrektur einschalten<br>Der Befehl muss in dem Kanal programmiert werden, in dem die Online-Werkzeugkorrektur eingeschaltet werden soll. |
| FTOCOF: | Online-Werkzeugkorrektur ausschalten<br>Der Befehl muss in dem Kanal programmiert werden, in dem die Online-Werkzeugkorrektur ausgeschaltet werden soll.<br>**Hinweis:**<br>Mit FTOCOF wird die Werkzeugkorrektur nicht weiter herausgefahren. In den schneidenspezifischen Korrekturdaten bleibt aber der mit PUTFTOC/PUTFTOCF eingerechnete Wert erhalten.<br>Zum endgültigen Deaktivieren der Online-WZK muss nach FTOCOF noch eine An-/Abwahl des Werkzeugs (T...) erfolgen. |

## 3.13.5 3D-Werkzeugradiuskorrektur

### 3.13.5.1 3D-Werkzeugradiuskorrektur für das 3D-Umfangsfräsen anwählen (CUT3DC, CUT3DCD, ISD)

Die 3D-Werkzeugradiuskorrektur (3D-WRK) für das 3D-Umfangsfräsen ohne Berücksichtigung von Begrenzungsflächen wird mit dem modal wirksamen G-Befehl CUT3DC bzw. CUT3DCD angewählt.

Die eigentliche Aktivierung erfolgt mit G41 bzw. G42. Ausgeschaltet wird die Werkzeugradiuskorrektur mit G40.

### Syntax

```
G41/G42 ORIC/ORID ISD=... CUT3DC/CUT3DCD CDOF2 X... Y... Z...
...
G40 X... Y... Z...
```

### Bedeutung

| | | |
|---|---|---|
| CUT3DC: | 3D-WRK für das Umfangsfräsen (nur bei aktiver 5-Achs-Transformation) | |
| CUT3DCD: | Auf ein Differenzwerkzeug bezogene 3D-WRK für das Umfangsfräsen (nur bei aktiver 5-Achs-Transformation)<br>Die Radiusdifferenz wird durch den WZ-Parameter $TC_DP15 festgelegt. | |
| G41/G42 X... Y... Z...: | Werkzeugradiuskorrektur einschalten | |
| | G41: | Werkzeugradiuskorrektur links von der Kontur |
| | G42: | Werkzeugradiuskorrektur rechts von der Kontur |
| | **Hinweis:**<br>Das Einschalten muss in einem Linearsatz (G0/G1) erfolgen. | |

| | | |
|---|---|---|
| CDOF2: | Kollisionsüberwachung für 3D-Umfangsfräsen ausschalten | |
| ORIC/ORID: | Über die G-Befehle ORIC und ORID wird das Verhalten bei Orientierungsänderungen an Außenecken festgelegt. | |
| | ORIC: | Orientierungsänderungen an Außenecken werden dem einzufügenden Kreissatz überlagert. |
| | ORID: | Orientierungsänderungen an Außenecken werden vor dem einzufügenden Kreissatz ausgeführt. |
| ISD=<Value>: | Mit der Adresse ISD kann beim Umfangsfräsen und **aktiver** 3D-Werkzeugradiuskorrektur die Eintauchtiefe des Werkzeugs geändert werden. | |
| | <Value>: | Länge der Eintauchtiefe |
| G40 X... Y... Z...: | Werkzeugradiuskorrektur ausschalten<br>**Hinweis:**<br>Das Ausschalten muss in einem Linearsatz (G0/G1) mit Geometrieachsbewegungen erfolgen. | |

### Hinweis

Die G-Befehle zur Anwahl der 3D-WRK werden im Anfahrsatz ausgewertet, d. h. typischerweise in dem Satz, der G41 oder G42 enthält.

G41 bzw. G42 kann auch in Sätzen ohne Verfahrbewegung in den für die Korrektur relevanten Geometrieachsen programmiert sein. In diesem Fall ist der Anfahrsatz der erste auf einen solchen Satz folgende Verfahrsatz.

Ein Wechsel der 3D-WRK-Variante bei aktiver Werkzeugradiuskorrektur wird ohne Alarm ignoriert.

## Beispiel

```
Programmcode                          Kommentar
                                      ; Definition des Werkzeugs D1:
$TC_DP1[1,1]=120                      ; Typ (Schaftfräser)
$TC_DP3[1,1]=20                       ; Längenkorrekturvektor
$TC_DP6[1,1]=8                        ; Radius

N10 X0 Y0 Z0 T1 D1 F12000             ; Anwahl des Werkzeugs.
N20 TRAORI(1)                         ; Einschalten der Transformation.
N30 G42 ORIC ISD=10 CUT3DC G64 X30    ; Aktivieren des 3D-Umfangsfräsens,
                                      ; Orientierungsänderungen an Außenecken kontinuierlich,
                                      ; Eintauchtiefe: 10mm
N40 ORIWKS A30 B15                    ; Orientierungsänderung an einer Ecke durch Angabe von Achspositionen.
N50 Y20 A3=1 C3=1                     ; Verfahrsatz mit Orientierungsänderung,
                                      ; Angabe der Orientierung mit Richtungsvektor.
N60 X50 Y30                           ; Verfahrsatz mit konstanter Orientierung.
N70 Y50 A3=0.5 B3=1 C3=5              ; Verfahrsatz mit Orientierungsänderung.
N80 M63                               ; Satz ohne Verfahrinformation.
```

| Programmcode | Kommentar |
|---|---|
| N90 X0 ISD=20 | ; Verfahrsatz mit Änderung der Eintauchtiefe. |
| N100 G40 Y0 | ; Deaktivieren der Werkzeugradiuskorrektur. |
| N110 M30 | |

## Weitere Informationen

### Bahn und Orientierung

Die hier benutzte Variante des Umfangsfräsens ist durch die Vorgabe einer Bahn (Leitlinie) und der zugehörigen Orientierung realisiert. Bei dieser Art der Bearbeitung ist auf der Bahn die Werkzeugform ohne Bedeutung. Entscheidend ist allein der Radius am Werkzeugeingriffspunkt.

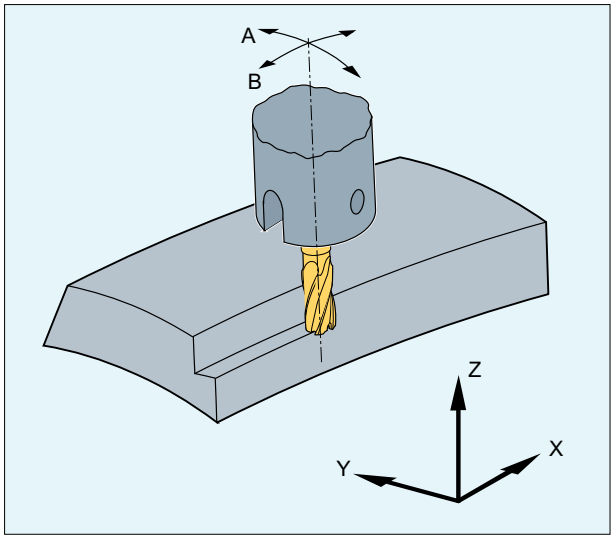



Bild 3-8 Umfangsfräsen

### Anfahrverhalten

Das Anfahrverhalten ist bei den 3D-Varianten der Werkzeugradiuskorrektur immer NORM.

### Verhalten an Außenecken

An Außenecken werden beim Umfangsfräsen mit 3D-WRK analog zu den Verhältnissen bei der 2½D-WRK die G-Befehle der Gruppe 18 (Eckenverhalten Werkzeugkorrektur) ausgewertet:

- G450: Übergangskreis (Werkzeug umfährt Werkstückecken auf einer Kreisbahn)
  Im Gegensatz zur Lösung bei der 2½D-WRK ist das eingefügte Konturelement an einer Außenecke immer ein Kreis mit dem Radius 0, auf den die Werkzeugradiuskorrektur so wirkt wie auf jede andere programmierte Bahn auch. Es ist nicht möglich, an Stelle der Kreise Kegelschnitte einzufügen. Die Adresse DISC hat deshalb in diesem Fall keine Bedeutung und wird nicht ausgewertet.
- G451: Schnittpunkt der Äquidistanten (Werkzeug schneidet in der Werkstückecke frei)
  Der Schnittpunkt wird bestimmt, indem die Offsetkurven der beiden beteiligten Sätze verlängert werden und deren Schnittpunkt in der Ebene senkrecht zur Werkzeugorientierung an der Ecke bestimmt wird.

Das Schnittpunktverfahren (G451) wird nicht verwendet, wenn zwischen den beteiligten Verfahrsätzen mindestens ein Satz eingefügt ist, der eine Änderung der Werkzeugorientierung enthält. In einem solchen Fall wird an der Ecke immer ein Kreis eingefügt.

### Verhalten bei Orientierungsänderungen an Außenecken

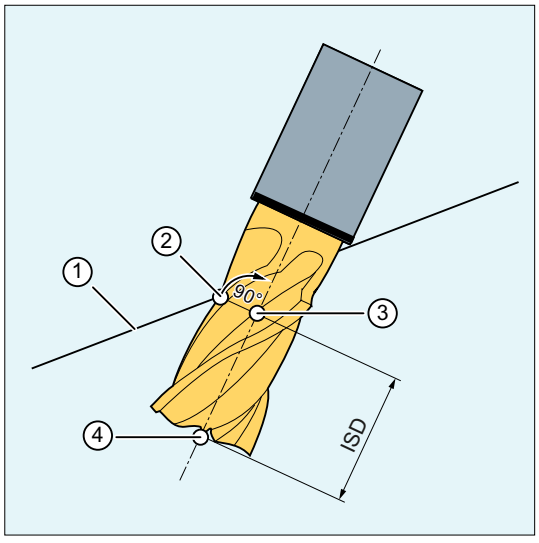
Über die G-Befehle ORIC und ORID kann festgelegt werden, ob Orientierungsänderungen, die zwischen den beiden die Ecke bildenden Verfahrsätzen programmiert wurden, vor Beginn des eingefügten Kreissatzes (ORID) oder gleichzeitig mit diesem (ORIC) ausgeführt werden.

### Eintauchtiefe

Die Eintauchtiefe des Fräsers ist der Abstand des Fräserhilfspunkts von der Werkzeugspitze.

Der Fräserhilfspunkt ist die senkrechte Projektion des Fräserbearbeitungspunkts auf der programmierten Bahn auf die Werkzeuglängsachse.

Mit der Eintauchtiefe wird somit die Lage des Bearbeitungspunkts auf der Mantelfläche des Werkzeugs eingestellt.




① Programmierte Bahn
② Fräserbearbeitungspunkt
③ Fräserhilfspunkt
④ Fräserspitze
ISD Eintauchtiefe (InSertion Depth)

Bild 3-9 Eintauchtiefe

### Werkzeugradiuskorrektur bezogen auf ein Differenzwerkzeug

Die auf ein Differenzwerkzeug bezogene 3D-WRK für das Umfangsfräsen wird durch den Befehl CUT3DCD aktiviert. Sie ist anzuwenden, wenn sich die programmierte Kontur auf die Mittelpunktsbahn eines Normwerkzeugs bezieht und die Bearbeitung mit einem davon abweichenden Werkzeug erfolgt. Bei der Berechnung der 3D-Werkzeugradiuskorrektur werden dann nur der Verschleißwert des Radius des aktiven Werkzeugs ($TC_DP15) und die gegebenenfalls programmierten Werkzeugkorrekturoffsets OFFN und TOFFR/TOFFLR eingerechnet. Der Grundradius ($TC_DP6) des aktiven Werkzeugs wird **nicht** eingerechnet.

#### Taschenfräsen mit schrägen Seitenwänden für Umfangsfräsen mit CUT3DC

Bei dieser 3D-Werkzeugradiuskorrektur wird eine Abweichung des Fräserradius kompensiert, indem in Richtung der Flächennormalen der zu bearbeitenden Fläche zugestellt wird. Dabei bleibt die Ebene, in der die Stirnseite des Fräsers liegt unverändert, wenn die Eintauchtiefe ISD gleich geblieben ist. Ein Fräser mit z. B. kleinerem Radius gegenüber einem Normwerkzeug würde dann den Taschenboden, der auch die Begrenzungsfläche darstellt, nicht erreichen. Für eine automatische Zustellung des Werkzeugs muss der Steuerung diese Begrenzungsfläche bekannt sein, siehe Kapitel "3D-Umfangsfräsen unter Berücksichtigung einer Begrenzungsfläche (CUT3DCC, CUT3DCCD) (Seite 775)".

#### Advanced Surface / Top Surface

**Hinweis**

Bei Verwendung der Werkzeugradiuskorrektur CUT3DCD in Kombination mit der lizenzpflichtigen Option "Advanced Surface" oder "Top Surface" sind die Einstellempfehlungen bezüglich "Advanced Surface" / "Top Surface" zu beachten!

Zur Überprüfung der eingestellten Daten steht über das SIOS-Portal ein spezielles Prüfprogramm zur Verfügung.

→ Prüfprogramm für Advanced Surface / Top Surface (https://support.industry.siemens.com/cs/ww/de/view/109738423)

### 3.13.5.2 3D-Werkzeugradiuskorrektur für das 3D-Stirnfräsen anwählen (CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD)

Die 3D-Werkzeugradiuskorrektur (3D-WRK) für das 3D-Stirnfräsen wird mit dem modal wirksamen G-Befehl CUT3DF, CUT3DFS, CUT3DFF bzw. CUT3DFD angewählt.

Die eigentliche Aktivierung erfolgt mit G41 bzw. G42.

Um die Werkzeugradiuskorrektur berechnen zu können, ist beim 3D-Stirnfräsen die Definition der Flächennormale der zu bearbeitenden Ebene erforderlich. Diese muss im Satz mit G41 bzw. G42 über die Adressen A4, B4, C4 und A5, B5, C5 erfolgen.

Ausgeschaltet wird die Werkzeugradiuskorrektur mit G40.

## Syntax

```
G41/G42 ORIC/ORID CUT3DF/CUT3DFS/CUT3DFF/CUT3DFD X... Y... Z... A4=... B4=...
C4=... A5=... B5=... C5=...
...
G40 X... Y... Z...
```

## Bedeutung

| | | |
|---|---|---|
| CUT3DFS: | 3D-WRK für das Stirnfräsen mit konstanter Orientierung. Die Werkzeugorientierung ist durch G17 - G19 festgelegt und wird durch Frames nicht beeinflusst. | |
| CUT3DFF: | 3D-WRK für das Stirnfräsen mit konstanter Orientierung. Die Werkzeugorientierung ist die durch G17 - G19 festgelegte und gegebenenfalls durch einen Frame gedrehte Richtung. | |
| CUT3DF: | 3D-WRK für das Stirnfräsen mit Orientierungsänderung (nur bei aktiver 5-Achs-Transformation) | |
| CUT3DFD: | Auf ein Differenzwerkzeug bezogene 3D-WRK für das Stirnfräsen mit Orientierungsänderung (nur bei aktiver 5-Achs-Transformation)<br>Die Radiusdifferenz wird durch den WZ-Parameter $TC_DP15 festgelegt.<br>**Hinweis**:<br>CUT3DFD ist nur in Kombination mit der "Glättung der Flächennormalen beim 3D-Stirnfräsen" möglich. Diese wird aktiviert über den Aufruf der lizenzpflichtigen Funktion "Top Surface" mittels CYCLE832(...). | |
| G41/G42 X... Y... Z...: | Werkzeugkorrektur einschalten<br>Das Verhalten bei G41 und bei G42 ist beim 3D-Stirnfräsen identisch.<br>**Hinweis:**<br>Das Einschalten muss in einem Linearsatz (G0/G1) erfolgen. | |
| A4/5=... B4/5=... C4/5=...: | Definition der Flächenormale der zu bearbeitenden Ebene | |
| | A4=... B4=... C4=...: | Definition am Satzanfang |
| | A5=... B5=... C5=...: | Definition am Satzende |
| ORIC/ORID: | Über die G-Befehle ORIC und ORID wird das Verhalten bei Orientierungsänderungen an Außenecken festgelegt. | |
| | ORIC: | Orientierungsänderungen an Außenecken werden dem einzufügenden Kreissatz überlagert. |
| | ORID: | Orientierungsänderungen werden vor dem Kreissatz ausgeführt. |
| G40 X... Y... Z...: | Werkzeugradiuskorrektur ausschalten<br>**Hinweis:**<br>Das Ausschalten muss in einem Linearsatz (G0/G1) mit Geometrieachsbewegungen erfolgen. | |

### Hinweis

G41 bzw. G42 kann auch in Sätzen ohne Verfahrbewegung in den für die Korrektur relevanten Geometrieachsen programmiert sein. In diesem Fall ist der Anfahrsatz der erste auf einen solchen Satz folgende Verfahrsatz.

Ein Wechsel der 3D-WRK-Variante bei aktiver Werkzeugradiuskorrektur wird ohne Alarm ignoriert.

## Beispiele

### Beispiel 1: 3D-Stirnfräsen mi CUT3DF

| **Programmcode** | **Kommentar** |
|---|---|
| N10 | ; Definition des Werkzeugs D1: |
| N20 $TC_DP1[1,1]=121 | ; Werkzeugtyp (Torusfräser) |
| N30 $TC_DP3[1,1]=20 | ; Längenkorrektur |
| N40 $TC_DP6[1,1]=5 | ; Radius |
| N50 $TC_DP7[1,1]=3 | ; Verrundungsradius |
| N60 | |
| N70 | |
| N80 X0 Y0 Z0 A0 B0 C0 G17 T1 D1 F12000 | ; Anwahl des Werkzeugs. |
| N90 TRAORI(1) | ; Orientierungstransformation anwählen. |
| N100 B4=-1 C4=1 | ; Definition der Ebene. |
| N110 **G41** ORID **CUT3DF** G64 X10 Y0 Z0 | ; Werkzeugkorrektur einschalten. |
| N120 X30 | |
| N130 Y20 A4=1 C4=1 | ; Außenecke, Neudefinition der Ebene. |
| N140 B3=1 C3=5 | ; Orientierungsänderung mit ORID. |
| N150 B3=1 C3=1 | ; Orientierungsänderung mit ORID. |
| N160 X-10 A5=1 C5=2 ORIC | |
| N170 A3=-2 C3=1 | ; Orientierungsänderung mit ORIC |
| N180 A3=-1 C3=1 | ; Orientierungsänderung mit ORIC |
| N190 Y-10 A4=-1 C4=3 | ; Neudefinition der Ebene. |
| N200 X-20 Y-20 Z10 | ; Innenecke mit Vorgängersatz. |
| N210 X-30 Y10 A4=1 C4=1 | ; Innenecke, Neudefinition der Ebene. |
| N220 A3=1 B3=0.5 C3=1.7 | ; Orientierungsänderung mit ORIC. |
| N230 X-20 Y30 A4=1 B4=-2 C4=3 ORID | |
| N240 A3 = 0.5 B3=-0.5 C3=1 | ; Orientierungsänderung. |
| N250 X0 Y30 C4=1 | ; Bahnbewegung, neue Ebene, |
| | ; Orientierung mit Relativprogrammierung. |
| N260 BSPLINE X20 Z15 | ; Splinebeginn, Relativprogrammierung der Orientierung |
| N270 X30 Y25 Z18 | ; bleibt während Spline aktiv. |
| N280 X40 Y20 Z13 | |
| N290 X45 Y0 PW=2 Z8 | |
| N300 Y-20 | |
| N310 G2 ORIMKS A30 B45 I-20 X25 Y-40 Z0 | ; Helix, Orientierung mit Achsprogrammierung. |
| N320 G1 X0 A3=-0.123 B3=0.456 C3=2.789 B4=-1 C4=5 B5=-1 C5=2 | ; Bahnbewegung, Orientierung, nicht konstante Ebene. |
| N330 X-20 G40 | ; Ausschalten der Werkzeugradiuskorrektur. |
| N340 M30 | |

### Beispiel 2: Aus einem CAD-System erzeugtes NC-Programm (Ausschnitt) mit CUT3DFD

| **Programmcode** | **Kommentar** |
|---|---|
| N01 G710 | |

| **Programmcode** | **Kommentar** |
|---|---|
| N03 T="12" | |
| | |
| N06 S5305 M03 | |
| N07 G642 | |
| ; Anfahren der Startposition im MKS unter Berücksichtigung der Werkzeuglänge. | |
| G00 G90 X-250.62787 Y-38.37944 A=DC(253.12719) B-12.49543 | |
| G00 G90 Z251.80052 | |
| ; Ende der Positionierung im MKS. | |
| ; | |
| TRAORI(1) | ; Orientierungstransformation anwählen. |
| G500 | |
| D1 | |
| CYCLE832(0.01, _TOP_SURFACE_SMOOTH_ON + _ORI_FINISH, 1) | ; Aufruf CYCLE832 mit:<br>; Konturtoleranz = 0,01 mm,<br>; Bearbeitungsart: Top Surface mit Glättung,<br>; Schlichten mit Eingabe einer Orientierungstoleranz,<br>; Orientierungstoleranz = 1 Grad |
| **CUT3DFD** | |
| N08 G90 G94 | |
| N09 G00 X-269.21195 Y128.32027 Z1.18577 A3=-.216361688 B3=.934284397 C3=-.283373051 | ; Die Sätze N09 bis N10 sind der schnelle Anteil der Anfahrbewegung an das Werkstück mit konstanter Orientierung. |
| N10 G00 X-251.90301 Y53.57752 Z23.85561 | |
| N11 G01 X-247.57578 Y34.89183 Z29.52308 F50000.00000 | ; In den Sätzen N11 bis N21 wird der langsame Anteil der Anfahrbewegung realisiert. Das Werkzeug befindet sich bereits nahe am Werkstück, außerdem sind jetzt die Formenbau-Einstellungen (z.B. COMPSURF) aus dem CYCLE832 aktiv (wegen aktivem G01). Der Bahnweg dieser sog. Einschwingphase für das Formenbau-Verhalten sollte etwa das 1000-fache der Konturtoleranz betragen (in diesem Beispiel also 10 mm). |
| N12 X-247.69126 Y33.82182 Z24.78219 F1061.00000 | |
| N13 X-247.76560 Y33.13299 Z21.73022 | |
| N14 X-247.82755 Y32.55897 Z19.18691 | |
| N15 X-247.87918 Y32.08062 Z17.06748 | |
| N16 X-247.92220 Y31.68200 Z15.30129 | |
| N17 X-247.95805 Y31.34981 Z13.82947 A3=-.216361686 B3=.934284391 C3=-.283373071 | |
| N18 X-247.98792 Y31.07299 Z12.60295 A3=-.216360662 B3=.934280801 C3=-.283385691 | |
| N19 X-248.01282 Y30.84230 Z11.58085 A3=-.216336015 B3=.934194446 C3=-.283689030 | |
| N20 X-248.03357 Y30.65006 Z10.72910 A3=-.216233089 B3=.933833626 C3=-.284952647 | |

| **Programmcode** | **Kommentar** |
|---|---|
| N21 X-248.05086 Y30.48986 Z10.01931<br>A5=-.060687572 B5=.974940255 C5=-.214029243<br>A3=-.215712821 B3=.932005189 C3=-.291263295 | |
| N22 **G41** X-248.06237 Y30.32400 Z9.36695<br>A5=-.060431854 B5=.973045457 C5=-.222554556<br>A3=-.214974689 B3=.929398552 C3=-.300007025<br>F1061.03295 | ; Ab N22 ist die Flächennormale erstmals vollständig über den gesamten Satz definiert (d.h., Flächennormale am Ende des Vorgängersatzes N21 ist vorhanden, somit die Flächennormale am Anfang des Satzes N22 und die Flächennormale am Ende des Satzes N22). Damit ist die Voraussetzung für das Einschalten der Werkzeugkorrektur mit G41/G42 erfüllt. |
| N23 X-248.07130 Y30.15119 Z8.71082<br>A5=-.060165696 B5=.971048883 C5=-.231179920<br>A3=-.214177198 B3=.926684940 C3=-.308841625 | |
| N24 X-248.07829 Y29.97126 Z8.05094<br>A5=-.059884286 B5=.968941717 C5=-.239928784<br>A3=-.213318480 B3=.923853466 C3=-.317789237 | |
| N25 X-248.08317 Y29.78487 Z7.38844<br>A5=-.059584206 B5=.966718449 C5=-.248807482<br>A3=-.212397895 B3=.920898045 C3=-.326854594 | |
| N26 X-248.08578 Y29.59254 Z6.72679<br>A5=-.059263963 B5=.964380907 C5=-.257793037<br>A3=-.211418355 B3=.917822366 C3=-.336012474 | |
| ... | |

**Hinweis**

Beim 3D-Stirnfräsen mit CUT3DFD ist für das Einschalten der Werkzeugkorrektur mit G41/G42 die Definition der Flächennormalen erforderlich. Die Programmierung von G41/G42 ohne Definition der Flächennormalen führt zur Ausgabe eines Alarms.

## Siehe auch

CYCLE832 - High Speed Settings (Seite 1126)

## Weitere Informationen

### 3D-Stirnfräsen

Für diese Art des 3D-Fräsens wird die zeilenweise Beschreibung der 3D-Bahnen auf der Werkstückoberfläche benötigt. Die Berechnungen werden unter Berücksichtigung der Werkzeugform und Werkzeugabmessungen üblicherweise im CAM durchgeführt. Der Postprozessor schreibt in das Teileprogramm – neben den NC-Sätzen – die Werkzeugorientierungen (bei aktiver 5-Achstransformation) und den G-Befehl für die gewünschte 3D-Werkzeugkorrektur. Hierdurch hat der Maschinenbediener die Möglichkeit – abweichend von dem für die Berechnung der NC-Bahnen verwendeten Werkzeug – geringfügig kleinere Werkzeuge einzusetzen.

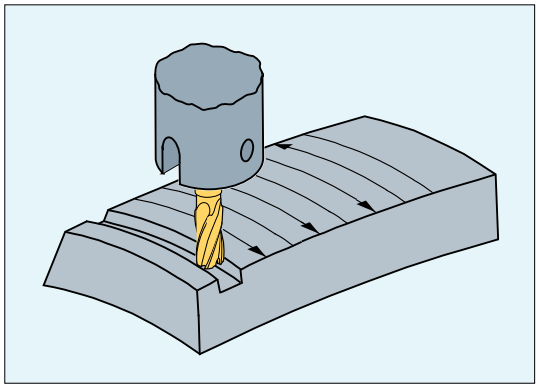


Bild 3-10 Stirnfräsen

### Anfahrverhalten

Das Anfahrverhalten ist bei den 3D-Varianten der Werkzeugradiuskorrektur immer NORM.

### Verhalten an Außenecken

Außenecken werden beim Stirnfräsen als Kreise mit dem Radius 0 behandelt, wobei die Kreisebene von der Endtangente des ersten und der Starttangente des zweiten Satzes aufgespannt wird. Damit ist auch eine Orientierungsänderung beim Satzübergang möglich. An einer Außenecke wird somit immer ein Kreis als Konturelement eingefügt. Das Schnittpunktverfahren steht beim Stirnfräsen nicht zur Verfügung.

### Verhalten bei Orientierungsänderungen an Außenecken

Über die G-Befehle ORIC und ORID kann festgelegt werden, ob Orientierungsänderungen, die zwischen den beiden die Ecke bildenden Verfahrsätzen programmiert wurden, vor Beginn des eingefügten Kreissatzes (ORID) oder gleichzeitig mit diesem (ORIC) ausgeführt werden.

### Werkzeugradiuskorrektur bezogen auf ein Differenzwerkzeug

Die auf ein Differenzwerkzeug bezogene 3D-Werkzeugradiuskorrektur wird durch den Befehl CUT3DFD angewählt. Sie ist anzuwenden, wenn sich die programmierte Kontur auf die Mittelpunktsbahn eines Normwerkzeugs bezieht und die Bearbeitung mit einem davon abweichenden Werkzeug erfolgt. Bei der Berechnung der 3D-Werkzeugradiuskorrektur werden dann nur der Verschleißwert des Radius des aktiven Werkzeugs ($TC_DP15) und die gegebenenfalls programmierten Werkzeugkorrekturoffsets OFFN und TOFFR/TOFFLR eingerechnet. Der Grundradius ($TC_DP6) des aktiven Werkzeugs wird **nicht** eingerechnet.

Das 3D-Stirnfräsen mit CUT3DFD ist nur in Kombination mit der "Glättung der Flächennormalen beim 3D-Stirnfräsen" möglich. Diese wird aktiviert über den Aufruf der lizenzpflichtigen Funktion "Top Surface" mittels CYCLE832(...). Die Aktivierung muss **vor** dem Einschalten der Werkzeugkorrektur mit G41/G42 erfolgen, und zwar nicht direkt vor dem Werkzeugeingriff, sondern schon eine Bahnweglänge vorher, die etwa dem 1000-fachen der Konturtoleranz entspricht (z. B. 1000 x 0,01 mm = 10 mm). Die Deaktivierung muss in umgekehrter Reihenfolge ablaufen: erst Ausschalten der Werkzeugkorrektur mit G40, dann, nach einer Bahnweglänge, die etwa dem 1000-fachen der Konturtoleranz entspricht, die Deaktivierung durch z. B. CUT2D (o. ä.).

Um die "Glättung der Flächennormalen beim 3D-Stirnfräsen" anwenden zu können, ist zusätzlich die Freischaltung der Funktion "Interpolation der Flächennormalen über Polynome" erforderlich:

MD28291 $MC_MM_SMOOTH_SURFACE_NORMALS = TRUE

**Hinweis**

Für das 3D-Stirnfräsen mit CUT3DFD in Kombination mit "Top Surface" sind die Einstellempfehlungen bezüglich "Top Surface" zu beachten!

Zur Überprüfung der eingestellten Daten steht über das SIOS-Portal spezielles Prüfprogramm zur Verfügung.

→ Prüfprogramm für Top Surface (https://support.industry.siemens.com/cs/ww/de/view/109738423)

## Siehe auch

CYCLE832 - High Speed Settings (Seite 1126)

### 3.13.5.3 3D-Umfangsfräsen unter Berücksichtigung einer Begrenzungsfläche (CUT3DCC, CUT3DCCD)

Beim 3D-Umfangsfräsen mit kontinuierlicher oder konstanter Veränderung der Werkzeugorientierung wird häufig die Werkzeugmittelpunktsbahn für ein definiertes Normwerkzeug programmiert. Da in der Praxis oft nicht die passenden Normwerkzeuge zur Verfügung stehen, kann ein von einem Normwerkzeug nicht allzu stark abweichendes Werkzeug (≤ 5 %) eingesetzt werden.

Mit CUT3DCCD wird für ein reales Differenzwerkzeug eine Begrenzungsfläche berücksichtigt, die das programmierte Normwerkzeug beschreiben würde. Das NC-Programm beschreibt die Mittelpunkbahn des Normwerkzeugs.

Mit CUT3DCC wird bei Verwendung von zylindrischen Werkzeugen eine Begrenzungsfläche berücksichtigt, die das programmierte Normwerkzeug erreicht hätte. Das NC-Programm beschreibt die Kontur auf der Bearbeitungsfläche.

Der Flächennormalenvektor der Begrenzungsfläche wird wie beim 3D-Stirnfräsen mit A4, B4, C4 und A5, B5, C5 angegeben.

## Syntax

```
G41/G42 CUT3DCCD/CUT3DCC CDOF2 X... Y... Z... A4=... B4=... C4=... A5=... B5=...
C5=...
...
G40 X... Y... Z...
```

## Bedeutung

| | | |
|---|---|---|
| CUT3DCCD: | 3D-WRK für das Umfangsfräsen unter Berücksichtigung einer Begrenzungsfläche mit Differenzwerkzeug auf der Werkzeugmittelpunktbahn: Zustellung zur Begrenzungsfläche | |
| CUT3DCC: | 3D-WRK für das Umfangsfräsen unter Berücksichtigung einer Begrenzungsfläche mit 3D-Radiuskorrektur: Kontur an der Bearbeitungsfläche | |
| G41/G42 X... Y... Z...: | Werkzeugradiuskorrektur einschalten | |
| | G41: | Werkzeugradiuskorrektur links von der Kontur |
| | G42: | Werkzeugradiuskorrektur rechts von der Kontur |
| | **Hinweis:**<br>Das Einschalten muss in einem Linearsatz (G0/G1) erfolgen. | |
| CDOF2: | Kollisionsüberwachung für 3D-Umfangsfräsen ausschalten | |
| A4/5=... B4/5=...<br>C4/5=...: | Definition der Flächenormale der Begrenzungsfläche | |
| | A4=... B4=... C4=...: | Definition am Satzanfang |
| | A5=... B5=... C5=...: | Definition am Satzende |
| G40 X... Y... Z...: | Werkzeugradiuskorrektur ausschalten<br>**Hinweis:**<br>Das Ausschalten muss in einem Linearsatz (G0/G1) mit Geometrieachsbewegungen erfolgen. | |

### Hinweis

Die G-Befehle zur Anwahl der 3D-WRK werden im Anfahrsatz ausgewertet, d. h. typischerweise in dem Satz, der G41 oder G42 enthält.

G41 bzw. G42 kann auch in Sätzen ohne Verfahrbewegung in den für die Korrektur relevanten Geometrieachsen programmiert sein. In diesem Fall ist der Anfahrsatz der erste auf einen solchen Satz folgende Verfahrsatz.

Ein Wechsel der 3D-WRK-Variante bei aktiver Werkzeugradiuskorrektur wird ohne Alarm ignoriert.

## Beispiel

```
Programmcode                                    Kommentar
N10 $TC_DP1[1,1]=120                            ; Zylinderfräser
N20 $TC_DP6[1,1]=10
N30 $TC_DP15[1,1]=-3
...
; Bearbeitung mit Zylinderfräser und CUT3DCCD
N110 TRAORI                                     ; Einschalten der Transformation.
N120 A4=0 B4=0 C4=1                             ; Definition der Flächenormale der Begren-
                                                  zungsfläche am Satzanfang.
N130 X0 Y0 Z0 A0 C0 T1 D1 F20000
N140 X10 Y0 Z0 G41 CUT3DCCD CDOF2 G64           ; Aktivieren des 3D-Umfangsfräsens unter
                                                  Berücksichtigung der Begrenzungsfläche +
                                                  Ausschalten der Kollisionsüberwachung.
```

```
Programmcode                                Kommentar
N150 X20
N160 X30 A45                                ; Stumpfer Winkel ==> keine Zustellung
N170 X40 A-45                               ; Spitzer Winkel ==> Zustellung
N180 X55
N190 Y10 Z10                                ; Bewegung in Werkzeugrichtung.
N200 Y20
N210 C45                                    ; Reine Orientierungsänderung.
N220 Y30 C90
N230 A5=-1 B5=0 C5=2 Y40                    ; Änderung der Oberfläche.
N240 Y50 G40                                ; Deaktivieren der Werkzeugradiuskorrek-
                                            tur.
...
```

## Weitere Informationen

### Werkzeugtyp

Der Werkzeugtyp (Werkzeugparameter $TC_DP1) wird ausgewertet. Es sind nur Fräswerkzeuge mit zylindrischem Schaft (Zylinder- oder Schaftfräser, Torusfräser und als Grenzfall der zylindrische Gesenkfräser) zugelassen. Das entspricht den Werkzeugtypen 1 - 399 mit Ausnahme der Nummern 111 und 155 bis 157.

### Normwerkzeuge mit Eckenverrundung

Die Eckenverrundung des Normwerkzeugs wird durch den Werkzeugparameter $TC_DP7 beschrieben. Aus dem Werkzeugparameter $TC_DP16 ergibt sich die Abweichung der Eckenverrundung des realen Werkzeugs gegenüber dem Normwerkzeug.

Beispiel: Torusfräser mit verringertem Radius gegenüber dem Normwerkzeug

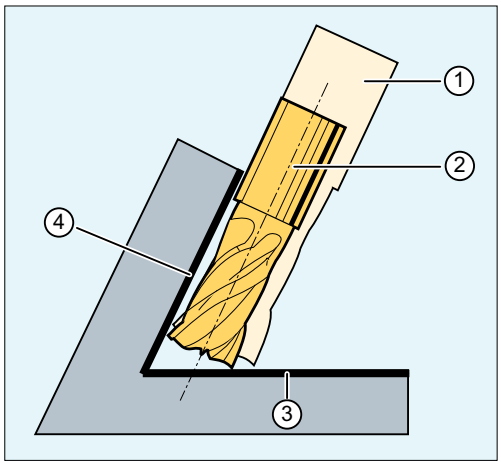
| Werkzeugtyp | Schaftradius (R) | Eckenradius (r) |
|---|---|---|
| Normwerkzeug mit Eckenverrundung | R = $TC_DP6 | r = $TC_DP7 |
| Reales Werkzeug mit Eckenverrundung<br>Werkzeugtypen 121 und 131 Torusfräser<br>(Schaftfräser mit Eckenverrundung) | R' = $TC_DP6 + $TC_DP15 + OFFN | r' = $TC_DP7 + $TC_DP16 |

In diesem Beispiel sind sowohl $TC_DP15 + OFFN als auch $TC_DP16 negativ.

### 3D-WRK mit CUT3DCCD: Werkzeugmittelpunktsbahn mit Zustellung bis zur Begrenzungsfläche

Wird ein Werkzeug verwendet, welches im Vergleich zum passenden Normwerkzeug einen kleineren Radius aufweist, dann wird ein in Längsrichtung zugestellter Fräser soweit weiter geführt, bis dieser den Taschenboden wieder berührt. Damit wird die Ecke, die von der Bearbeitungs- und der Begrenzungsfläche gebildet wird, so weit ausgeräumt, wie dies das Werkzeug zulässt. Es handelt sich dabei um eine gemischte Bearbeitungsweise aus Umfangs-

und Stirnfräsen. Analog zu einem Werkzeug mit verringertem Radius, wird beim Werkzeug mit vergrößerten Radius in die entgegengesetzte Richtung entsprechend zugestellt.




① Normwerkzeug
② Werkzeug mit kleinerem Radius zugestellt bis zur Begrenzungsfläche
③ Begrenzungsfläche
④ Bearbeitungsfläche

Gegenüber allen anderen Werkzeugkorrekturen der G-Gruppe 22 hat ein für CUT3DCCD angegebener Werkzeugparameter $TC_DP6 keine Bedeutung für den Werkzeugradius und beeinflusst die resultierende Korrektur nicht. Der Korrekturoffset ergibt sich aus der Summe des Verschleißwerts des Werkzeugradius (Werkzeugparameter $TC_DP15) und einem zur Berechnung des senkrechten Offsets zur Begrenzungsfläche programmierten Werkzeugoffset OFFN.

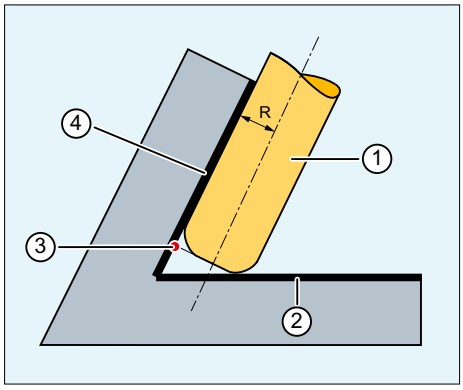
Ob die zu bearbeitende Fläche links oder rechts von der Bahn liegt, kann aus dem erzeugten Teileprogramm nicht entnommen werden. Es wird deshalb von einem positiven Radius und einem negativen Verschleißwert des Originalwerkzeugs ausgegangen. Ein negativer Verschleißwert beschreibt immer ein Werkzeug mit verringertem Durchmesser.

### Verwendung von zylindrischen Werkzeugen

Bei der Verwendung von zylindrischen Werkzeugen ist eine Zustellung nur dann erforderlich, wenn die Bearbeitungsfläche und die Begrenzungsfläche einen spitzen Winkel (kleiner als 90 Grad) bilden. Werden Torusfräser (Schaftfräser mit Eckverrundung) verwendet, dann erfordert dies sowohl bei spitzen als auch bei stumpfen Winkeln eine Zustellung in Längsrichtung des Werkzeugs.

### 3D-WRK mit CUT3DCC: Kontur an der Bearbeitungsfläche

Ist CUT3DCC mit einem Torusfräser aktiv, so bezieht sich die programmierte Bahn auf einen fiktiven Zylinderfräser gleichen Durchmessers. Der hieraus resultierende Bahnbezugspunkt ist bei Verwendung eines Torusfräser im folgenden Bild dargestellt.




① Torusfräser
② Begrenzungsfläche
③ Bahnbezugspunkt
④ Bearbeitungsfläche
R Schaftradius (Werkzeugradius)

Es ist zulässig, dass der Winkel zwischen Bearbeitungs- und Begrenzungsfläche auch innerhalb eines Satzes von einem spitzen in einem stumpfen Winkel oder umgekehrt übergeht.

Gegenüber dem Normwerkzeug darf das verwendete reale Werkzeug sowohl größer als auch kleiner sein. Dabei darf der resultierende Eckenradius nicht negativ werden und das Vorzeichen des resultierenden Werkzeugradius muss erhalten bleiben.

Bei CUT3DCC bezieht sich das NC-Teileprogramm auf die Kontur an der Bearbeitungsfläche. Es wird hierbei wie bei der herkömmlichen Werkzeugradienkorrektur der Gesamtradius herangezogen, der sich aus folgenden Komponenten zusammensetzt:

- Werkzeugradius (Werkzeugparameter $TC_DP6)
- Verschleißwert (Werkzeugparameter $TC_DP15)
- Einem zur Berechnung des senkrechten Offsets zur Begrenzungsfläche programmierten Werkzeugoffset OFFN

Die Lage der Begrenzungsfläche wird aus folgender Differenz bestimmt:

Abmessungen des Normwerkzeugs - Werkzeugradius (WZ-Parameter $TC_DP6)

### Advanced Surface / Top Surface

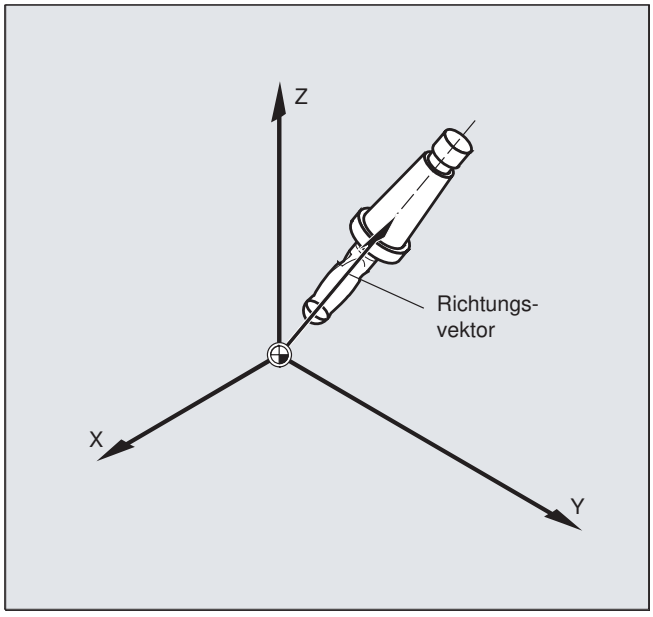
**Hinweis**

Bei Verwendung der Werkzeugradiuskorrektur CUT3DCC / CUT3DCCD in Kombination mit der lizenzpflichtigen Funktion "Advanced Surface" oder "Top Surface" sind die Einstellempfehlungen bezüglich "Advanced Surface" / "Top Surface" zu beachten! Zur Überprüfung der eingestellten Daten steht über das SIOS-Portal ein spezielles Prüfprogramme zur Verfügung.

→ Prüfprogramm für Advanced Surface / Top Surface (https://support.industry.siemens.com/cs/ww/de/view/109738423)

## 3.13.6 Werkzeugorientierung (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST)

Unter Werkzeugorientierung versteht man die geometrische Ausrichtung des Werkzeugs im Raum. Bei einer 5-Achs-Bearbeitungsmaschine ist die Werkzeugorientierung über Programmbefehle einstellbar.




Mit `OSD` und `OST` aktivierte Überschleifbewegungen der Orientierung werden je nach Interpolationsart für die Werkzeugorientierung unterschiedlich gebildet.

Bei aktiver Vektorinterpolation wird der geglättete Orientierungsverlauf auch mittels Vektorinterpolation interpoliert. Dagegen wird bei aktiver Rundachsinterpolation die Orientierung direkt mittels Rundachsbewegungen geglättet.

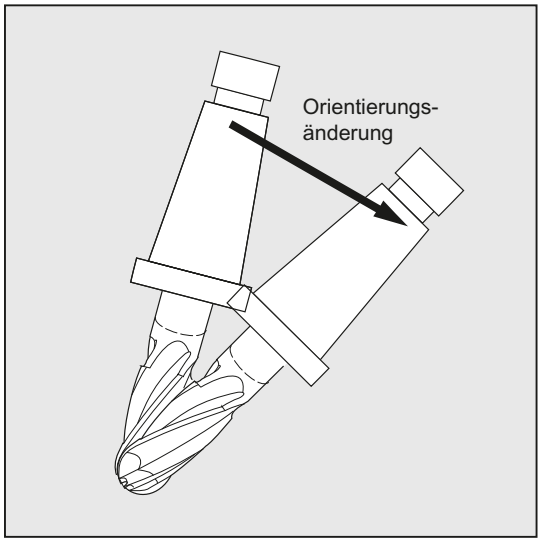
### Programmierung

**Programmierung der Orientierungsänderung:**

Eine Orientierungsänderung des Werkzeugs kann programmiert werden durch:

- direkte Programmierung der Rundachsen `A`, `B`, `C` (Rundachsinterpolation)
- Euler- oder RPY-Winkel
- Richtungsvektor (Vektorinterpolation durch Angabe von `A3` oder `B3` oder `C3`)
- `LEAD/TILT` (Stirnfräsen)

Das Bezugskoordinatensystem ist entweder das Maschinenkoordinatensystem (`ORIMKS`) oder das aktuelle Werkstückkoordinatensystem (`ORIWKS`).




### Programmierung der Werkzeugorientierung:

| | |
|---|---|
| ORIC: | Orientierung und Bahnbewegung parallel |
| ORID: | Orientierung und Bahnbewegung nacheinander |
| OSOF: | keine Orientierungsglättung |
| OSC: | Orientierung konstant |
| OSS: | Orientierungsglättung nur am Satzanfang |
| OSSE: | Orientierungsglättung am Satzanfang und -ende |
| ORIS: | Geschwindigkeit der Orientierungsänderung bei eingeschalteter Orientierungsglättung in Grad pro mm (gilt für OSS und OSSE) |
| OSD: | Überschleifen der Orientierung durch Vorgabe der Überschleiflänge mit dem Settingdatum:<br>SD42674 $SC_ORI_SMOOTH_DIST |
| OST: | Überschleifen der Orientierung durch Vorgabe der Winkeltoleranz in Grad bei Vektorinterpolation mit dem Settingdatum:<br>SD42676 $SC_ORI_SMOOTH_TOL<br>Bei Rundachsinterpolation wird die vorgegebene Toleranz als maximale Abweichung der Orientierungsachsen angenommen. |

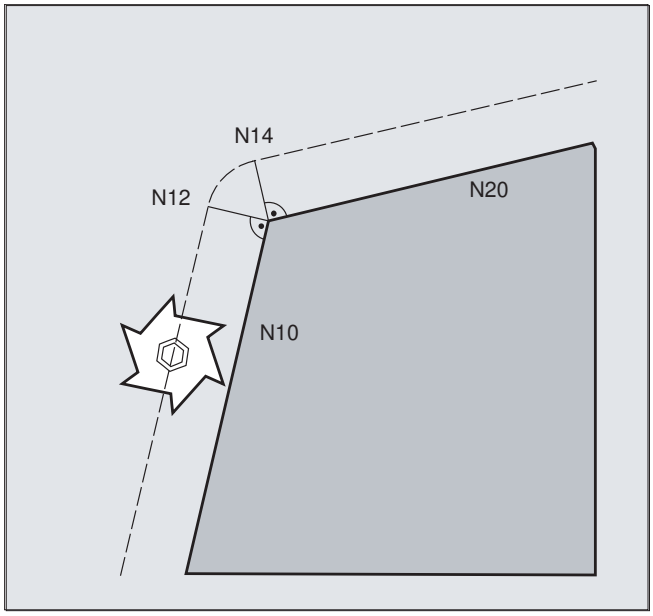
### Hinweis

Alle Befehle zum Überschleifen der Werkzeugorientierung (OSOF, OSC, OSS, OSSE, OSD und OST) sind in der G-Gruppe 34 zusammengefasst. Sie sind modal wirksam, d. h. es kann immer nur einer dieser Befehle wirken.

## Beispiele

### Beispiel 1: ORIC

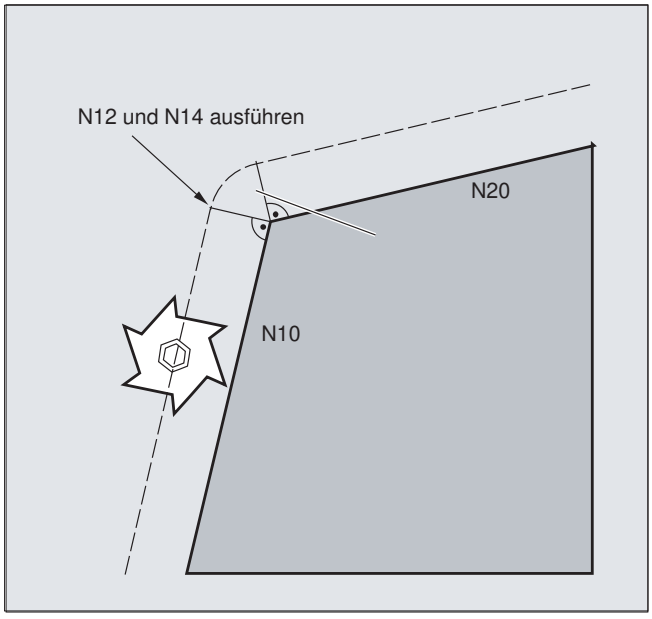
Sind zwischen den Verfahrsätzen `N10` und `N20` zwei oder mehrere Sätze mit Orientierungsänderungen (z. B. `A2=... B2=... C2=...`) programmiert und `ORIC` ist aktiv, so wird der eingefügte Kreissatz entsprechend dem Betrag der Winkeländerungen auf diese Zwischensätze aufgeteilt.




| **Programmcode** | **Kommentar** |
|---|---|
| `ORIC` | |
| `N8 A2=… B2=… C2=…` | |
| `N10 X… Y… Z…` | |
| `N12 C2=… B2=…` | `; Der Kreissatz, der an der Außenecke eingefügt wird,` |
| `N14 C2=… B2=…` | `verteilt sich auf N12 und N14, entsprechend der Ori-` |
| | `entierungsänderung. Kreisbewegung und Orientierungs-` |
| | `änderung werden hierbei parallel ausgeführt.` |
| `N20 X =…Y=… Z=… G1 F200` | |

### Beispiel 2: ORID

Ist `ORID` aktiv, so werden alle Sätze zwischen den beiden Verfahrsätzen am Ende des ersten Verfahrsatzes ausgeführt. Der Kreissatz mit konstanter Orientierung wird unmittelbar vor dem zweiten Verfahrsatz ausgeführt.



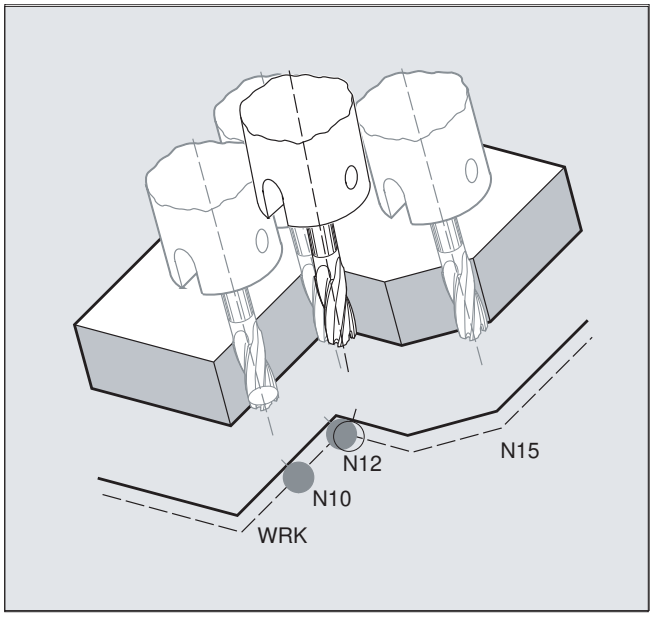

| **Programmcode** | **Kommentar** |
|---|---|
| ORID | |
| N8 A2=… B2=… C2=… | |
| N10 X… Y… Z… | |
| N12 A2=… B2=… C2=… | ; Der Satz N12 und N14 wird am Ende von N10 ausgeführt. Danach wird der Kreissatz mit der aktuellen Orientierung ausgefahren. |
| N14 M20 | ; Hilfsfunktionen etc. |
| N20 X… Y… Z… | |

## Hinweis

Für die Art der Orientierungsänderung an einer Außenecke ist der Programmbefehl maßgebend, welcher im ersten Verfahrsatz einer Außenecke aktiv ist.

**Ohne Orientierungsänderung:** Wird die Orientierung an der Satzgrenze nicht verändert, so ist der Werkzeugquerschnitt ein Kreis, der die beiden Konturen berührt.

### Beispiel 3: Änderung der Orientierung an einer Innenecke



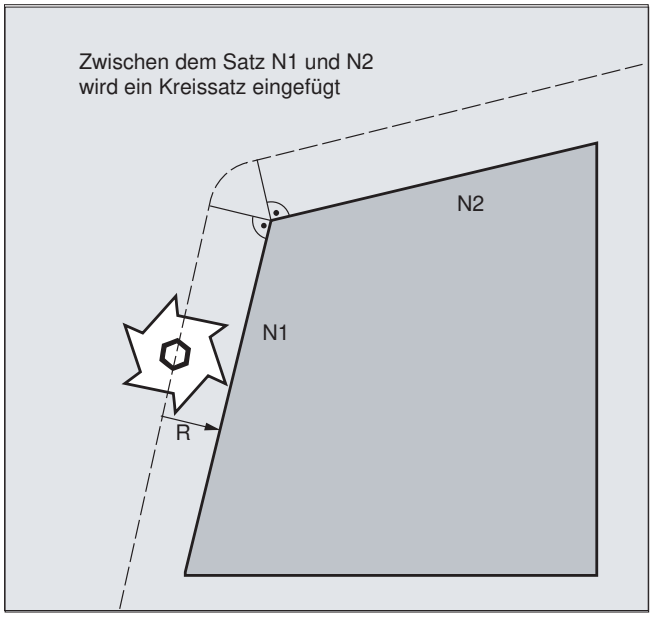

| **Programmcode** |
|---|
| ORIC |
| N10 X …Y… Z… G1 F500 |
| N12 X …Y… Z… A2=… B2=… C2=… |
| N15 X …Y… Z… A2=… B2=… C2=… |

## Weitere Informationen

### Verhalten an Außenecken

An einer Außenecke wird immer ein Kreissatz mit dem Radius des Fräsers eingefügt.

Mit den Programmbefehlen `ORIC` bzw. `ORID` kann festgelegt werden, ob Orientierungsänderungen, die zwischen Satz `N1` und `N2` programmiert wurden, vor Beginn des eingefügten Kreissatzes oder gleichzeitig mit diesem ausgeführt werden.




Ist an Außenecken eine Orientierungsänderung notwendig, so kann diese wahlweise parallel zur Interpolation oder getrennt mit der Bahnbewegung erfolgen.

Bei `ORID` werden zunächst die eingefügten Sätze ohne Bahnbewegung ausgeführt. Der Kreissatz wird unmittelbar vor dem zweiten der beiden Verfahrsätze eingefügt, durch welche die Ecke gebildet wird.

Sind an einer Außenecke mehrere Orientierungssätze eingefügt und `ORIC` ist angewählt, so wird die Kreisbewegung entsprechend den Beträgen der Orientierungsänderungen der einzelnen eingefügten Sätze auf diese verteilt.

### Überschleifen der Orientierung mit OSD bzw. OST

Beim Überschleifen mit `G642` kann die maximale Abweichung für die Konturachsen und die Orientierungsachsen nicht sehr unterschiedlich sein. Die kleinere Toleranz von beiden bestimmt die Form der Überschleifbewegung bzw. Winkeltoleranz, den Orientierungsverlauf relativ stark zu glätten, ohne dabei größere Konturabweichungen hinnehmen zu müssen.

Durch Aktivierung von `OSD` bzw. `OST` ist es möglich, mit einer vorgegebenen Überschleiflänge bzw. Winkeltoleranz sehr geringe Abweichungen des Orientierungsverlaufs ohne gravierende Konturabweichungen "großzügig" zu glätten.

**Hinweis**

Im Unterschied zum Überschleifen der Kontur (und dem Orientierungsverlauf) mit `G642` wird beim Überschleifen der Orientierung mit `OSD` bzw. `OST` kein eigener Satz gebildet, sondern die Überschleifbewegung wird direkt in die programmierten Originalsätze eingefügt.

Mit `OSD` bzw. `OST` können keine Satzübergänge überschliffen werden bei denen ein Wechsel der Interpolationsart für die Werkzeugorientierung (Vektor → Rundachse, Rundachse → Vektor) stattfindet. Diese Satzübergänge können gegebenenfalls mit den herkömmlichen Überschleiffunktionen `G641`, `G642` bzw. `G643` überschliffen werden.

## 3.13.7 Freie D-Nummernvergabe, Schneidennummer

### 3.13.7.1 Freie D-Nummernvergabe, Schneidennummer (Adresse CE)

#### D-Nummer

Die D-Nummern können als Korrekturnummern verwendet werden. Zusätzlich kann über die Adresse CE die Nummer der Schneide adressiert werden. Über die Systemvariable $TC_DPCE kann die Schneidenummer beschrieben werden.

Voreinstellung: Korrekturnr. == Schneidennr.

Über Maschinendaten werden die maximale Anzahl der D-Nummern (Schneidennummern) und die maximale Schneidenanzahl pro Werkzeug festgelegt (→ Maschinenhersteller). Die folgenden Befehle sind nur sinnvoll, wenn die maximale Schneidennummer (MD18105) größer als die Anzahl der Schneiden pro Werkzeug (MD18106) festgelegt wurde. Beachten Sie die Angaben des Maschinenherstellers.

#### Weitere Informationen

Funktionshandbuch Werkzeuge

### 3.13.7.2 Freie D-Nummernvergabe: D-Nummern prüfen (CHKDNO)

Mit dem Befehl `CKKDNO` prüfen Sie, ob die vorhandenen D-Nummern eindeutig vergeben worden sind. Die D-Nummern aller innerhalb einer TO-Einheit definierten Werkzeuge dürfen nur einmal auftreten. Ersatzwerkzeuge werden dabei nicht berücksichtigt.

#### Syntax

```
state=CHKDNO(Tno1,Tno2,Dno)
```

#### Bedeutung

| | | |
|---|---|---|
| `state`: | =TRUE: | Die D-Nummern wurden für den überprüften Bereich eindeutig vergeben. |
| | =FALSE: | Es erfolgte eine D-Nummernkollision oder die Parametrierung ist ungültig. Über Tno1, Tno2 und Dno werden die Parameter übergeben, die zur Kollision führten. Diese Daten können im Teileprogramm ausgewertet werden. |
| `CHKDNO(Tno1,Tno2)`: | Es werden alle D-Nummern der genannten Werkzeuge geprüft. | |
| `CHKDNO(Tno1)`: | Es werden alle D-Nummern von Tno1 gegen alle anderen Werkzeuge geprüft. | |
| `CHKDNO`: | Es werden alle D-Nummern aller Werkzeuge gegen alle anderen Werkzeuge geprüft. | |

### 3.13.7.3 Freie D-Nummernvergabe: D-Nummern umbenennen (GETDNO, SETDNO)

D-Nummern müssen eindeutig vergeben werden. Zwei verschiedene Schneiden eines Werkzeuges können nicht dieselbe D-Nummer haben.

#### GETDNO

Dieser Befehl liefert die D-Nummer einer bestimmten Schneide (ce) eines Werkzeuges mit der T-Nummer t. Existiert keine D-Nummer zu den eingegebenen Parametern, wird d=0 gesetzt. Ist die D-Nummer ungültig wird ein Wert größer 32000 zurückgegeben.

#### SETDNO

Mit diesem Befehl weisen Sie den Wert d der D-Nummer einer Schneide ce des Werkzeuges t zu. Über state wird das Ergebnis dieser Anweisung zurückgegeben (TRUE oder FALSE). Existiert kein Datensatz zu den eingegebenen Parametern wird FALSE zurückgegeben. Syntaxfehler erzeugen einen Alarm. Die D-Nummer kann nicht explizit auf 0 gesetzt werden.

## Syntax

```
d = GETDNO (t,ce)
state = SETDNO (t,ce,d)
```

## Bedeutung

| | |
|---|---|
| d: | D-Nummer der Schneide des Werkzeuges |
| t: | T-Nummer des Werkzeuges |
| ce: | Schneidennummer (CE-Nummer) des Werkzeuges |
| state: | Gibt an, ob der Befehl fehlerfrei ausgeführt werden konnte (TRUE oder FALSE). |

## Beispiel Umbenennen einer D-Nummer

| **Programmierung** | **Kommentar** |
|---|---|
| $TC_DP2[1,2] = 120 | |
| $TC_DP3[1,2] = 5.5 | |
| $TC_DPCE[1,2] = 3 | ; Schneidennummer CE |
| ... | |
| N10 def int DNrAlt, DNrNeu = 17 | |
| N20 DNrAlt = GETDNO(1,3) | |
| N30 SETDNO(1,3,DNrNeu) | |

Damit wird der Schneide CE=3 der neue D-Wert 17 zugewiesen. Jetzt werden die Daten dieser Schneide über die D-Nummer 17 angesprochen; sowohl über die Systemvariablen als auch in der Programmierung mit der NC-Adresse.

### 3.13.7.4 Freie D-Nummernvergabe: T-Nummer zur vorgegebenen D-Nummer ermitteln (GETACTTD)

Die vordefinierte Funktion GETACTTD ermittelt zu einer absoluten D-Nummer die dazugehörige T-Nummer. Es erfolgt keine Prüfung auf Eindeutigkeit. Gibt es mehrere gleiche D-Nummern innerhalb einer TO-Einheit, wird die T-Nummer des ersten gefundenen Werkzeugs zurückgegeben.

#### Syntax

```
<status>=GETACTTD(<TNr>,<DNr>)
```

#### Bedeutung

| | | | |
|---|---|---|---|
| GETACTTD(): | Funktionsaufruf | | |
| <DNr>: | D-Nummer, für die die T-Nummer gesucht werden soll. | | |
| | Datentyp: | INT | |
| <TNr>: | Gefundene T-Nummer | | |
| | Datentyp: | VAR INT | |
| <status>: | Ergebnis | | |
| | Datentyp: | INT | |
| | Wert: | 0 | Die T-Nummer wurde gefunden. <TNr> erhält den Wert der T-Nummer. |
| | | -1 | Zur angegebenen D-Nummer existiert keine T-Nummer, <TNr>=0. |
| | | -2 | Die D-Nummer ist nicht absolut. <TNr> erhält den Wert des ersten gefundenen Werkzeugs, das die D-Nummer mit dem Wert <DNr> enthält. |
| | | -5 | Die Funktion konnte aus einem anderen Grund nicht ausgeführt werden. |

### 3.13.7.5 Freie D-Nummernvergabe: D-Nummern ungültig setzen (DZERO)

Der Befehl DZERO dient zur Unterstützung während dem Umrüsten. So gekennzeichnete Korrekturdatensätze werden nicht mehr vom Befehl CHKDNO geprüft. Um sie wieder zugänglich zu machen, muss die D-Nummer wieder mit SETDNO gesetzt werden.

#### Syntax

```
DZERO
```

#### Bedeutung

| | |
|---|---|
| DZERO: | Kennzeichnet alle D-Nummern der TO-Einheit als ungültig. |

## 3.13.8 Werkzeugträgerkinematik

### Voraussetzungen

Ein Werkzeugträger kann ein Werkzeug nur dann in alle möglichen Raumrichtungen orientieren, wenn

- zwei Drehachsen $V_1$ und $V_2$ vorhanden sind.
- die Drehachsen aufeinander senkrecht stehen.
- die Werkzeuglängsachse senkrecht auf der zweiten Drehachse $V_2$ steht.

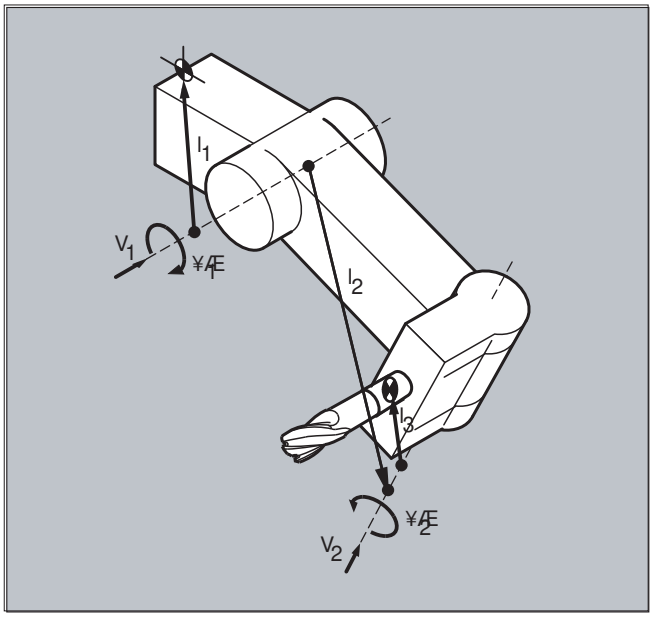
Zusätzlich gilt bei Maschinen, bei denen alle möglichen Orientierungen einstellbar sein müssen, folgende Forderung:

- die Werkzeugorientierung muss senkrecht auf der ersten Drehachse $V_1$ stehen.

### Funktion

Die Werkzeugträgerkinematik mit maximal zwei Drehachsen $v_1$ oder $v_2$ wird über die 17 Systemvariablen $TC_CARR1[m] bis $TC_CARR17[m] beschrieben. Die Beschreibung des Werkzeugträgers besteht aus:

- dem vektoriellen Abstand von der ersten Drehachse zum Bezugspunkt des Werkzeugträgers $I_1$, dem vektoriellen Abstand von erster zu zweiter Drehachse $I_2$, dem vektoriellen Abstand von zweiter Drehachse zum Bezugspunkt des Werkzeugs $I_3$.
- den Richtungsvektoren beider Drehachsen $V_1$, $V_2$.
- den Drehwinkeln α1, α2 um die beiden Achsen. Die Drehwinkel werden mit Blickrichtung in Richtung der Drehachsvektoren im Uhrzeigersinn positiv gezählt.




Für Maschinen mit **aufgelöster Kinematik** (sowohl Werkzeug als auch Werkstück sind drehbar) wurden die Systemvariablen um die Einträge $TC_CARR18[m] bis $TC_CARR23[m] erweitert.

## Parameter

| Funktion der Systemvariablen für orientierbare Werkzeugträger | | | |
|---|---|---|---|
| **Bezeichnung** | **x-Komponente** | **y-Komponente** | **z-Komponente** |
| $l_1$ Offsetvector | $TC_CARR1[m] | $TC_CARR2[m] | $TC_CARR3[m] |
| $l_2$ Offsetvector | $TC_CARR4[m] | $TC_CARR5[m] | $TC_CARR6[m] |
| $v_1$ Drehachse | $TC_CARR7[m] | $TC_CARR8[m] | $TC_CARR9[m] |
| $v_2$ Drehachse | $TC_CARR10[m] | $TC_CARR11[m] | $TC_CARR12[m] |
| $\alpha_1$ Drehwinkel<br>$\alpha_2$ Drehwinkel | $TC_CARR13[m]<br>$TC_CARR14[m] | | |
| $l_3$ Offsetvector | $TC_CARR15[m] | $TC_CARR16[m] | $TC_CARR17[m] |

| Erweiterungen der Systemvariablen für orientierbare Werkzeugträger | | | |
|---|---|---|---|
| **Bezeichnung** | **x-Komponente** | **y-Komponente** | **z-Komponente** |
| $l_4$ Offsetvector | $TC_CARR18[m] | $TC_CARR19[m] | $TC_CARR20[m] |
| **Achsbezeichner**<br>Drehachse $v_1$<br>Drehachse $v_2$ | Achsbezeichner der Drehachsen $v_1$und $v_2$ (Vorbelegung ist Null)<br>$TC_CARR21[m]<br>$TC_CARR22[m] | | |
| **Kinematiktyp** | $TC_CARR23[m] | | |
| **T**ool<br>**P**art<br>**M**ixed mode | Kinematiktyp-T -> | Kinematiktyp-P -> | Kinematiktyp-M |
| | Nur das Werkzeug ist drehbar (Vorbelegung) | Nur das Werkstück ist drehbar | Werkstück & Werkzeug sind drehbar |
| **Offset** der<br>Drehachse $v_1$<br>Drehachse $v_2$ | Winkel in Grad der Drehachsen $v_1$und $v_2$ bei Einnahme der Grundstellung<br>$TC_CARR24[m]<br>$TC_CARR25[m] | | |
| **Winkeloffset** der<br>Drehachse $v_1$<br>Drehachse $v_2$ | Offset der Hirth-Verzahnung in Grad der Drehachsen $v_1$und $v_2$<br>$TC_CARR26[m]<br>$TC_CARR27[m] | | |
| **Winkelinkrem.**<br>$v_1$ Drehachse<br>$v_2$ Drehachse | Inkrement der Hirth-Verzahnung in Grad der Drehachsen $v_1$und $v_2$<br>$TC_CARR28[m]<br>$TC_CARR29[m] | | |
| **Min.-Position**<br>Drehachse $v_1$<br>Drehachse $v_2$ | Software-Limit für Minimalposition der Drehachsen $v_1$und $v_2$<br>$TC_CARR30[m]<br>$TC_CARR31[m] | | |
| **Max.-Position**<br>Drehachse $v_1$<br>Drehachse $v_2$ | Software-Limits für Maximalposition der Drehachsen $v_1$und $v_2$<br>$TC_CARR32[m]<br>$TC_CARR33[m] | | |
| **Werkzeugträger Name** | Anstelle einer Zahl kann ein Werkzeugträger einem Namen bekommen.<br>$TC_CARR34[m] | | |
| **Anwender:**<br>Achsname 1<br>Achsname 2<br>Kennung | Beabsichtigte Verwendung innerhalb der Messzyklen vom Anwender.<br>$TC_CARR35[m]<br>$TC_CARR36[m]<br>$TC_CARR37[m] | | |
| **Position** | $TC_CARR38[m] | $TC_CARR39[m] | $TC_CARR40[m] |
| **Feinver-schiebung** | Parameter, die **zu den Werten in den Basisparameternaddiert** werden können. | | |
| $l_1$ Offsetvector | $TC_CARR41[m] | $TC_CARR42[m] | $TC_CARR43[m] |
| $l_2$ Offsetvector | $TC_CARR44[m] | $TC_CARR45[m] | $TC_CARR46[m] |

| Erweiterungen der Systemvariablen für orientierbare Werkzeugträger | | | |
|---|---|---|---|
| $l_3$ Offsetvector | $TC_CARR55[m] | $TC_CARR56[m] | $TC_CARR57[m] |
| $l_4$ Offsetvector | $TC_CARR58[m] | $TC_CARR59[m] | $TC_CARR60[m] |
| $v_1$ Drehachse | $TC_CARR64[m] | | |
| $v_2$ Drehachse | $TC_CARR65[m] | | |

**Hinweis**

**Erklärungen zu den Parametern**

Mit "m" wird jeweils die Nummer des zu beschreibenden Werkzeugträgers angegeben.

$TC_CARR47 bis $TC_CARR54 sowie $TC_CARR61 bis $TC_CARR63 sind nicht definiert und führen beim Versuch hierauf lesend oder schreiben zuzugreifen, zu einem Alarm.

Die Anfangs- bzw. Endpunkte der Abstandsvektoren auf den Achsen können frei gewählt werden. Die Drehwinke $\alpha_1$, $\alpha_2$ um die beiden Achsen werden im Grundzustand des Werkzeugträgers mit 0° definiert. Die Kinematik eines Werkzeugträgers kann so auf beliebig viele Möglichkeiten beschrieben werden.

Werkzeugträger mit nur einer oder keiner Drehachse können durch Nullsetzen der Richtungsvektoren einer oder beider Drehachsen beschrieben werden.
Bei einem Werkzeugträger ohne Drehachse wirken die Abstandsvektoren wie zusätzliche Werkzeugkorrekturen, deren Komponenten beim Umschalten der Bearbeitungsebenen (G17 bis G19)° nicht beeinflusst werden.

## Erweiterungen der Parameter

### Parameter der Drehachsen

Die Systemvariablen wurden um die Einträge $TC_CARR24[m] bis $TC_CARR33[m] erweitert und wie folgt beschrieben:

| | |
|---|---|
| Den **Offset** der Drehachsen $v_1$, $v_2$ | Veränderung der Position der Drehachse $v_1$ oder $v_2$ bei Grundstellung des orientierbaren Werkzeugträgers. |
| Den **Winkeloffset/ Winkelinkrement** Drehachsen $v_1$, $v_2$ | Offset oder Winkelinkrement der Hirth-Verzahnung der Drehachsen $v_1$ und $v_2$. Programmierter oder berechneter Winkel wird auf den nächstliegenden Wert gerundet, der sich bei ganzzahligem n aus phi = s + n * d ergibt. |
| **Minimal- und Maximalposition** Drehachsen $v_1$, $v_2$ | Der Minimalposition/Maximalposition der Drehachse Grenzwinkel (Software-Limit) der Drehachse v1 und v2. |

### Parameter für den Anwender

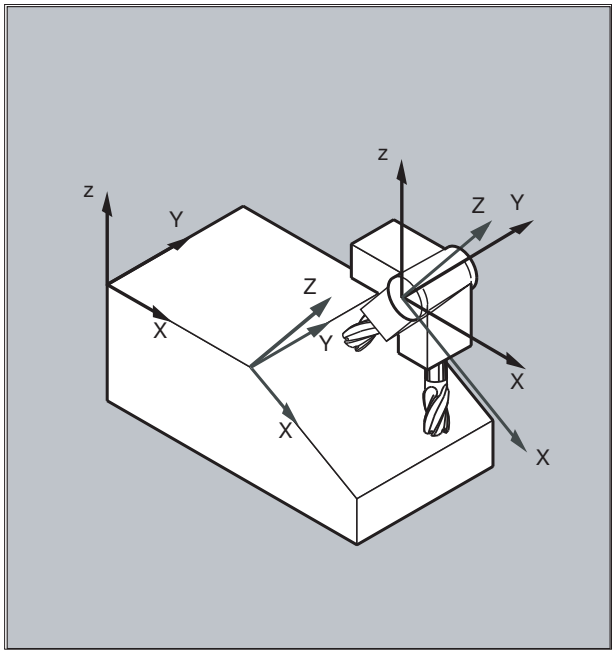
$TC_CARR34 bis $TC_CARR40 enthalten Parameter, die den Anwender zur freien Verfügung stehen und bis zum SW 6.4 standardmäßig innerhalb der NCK nicht weiter ausgewertet werden oder keine Bedeutung haben.

### Parameter der Feinverschiebung

$TC_CARR41 bis $TC_CARR65 enthalten Feinverschiebungsparameter, die zu den Werten in den Basisparametern addiert werden können. Der einem Basisparameter zugeordnete Feinverschiebungswert ergibt sich, wenn zur Parameternummer der Wert 40 addiert wird.

## Beispiel

Der im folgenden Beispiel verwendete Werkzeugträger lässt sich durch eine Drehung um die Y-Achse vollständig beschreiben.




| Programmcode | Kommentar |
|---|---|
| N10 $TC_CARR8[1]=1 | ; Definition der Y-Komponente der ersten Drehachse des Werkzeugträgers 1. |
| N20 $TC_DP1[1,1]=120 | ; Definition eines Schaftfräsers. |
| N30 $TC_DP3[1,1]=20 | ; Definition eines Schaftfräsers mit Länge 20 mm. |
| N40 $TC_DP6[1,1]=5 | ; Definition eines Schaftfräsers mit Radius 5 mm. |
| N50 ROT Y37 | ; Framedefinition mit Drehung von 37° um die Y-Achse. |
| N60 X0 Y0 Z0 F10000 | ; Ausgangsposition anfahren. |
| N70 G42 CUT2DF TCOFR TCARR=1 T1 D1 X10 | ; Radiuskorrektur, Werkzeuglängenkorrektur im gedrehten Frame einstellen, Werkzeugträger 1, Werkzeug 1 anwählen. |
| N80 X40 | ; Bearbeitung unter einer Drehung von 37° durchführen. |
| N90 Y40 | |

| Programmcode | Kommentar |
|---|---|
| N100 X0 | |
| N110 Y0 | |
| N120 M30 | |

## Weitere Informationen

### Aufgelöste Kinematik

Für Maschinen mit aufgelöster Kinematik (sowohl Werkzeug als auch Werkstück sind drehbar) wurden die Systemvariablen um die Einträge `$TC_CARR18[m]` bis `$TC_CARR23[m]` erweitert und wie folgt beschrieben:

Der drehbare Werkzeugtisch bestehend aus:

- dem vektoriellen Abstand der zweiten Drehachse $V_2$ zum Bezugspunkt eines drehbaren Werkzeugtisches $I_4$ der dritten Drehachse.

Die Rundachsen bestehend aus:

- den beiden Kanalbezeichnern für den Bezug der Drehachsen $V_1$und $V_2$, auf deren Position gegebenenfalls bei der Bestimmung der Orientierung des orientierbaren Werkzeugträgers zugegriffen wird.

Der Kinematiktyp mit einem der Werte T, P oder M:

- Kinematiktyp T: Nur das Werkzeug ist drehbar.
- Kinematiktyp P: Nur das Werkstück ist drehbar.
- Kinematiktyp M: Werkzeug und Werkstück sind drehbar.

### Löschen der Werkzeugträgerdaten

Mit `$TC_CARR1[0]=0` können die Daten aller Werkzeugträgerdatensätze gelöscht werden.

Der Kinematiktyp `$TC_CARR23[T]=T` muss mit einen der drei zulässigen Groß- oder Kleinbuchstaben (T,P,M) belegt werden und sollte aus diesen Grund nicht gelöscht werden.

### Ändern der Werkzeugträgerdaten

Jeder der beschriebenen Werte kann durch Zuweisung eines neuen Wertes im Teileprogramm verändert werden. Jedes andere Zeichen als T, P oder M führt bei dem Versuch, den orientierbaren Werkzeugträger zu aktivieren, zu einen Alarm.

### Lesen der Werkzeugträgerdaten

Jeder der beschriebenen Werte kann durch Zuweisung an eine Variable im Teileprogramm gelesen werden.

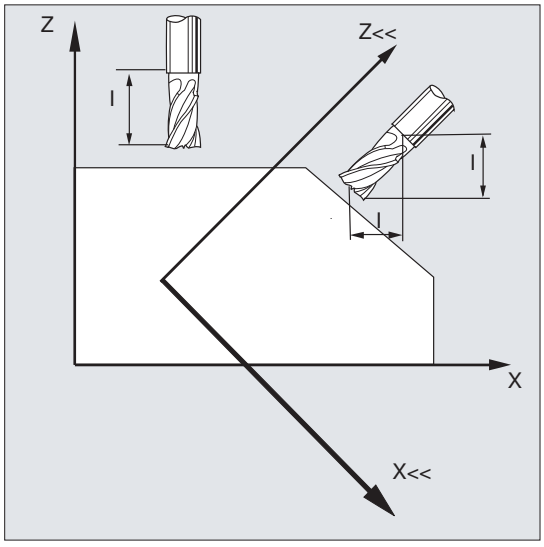
### Feinverschiebungen

Ein unzulässiger Feinverschiebungswert wird erst erkannt, wenn ein orientierbarer Werkzeugträger aktiviert wird, der solch einen Wert enthält und gleichzeitig das Settingdatum SD42974 $SC_TOCARR_FINE_CORRECTION = TRUE ist.

Der Betrag der zulässigen Feinverschiebung wird über Maschinendaten auf einen maximal zulässigen Wert begrenzt.

## 3.13.9 Werkzeuglängenkorrektur für orientierbare Werkzeugträger (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ)

Mit veränderter Raumorientierung des Werkzeugs ändern sich auch dessen Werkzeuglängenkomponenten.




Nach Umrüsten, z. B. durch manuelle Einstellung oder Wechsel des Werkzeugträgers mit fester räumlicher Ausrichtung, müssen daher die Werkzeuglängenkomponenten neu ermittelt werden. Dies erfolgt mit den Wegbefehlen `TCOABS` und `TCOFR`.

Bei einem orientierbaren Werkzeugträger eines aktiven Frames kann bei Werkzeuganwahl mit `TCOFRZ`, `TCOFRY` und `TCOFRX` die Richtung, in die das Werkzeug zeigen soll, bestimmt werden.

### Syntax

```
TCARR=[<m>]
TCOABS
TCOFR
TCOFRZ
TCOFRY
TCOFRX
```

### Bedeutung

| | |
|---|---|
| `TCARR=[<m>]`: | Werkzeugträger mit der Nummer "m" anfordern |
| `TCOABS`: | Werkzeuglängenkomponenten aus der aktuellen Werkzeugträgerorientierung berechnen |
| `TCOFR`: | Werkzeuglängenkomponenten aus der Orientierung des aktiven Frames bestimmen |
| `TCOFRZ`: | Orientierbarer Werkzeugträger aus aktiven Frame, dessen Werkzeug in Z-Richtung zeigt |

| | |
|---|---|
| TCOFRY: | Orientierbarer Werkzeugträger aus aktiven Frame, dessen Werkzeug in Y-Richtung zeigt |
| TCOFRX: | Orientierbarer Werkzeugträger aus aktiven Frame, dessen Werkzeug in X-Richtung zeigt |

## Weitere Informationen

### Werkzeuglängenkorrektur aus Trägerorientierung (TCOABS)

TCOABS berechnet die Werkzeuglängenkorrektur aus den aktuellen Orientierungswinkeln des Werkzeugträgers; abgelegt in den Systemvariablen $TC_CARR13 und $TC_CARR14.

Zur Definition der Werkzeugträgerkinematik mit Systemvariablen siehe
" Werkzeugträgerkinematik (Seite 789) ".

Zur Neuberechnung der Werkzeuglängenkorrektur bei Frame-Wechsel muss das Werkzeug nochmals angewählt werden.

### Werkzeugrichtung aus aktiven Frame

Der orientierbare Werkzeugträger kann so eingestellt werden, dass das Werkzeug in folgende Richtungen zeigt:

- mit TCOFR bzw. TCOFRZ in Z-Richtung
- mit TCOFRY in Y-Richtung
- mit TCOFRX in X-Richtung

Ein Umschalten zwischen TCOFR und TCOABS bewirkt eine Neuberechnung der Werkzeuglängenkorrektur.

### Werkzeugträger anfordern (TCARR)

Mit TCARR werden mit der Werkzeugträgernummer m dessen Geometriedaten angefordert (Korrekturspeicher).

Mit m=0 wird der aktive Werkzeugträger abgewählt.

Die Geometriedaten des Werkzeugträgers werden erst nach Aufruf eines Werkzeugs aktiv. Das angewählte Werkzeug bleibt über den Wechsel eines Werkzeugträgers hinaus aktiv.

Die aktuellen Geometriedaten des Werkzeugträgers können auch im Teileprogramm über die entsprechenden Systemvariablen definiert werden.

### Neuberechnung der Werkzeuglängenkorrektur (TCOABS) bei Frame-Wechsel

Zur Neuberechnung der Werkzeuglängenkorrektur bei Frame-Wechsel muss das Werkzeug nochmals angewählt werden.

**Hinweis**

Die Werkzeugorientierung muss dem aktiven Frame manuell angepasst werden.

Bei der Berechnung der Werkzeuglängenkorrektur werden in einem Zwischenschritt auch die Drehwinkel des Werkzeugträgers berechnet. Da bei Werkzeugträgern mit zwei Drehachsen im Allgemeinen zwei Drehwinkelpaare existieren, mit denen die Werkzeugorientierung dem aktiven Frame angepasst werden kann, müssen die in den Systemvariablen abgelegten

Drehwinkelwerte zumindest annähernd den mechanisch eingestellten Drehwinkeln entsprechen.

**Hinweis**

**Werkzeugorientierung**

Die Steuerung kann die über die Frame-Orientierung berechneten Verdrehwinkel nicht auf die Einstellbarkeit an der Maschine überprüfen.

Sind die Drehachsen des Werkzeugträgers konstruktiv so angeordnet, dass die durch die Frame-Orientierung berechnete Werkzeugorientierung nicht erreicht werden kann, wird ein Alarm ausgegeben.

Die Kombination von Werkzeugfeinkorrektur und den Funktionalitäten zur Werkzeuglängenkorrektur bei beweglichen Werkzeugträgern ist nicht zulässig. Beim Versuch beide Funktionen gleichzeitig aufzurufen, erfolgt eine Fehlermeldung.

Mit `TOFRAME` ist es möglich, einen Frame aufgrund der Orientierungsrichtung des angewählten Werkzeugträgers zu definieren. Genauere Informationen siehe Kapitel "Frames".

Bei aktiver Orientierungstransformation (3-, 4-, 5-Achstransformation) kann ein Werkzeugträger mit von der Null-Lage abweichender Orientierung angewählt werden, ohne dass dabei ein Alarm ausgegeben wird.

**Übergabeparameter von Standard- und Messzyklen**

Für die Übergabeparameter von Standard- und Messzyklen gelten definierte Wertebereiche.

Bei Winkelwerten ist der Wertebereich wie folgt festgelegt:

- Drehung um 1. Geometrieachse: -180 Grad bis +180 Grad
- Drehung um 2. Geometrieachse: -90 Grad bis +90 Grad
- Drehung um 3. Geometrieachse: -180 Grad bis +180 Grad

Siehe Kapitel Frames, "Programmierbare Drehung (ROT, AROT, RPL)".

**Hinweis**

Bei der Übergabe von Winkelwerten an einen Standard- oder Messzyklus ist zu beachten:

**Werte kleiner als die Rechenfeinheit der NC sind auf Null zu runden!**

Die Rechenfeinheit der NC für Winkelpositionen ist festgelegt im Maschinendatum:

MD10210 $MN_INT_INCR_PER_DEG

## 3.13.10 Orientierbare Werkzeugträger nach Maschinenvermessung modifizieren (CORRTC)

Mit der Funktion `CORRTC` können gemessene kinematische Kettenelemente eines Toolcarriers in spezielle Korrekturelemente geschrieben werden.

**Hinweis**

Die mit der Funktion CORRTC eingeschriebenen Korrekturwerte sind nicht sofort im ToolCarrier wirksam. Erst nach Abwahl des ToolCarriers, NEWCONF und Anwahl des ToolCarriers sind die Korrekturwerte wirksam.

## Syntax

```
<_Corr_Status> = CORRTC(<_Corr_Vect>, <_Corr_Index>, <_Corr_Mode>,
[ <_No_Alarm>])
```

## Bedeutung

| | | | |
|---|---|---|---|
| CORRTC: | Funktionsaufruf | | |
| <_Corr_Status>: | Rückgabewert der Funktion | | |
| | Datentyp: | INT | |
| | Werte: | 0 | Die Funktion wurde ohne Fehler ausgeführt. |
| | | 1 | Es ist kein Toolcarrier aktiv. |
| | | 2 | Der aktive Toolcarrier wurde nicht mit kinematischen Ketten definiert. |
| | | 10 | Der Aufrufparameter <_Corr_Index> ist negativ. |
| | | 11 | Der Aufrufparameter <_Corr_Mode> ist negativ. |
| | | 12 | Ungültiger Verweis auf den Abschnitt einer Teilkette (_CORR_INDEX). |
| | | 13 | In dem Abschnitt, auf den mit dem Parameter _CORR_INDEX verwiesen wird, ist kein Korrekturelement definiert ($TC_CARR_CORR_ELEM). |
| | | 20 | Die 100er-Stelle von _CORR_MODE ist ungültig. Es sind nur die Werte 0 und 1 zulässig. |
| | | 21 | Die 1000er-Stelle von _CORR_MODE ist ungültig. Es sind nur die Werte 0 und 1 zulässig. |
| | | 30 | Bei der Korrektur eines Offsetvektors ist die Abweichung gegenüber dem aktuellen Wert in mindestens einer Koordinate größer als der durch das Settingdatum SD41612 $SN_CORR_TOCARR_LIN_MAX vorgegebene Maximalwert. |
| | | 31 | Der Versuch, eine Systemvariable zu beschreiben, wurde wegen fehlender Schreibrechte abgewiesen. |
| <_Corr_Vect>: | Korrekturvektor<br>Der Inhalt des Korrekturvektors wird durch die nachfolgenden Parameter <Corr_Index> und <_Corr_Mode> definiert.<br>Ist <_Corr_Status> = 30, wird der Inhalt des Vektors überschrieben (s. o.). | | |
| | Datentyp: | REAL | |
| <_Corr_Index>: | Bezeichnet den Abschnitt, für den der Richtungsvektor des Korrekturelements korrigiert werden soll. | | |
| | Datentyp: | INT | |

| | | | |
|---|---|---|---|
| <_Corr_Mode>: | Korrekturmodus | | |
| | Datentyp: | INT | |
| | Der Parameter <Corr_Mode> ist dezimalcodiert (1er- bis 1000er-Stelle): | | |
| | 1er-Stelle: | Reserviert | |
| | 10er-Stelle: | Bestimmt, wie das Korrekturelement, auf das der Inhalt von <_Corr_Index> verweist, modifiziert werden soll. | |
| | | xx**0**x | Der Korrekturvektor wird unmittelbar in das Korrekturelement geschrieben.<br>Diese Variante kann dazu verwendet werden, das Korrekturelement unmittelbar zu beschreiben, ohne dass der Index <n> des betreffenden Systemdatums ($NK_OFF_DIR[<n>, ...]) bekannt sein muss. |
| | | xx**1**x | Wie 0, jedoch mit dem Unterschied, dass der übergebene Korrekturwert in Weltkoordinaten interpretiert wird.<br>Ein Unterschied zwischen den Varianten 0 und 1 kann immer dann entstehen, wenn die kinematische Kette in Grundstellung (Positionen aller Rundachsen gleich 0) weitere Drehungen enthält. |
| | | xx**2**x | Wie 1, jedoch mit dem Unterschied, dass sich der Korrekturwert auf den gesamten Abschnitt bezieht, d. h., in das Korrekturelement wird ein solcher Wert eingetragen, dass der gesamte Abschnitt die durch den Korrekturwert definierte Länge erreicht. |
| | 100er-Stelle: | Bestimmt, wie der Inhalt des Parameters <_Corr_Vect> zu interpretieren ist. | |
| | | x**0**xx | Der übergebene Korrekturvektor <_Corr_Vect> enthält die gesamte neue Länge des Korrekturelements bzw. des Abschnitts, auf den <_Corr_Index> in Verbindung mit der 10er-Stelle von <_Corr_Mode> verweist (absolute Korrektur). |
| | | x**1**xx | Der übergebene Korrekturvektor <_Corr_Vect> enthält nur die Differenz gegenüber der aktuellen Länge des Korrekturelements bzw. des Abschnitts, auf den <_Corr_Index> in Verbindung mit der 10er-Stelle von <_Corr_Mode> verweist (inkrementelle Korrektur). |
| | 1000er-Stelle: | Bestimmt, ob die maximal zulässige Korrektur durch das Settingdatum $SN_CORR_TOCARR_LIN_MAX begrenzt werden soll oder nicht. | |
| | | **0**xxx | Die Grenzwertüberwachung ist aktiv. |
| | | **1**xxx | Die Grenzwertüberwachung wird unterdrückt. |
| <_No_Alarm>: | Verhalten im Fehlerfall (Rückgabewert > 0) (**optional**) | | |
| | Datentyp: | BOOL | |
| | Wert: | FALSE (Default) | Im Fehlerfall wird die Programmabarbeitung angehalten und ein Alarm wird ausgegeben. |
| | | TRUE | Im Fehlerfall wird die Programmabarbeitung nicht angehalten und es wird kein Alarm angezeigt.<br>Anwendungsfall: Anwenderspezifische Reaktion entsprechend Rückgabewert |

## Weitere Informationen zu CORRTC

Die kinematische Struktur eines Toolcarriers wird durch eine (Typ T und Typ P) oder zwei (Typ M) kinematische Ketten (Teilketten) beschrieben, die vom dazugehörigen Bezugspunkt Maschinenbezugspunkt oder Werkzeugträgerbezugspunkt) ausgehen. Eine der beiden Ketten, die Werkzeug-Kette, endet am Bezugspunkt des Werkzeugs, die andere, die Werkstück-Kette im Nullpunkt des Basiskoordinatensystems.

Die Funktion CORRTC schreibt Achsrichtungen bei Maschinen mit ToolCarriern in spezielle Korrekturelemente. Eine kinematische Kette wird unter anderem mit Elementen vom Typ OFFSET beschrieben, die über $NK_TYPE definiert werden.

### CORRTC arbeitet mit Abschnitten

Die beiden Teilketten können sich jeweils in maximal vier Abschnitte aufteilen:

- Abschnitt 1 beginnt am Startpunkt der Kette und endet an der ersten Rundsachse.
- Abschnitt 2 ist der Abschnitt zwischen Rundachse 1 und Rundachse 2.
- Abschnitt 3 ist der Abschnitt zwischen Rundachse 2 und dem Ende der Kette.

Die folgende Grafik zeigt einen orientierbaren Werkzeugträger mit 2 Rundachsen.

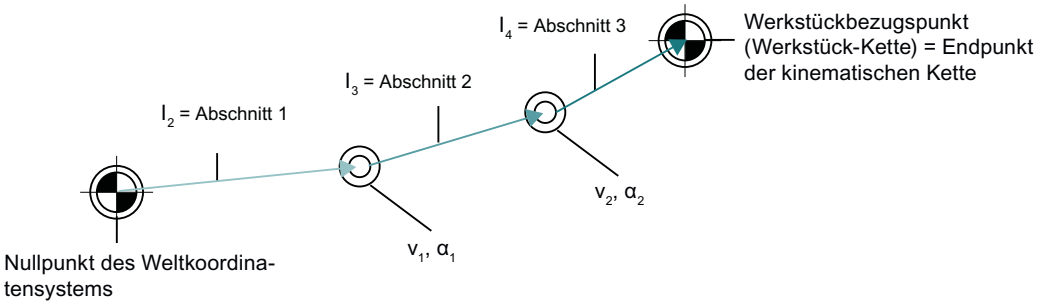



Bild 3-11 CORRTC Beispiel

Die Abschnitte sind eindeutig definiert: Durchläuft man die kinematische Teil-Kette von ihrem Startpunkt bis zum Endpunkt, so hat der erste Abschnitt den Index 1, der nächste den Index 2 usw.

### Korrekturelemente

Auf jeweils ein konstantes kinematisches Kettenelement (Kettenelement des Typs $NK_TYPE[<n>] = "OFFSET") in jedem dieser Abschnitte kann mit der Systemvariablen $TC_CARR_CORR_ELEM [, 0 ... 3] verwiesen werden. In diese so bezeichneten Elemente werden mithilfe der Funktion CORRTC die Korrekturwerte geschrieben, die bei der Maschinenvermessung ermittelt wurden.

Die Reihenfolge der Verweise in $TC_CARR_CORR_ELEM[m, 0..3] muss mit den oben beschriebenen Abschnitten korrespondieren, d.h. in $TC_CARR_CORR_ELEM[m, 0] kann nur ein Kettenelement stehen, das zu dem Offsetvektor l1 gehört, usw..

Der Bezugswert ist dabei immer der entsprechende Wert, der in dem beim Aufruf von CORRTC aktiven Toolcarrier wirksam ist. Nach der Anwahl des Toolcarriers veränderte Inhalte der

Kinematikdaten in der Datenhaltung haben keinen Einfluss auf die Wirkungsweise der Funktion CORRTC.

## 3.13.11 Online-Werkzeuglängenkorrektur (TOFFON, TOFFOF)

Über die Systemvariable $AA_TOFF[<n> ] können die effektiven Werkzeuglängen entsprechend der drei Werkzeugrichtungen dreidimensional in Echtzeit überlagert werden.

Als Index <n> werden die drei Geometrieachsbezeichner verwendet. Damit ist die Anzahl der aktiven Korrekturrichtungen durch die zur selben Zeit aktiven Geometrieachsen festgelegt.

Alle Korrekturen können gleichzeitig aktiv sein.

Die Funktion Online-Werkzeuglängenkorrektur ist anwendbar bei:

- Orientierungstransformation TRAORI
- Orientierbare Werkzeugträger TCARR

**Hinweis**

Die Online-Werkzeuglängenkorrektur ist eine **Option,** die vorher frei geschaltet werden muss. Nur in Verbindung mit einer aktiven Orientierungstransformation oder einem aktiven orientierbaren Werkzeugträger ist diese Funktion sinnvoll.

### Syntax

```
TRAORI
TOFFON(<Korrekturrichtung>[,<Offsetwert>])
WHEN TRUE DO $AA_TOFF[<Korrekturrichtung>]              ; In Synchronaktionen.
...
TOFFOF(<Korrekturrichtung>)
```

### Bedeutung

| | | |
|---|---|---|
| TOFFON: | Online-Werkzeuglängenkorrektur **aktivieren** | |
| | <Korrekturrichtung>: | Werkzeugrichtung (X, Y, Z), in der die Online-Werkzeuglängenkorrektur wirksam sein soll. |
| | <Offsetwert>: | Bei der Aktivierung kann für die entsprechende Korrekturrichtung ein Offsetwert angegeben werden, der sofort herausgefahren wird. |
| TOFFOF: | Online-Werkzeuglängenkorrektur **zurücksetzen**<br>Die Korrekturwerte in der angegebenen Korrekturrichtung werden zurückgesetzt und es wird ein Vorlaufstopp ausgelöst. | |

## Beispiele

### Beispiel 1: Anwahl der Werkzeuglängenkorrektur

| **Programmcode** | **Kommentar** |
|---|---|
| MD21190 $MC_TOFF_MODE =1 | ; Absolute Werte werden angefahren. |
| MD21194 $MC_TOFF_VELO[0] =1000 | |
| MD21196 $MC_TOFF_VELO[1] =1000 | |
| MD21194 $MC_TOFF_VELO[2] =1000 | |
| MD21196 $MC_TOFF_ACCEL[0] =1 | |
| MD21196 $MC_TOFF_ACCEL[1] =1 | |
| MD21196 $MC_TOFF_ACCEL[2] =1 | |
| N5 DEF REAL XOFFSET | |
| N10 TRAORI(1) | ; Transformation ein. |
| N20 TOFFON(Z) | ; Aktivierung der Online-WZL-Korrektur für die Z-Werkzeugrichtung. |
| N30 WHEN TRUE DO $AA_TOFF[Z]=10 G4 F5 | ; Für die Z-Werkzeugrichtung wird eine WZL-Korrektur von 10 interpoliert. |
| ... | |
| N100 XOFFSET=$AA_TOFF_VAL[X] | ; Aktuelle Korrektur in X-Richtung zuweisen. |
| N120 TOFFON(X,-XOFFSET) G4 F5 | ; Für die X-Werkzeugrichtung wird die WZL-Korrektur wieder zu 0 zurückgefahren. |

### Beispiel 2: Abwahl der Werkzeuglängenkorrektur

| **Programmcode** | **Kommentar** |
|---|---|
| N10 TRAORI(1) | ; Transformation ein. |
| N20 TOFFON(X) | ; Aktivierung der Online-WZL-Korrektur für die X-Werkzeugrichtung. |
| N30 WHEN TRUE DO $AA_TOFF[X]=10 G4 F5 | ; Für die X-Werkzeugrichtung wird eine WZL-Korrektur von 10 interpoliert. |
| ... | |
| N80 TOFFOF(X) | ; Positionsoffset der X-Werkzeugrichtung wird gelöscht:<br>...$AA_TOFF[X]=0<br>Es wird keine Achse verfahren.<br>Zur aktuellen Position im WKS wird der Positionsoffset entsprechend der aktuellen<br>Orientierung hinzugerechnet. |

## Weitere Informationen

### Satzaufbereitung

Bei der Satzaufbereitung im Vorlauf wird der im Hauptlauf wirksame aktuelle Werkzeuglängenoffset mit berücksichtigt. Um die maximal zulässigen Achsgeschwindigkeiten weitgehend ausnutzen zu können, ist es erforderlich, die Satzaufbereitung mit einem Vorlaufstopp STOPRE anzuhalten, während ein Werkzeugoffset aufgebaut wird.

Der Werkzeugoffset ist zum Vorlaufzeitpunkt auch immer dann bekannt, wenn die Werkzeuglängenkorrekturen nach Programmstart nicht mehr verändert werden, oder wenn nach einer Veränderung der Werkzeuglängenkorrekturen mehr Sätze abgearbeitet wurden als der IPO-Buffer zwischen Vorlauf und Hauptlauf aufnehmen kann.

### Variable $AA_TOFF_PREP_DIFF

Das Maß für die Differenz zwischen der aktuellen im Interpolator wirksamen Korrektur und der Korrektur, die zum Zeitpunkt der Satzaufbereitung wirksam war, kann in der Variablen $AA_TOFF_PREP_DIFF[<n>] abgefragt werden.

### Maschinendaten und Settingdaten einstellen

Für die Online-Werkzeuglängenkorrektur stehen folgende Systemdaten zur Verfügung:

- MD20610 $MC_ADD_MOVE_ACCEL_RESERVE (Beschleunigungsreserve für überlagerte Bewegung)
- MD21190 $MC_TOFF_MODE
  Inhalt der Systemvariable $AA_TOFF[<n>] wird als absoluter Wert herausgefahren oder aufintegriert.
- MD21194 $MC_TOFF_VELO (Geschwindigkeit der Online-Werkzeuglängenkorrektur)
- MD21196 $MC_TOFF_ACCEL (Beschleunigung der Online-Werkzeuglängenkorrektur)
- Settingdatum zur Vorgabe von Grenzwerten:
  SD42970 $SC_TOFF_LIMIT (Obergrenze des Werkzeuglängenkorrekturwertes)

## 3.13.12 Modifikation der Korrekturdaten bei drehbaren Werkzeugen

### 3.13.12.1 Orientierungen berechnen (ORISOLH)

Die vordefinierte Funktion ORISOLH unterstützt den Anwender dabei, die Rundachspositionen einer Maschine so einzustellen, dass ein Drehwerkzeug in eine definierte, kinematikunabhängige Lage relativ zum Werkstück gebracht wird. Voraussetzung ist, dass eine 6-Achstransformation aktiv ist, die mit kinematischen Ketten parametriert ist.

Es stehen zwei Grundfunktionen zur Verfügung:

- Werkzeug ausrichten
  Die beiden Winkel β und γ werden vorgegeben. Die Funktion berechnet die dazu nötigen Winkel der drei Orientierungsachsen.
- Werkzeug ausrichten direkt
  Die Winkel der zweiten und dritten Orientierungsachse werden vorgegeben. Die Funktion berechnet daraus die zugehörigen Winkel β und γ, sowie den Winkel der noch fehlenden ersten Orientierungsachse.

**Hinweis**

**Reihenfolge der Orientierungsachsen**

Durchläuft man die kinematische Kette, die den Aufbau der Maschine beschreibt, vom Werkstück bis zum Werkzeug, dann gelten für die Reihenfolge der drei Orientierungsachsen einer 6-Achstransformation folgende Festlegungen:

- Die Orientierungsachse, die am nächsten am Werk**stück** liegt, ist die **erste** Orientierungsachse.
- Die Orientierungsachse, die am nächsten am Werk**zeug** liegt, ist die **dritte** Orientierungsachse.

Die erste Orientierungsachse wird in der Regel eine Spindel sein, und die entsprechende Drehung wird deshalb in diesen Fällen durch einen drehenden Frame realisiert.

## Syntax

```
<RetVal> = ORISOLH(<Cntrl>,<W1>,<W2>)
```

## Bedeutung

| | | | |
|---|---|---|---|
| ORISOLH: | Funktionsaufruf | | |
| <RetVal>: | Rückgabewert der Funktion | | |
| | Datentyp: | INT | |
| | Wertebereich: | 0, -2, -3, ..., -17 | |
| | Werte: | 0 | Funktion wurde ohne Fehler beendet. |
| | | -2 | Es ist keine gültige Transformation (6-Achs-Orientierungstransformation) aktiv. |
| | | -3 | Der erste Parameter (<Cntrl>) ist negativ. |
| | | -4 | Die 1er-Stelle des ersten Parameters (<Cntrl>) ist ungültig.<br>Es sind nur die Werte 0 und 1 zulässig. |
| | | -5 | Die 10er-Stelle des ersten Parameters (<Cntrl>) ist ungültig.<br>Es sind nur die Werte 0 bis 3 zulässig. |
| | | -6 | Die 100er-Stelle des ersten Parameters (<Cntrl>) ist ungültig.<br>Es sind nur die Werte 0 und 1 zulässig. |
| | | -7 | Die 1000er-Stelle des ersten Parameters (<Cntrl>) ist ungültig.<br>Es sind nur die Werte 0 bis 3 zulässig. |
| | | -8 | Bei der Funktion "Werkzeug ausrichten direkt" ist der Winkel γ zu groß. |
| | | -9 | Bei der Funktion "Werkzeug ausrichten direkt" verletzt mindestens eine der vorgegebenen Achspositionen eine Achsgrenze. |
| | | -10 | Es ist kein Werkzeug aktiv. |
| | | -11 | Die verlangte Orientierung ist nicht einstellbar. |
| | | -12 | Die Anpassung der freien Achswinkel bei Hirthverzahnung ist für die erste oder einzige Lösung nicht möglich. |
| | | -13 | Die Anpassung der freien Achswinkel bei Hirthverzahnung ist für die zweite Lösung nicht möglich. |
| | | -14 | Die Anpassung der freien Achswinkel bei Hirthverzahnung ist für keine der beiden Lösungen möglich. |
| | | -15 | Die erste Orientierungsachse ist als Hirthachse parametriert. |
| | | -16 | Sowohl die zweite als auch die dritte Rundachse ist als Hirthachse parametriert. Es darf nur maximal eine der beiden Achsen Hirthachse sein. |
| | | -17 | Bei der Funktion "Schwenken direkt" ist mindestens eine der vorgegebenen Achspositionen nicht mit der zugehörigen Hirthverzahnung kompatibel. |

| | | | |
|---|---|---|---|
| <Cntrl>: | Steuert das Verhalten der Funktion | | |
| | Datentyp: | INT | |
| | Der Parameter <Cntrl> ist dezimalcodiert (1er- bis 1000er-Stelle): | | |
| | 1er-Stelle: | Die 1er-Stelle steuert das Verhalten bei Fehlern. | |
| | | xxx**0** | Im Fehlerfall (Rückgabewert < 0) wird die Programmbearbeitung abgebrochen und der Alarm 14106 ausgegeben.<br>**Hinweis:**<br>Der Alarm wird unabhängig vom Wert der 1er-Stelle auch dann ausgegeben, wenn der Parameter <Cntrl> negativ ist. |
| | | xxx**1** | Im Fehlerfall (Rückgabewert < 0) wird **kein** Alarm ausgegeben. Der Anwender hat im Programm die Möglichkeit, selbst geeignet zu reagieren. |
| | 10er-Stelle: | Steuert das Verhalten bei Vorhandensein einer Orientierungsachse mit Hirthverzahnung.<br>**Hinweis:**<br>Dieser Parameter wird nur bei der Funktion "Werkzeug ausrichten" ausgewertet (d. h., wenn die 100er-Stelle den Wert "0" hat). | |
| | | xx**0**x | Die Achsposition wird auf die nächstgelegene Position gerundet. |
| | | xx**1**x | Die Achspositionen werden so gerundet, dass die Abweichung des Winkels β von seinem programmierten Wert minimal wird. |
| | | xx**2**x | Die Achspositionen werden so gerundet, dass der Winkel β gleich dem größten möglichen Wert wird, der kleiner als der programmierte Wert ist (β wird abgerundet). |
| | | xx**3**x | Die Achspositionen werden so gerundet, dass der Winkel β gleich dem kleinsten möglichen Wert wird, der größer als der programmierte Wert ist (β wird aufgerundet). |
| | 100er-Stelle: | Gibt an, welche Funktion ausgeführt werden soll, bzw. welche Bedeutung die beiden nachfolgenden Parameter <W1> und <W2> haben. | |
| | | x**0**xx | Funktion "Werkzeug ausrichten"<br>Parameter <W1> und <W2> haben folgende Bedeutung:<br>• <W1> = β<br>• <W2> = γ<br>Es werden die zugehörigen Winkel der Orientierungsachsen berechnet. |
| | | x**1**xx | Funktion "Werkzeug ausrichten direkt"<br><W1> ist die Positionsvorgabe für die zweite Orientierungsachse, <W2> die Positionsvorgabe für die dritte Orientierungsachse einer 6-Achstransformation. Es werden die Position der ersten Orientierungsachse und die Winkel β und γ bestimmt, die mit den beiden Positionsvorgaben kompatibel sind.<br>Falls kein Fehler auftritt, werden in der Systemvariablen $P_ORI_POS[<n>, <m>] immer zwei Lösungen ausgegeben. Dabei verweist der erste Index <n> (0 oder 1) auf die Lösung und der zweite Index <m> (0 ... 2) auf die Orientierungsachse: |

| | | | |
|---|---|---|---|
| | | | • $P_ORI_POS[0/1, **0**]: Position der ersten Orientierungsachse<br>• $P_ORI_POS[0/1, **1**]: Winkel β<br>• $P_ORI_POS[0/1, **2**]: Winkel γ<br>Es wird überprüft, ob die Positionsvorgaben <W1> und <W2> mit eventuell aktiven Hirthverzahnungen oder aktiven Softwarelimits kompatibel sind. Ist das nicht der Fall, wird eine entsprechende Fehlernummer zurückgegeben (siehe Parameter <RetVal>).<br>Bei einer beliebigen Wahl der beiden Winkel <W1> und <W2> liegt die Schneide des Werkzeugs im Allgemeinen nicht in der Bearbeitungsebene. Der Winkel γ, um den die Schneide aus der Bearbeitungsebene gedreht ist, darf nicht größer sein als der Grenzwert, der durch das Settingdatum SD42999 $SC_ORISOLH_INCLINE_TOL bestimmt wird. |
| | 1000er-Stelle: | Gibt an, wie die Positionen der Lösungen gegebenenfalls modifiziert werden, wenn die 100er-Stelle den Wert "0" hat, d. h. bei der Funktion "Werkzeug ausrichten". | |
| | | **0**xxx | Die berechneten Achspositionen sollen möglichst nahe bei den aktuellen Maschinenachspositionen liegen. |
| | | **1**xxx | Die berechneten Achspositionen sollen bei Moduloachsen möglichst nahe bei der Mitte des Modulobereichs liegen, bei anderen Achsen möglichst nahe bei 0. Für Nicht-Moduloachsen bedeutet das, dass die Achspositionen auf den Bereich -180° … +180° reduziert werden. |
| | | **2**xxx | Die berechneten Achspositionen sollen unabhängig vom Achstyp auf den Bereich -180° … +180° reduziert werden. |
| <W1>: | Erster Winkel<br>Die Bedeutung ergibt sich aus der 100er-Stelle des Parameters <Cntrl>. | | |
| | Datentyp: | REAL | |
| <W1>: | Zweiter Winkel<br>Die Bedeutung ergibt sich aus der 100er-Stelle des Parameters <Cntrl>. | | |
| | Datentyp: | REAL | |

**Hinweis**

Nicht programmierte Parameter haben den Standardwert "0".

## Weitere Informationen

Die Anzahl gefundener Lösungen zusammen mit weiteren Statusinformationen beim Ausführen der Funktion ORISOLH können über folgende Systemvariablen gelesen werden:

| Systemvariable | Bedeutung | | |
|---|---|---|---|
| $P_ORI_POS [<n>, <m>] | Liefert die Winkel der Orientierungsachsen, die sich bei Orientierungsprogrammierung ergeben. | | |
| | <n>: | Index der Lösung | |
| | | Wertebereich: | 0, 1 |
| | <m>: | Index der Orientierungsachse | |
| | | Wertebereich: | 0 ... 2 |
| | | | Die Reihenfolge der Orientierungsachsen (1 ... 3) bezieht sich auf die Definition der Achsen in $NT_ROT_AX_NAME. |
| | Beim Aufruf der Funktion ORISOLH im Modus "Werkzeug ausrichten direkt" enthalten die Variablen $P_ORI_POS[0/1, 1] und P_ORI_POS[0/1, 2] die zu den beiden Lösungen gehörenden Werte der beiden Winkel β und γ.<br>Die erste in $P_ORI_POS[<n>, <m>] eingetragene Lösung, d. h. mit dem Index <n> = 0, ist immer die, die von der Steuerung gewählt wird, wenn man die verlangte Orientierung direkt anfährt. Der zweite Index <m> bezieht sich dabei auf die Orientierungsachse, d. h. auf $NT_ROT_AX_NAME.<br>Die in $P_ORI_POS[<n>, <m>] eingetragenen Achspositionen berücksichtigen die Offsets, die in $NK_OFF und $NK_OFF_FINE eingetragen sind, d. h., diese Achswinkel können in nachfolgenden Sätzen ohne weitere Modifikation dazu verwendet werden, die gewünschte Orientierung einzustellen.<br>Ist eine Rundachse eine Hirthachse, werden die Lösungspositionen auf die nächstgelegene Raststellung der Hirthverzahnung gerundet. In der Systemvariablen $P_ORI_DIFF können für Hirth-verzahnte Rundachsen die Differenzen der Achspositionen für die exakten Lösungen und die an die Hirth-Rasterung angepassten Lösungen gelesen werden. | | |
| $P_ORI_DIFF [<n>, <m>] | Liefert die Differenz zwischen den exakten und den in $P_ORI_POS zur Verfügung gestellten Positionen der Orientierungsachsen, die sich bei Orientierungsprogrammierung ergeben. | | |
| | <n>: | Index der Lösung | |
| | | Wertebereich: | 0, 1 |
| | <m>: | Index der Orientierungsachse | |
| | | Wertebereich: | 0 ... 2 |
| | | | Die Reihenfolge der Orientierungsachsen (1 ... 3) bezieht sich auf die Definition der Achsen in $NT_ROT_AX_NAME. |
| | Der Inhalt kann nur dann ungleich Null sein, wenn die Positionen gerastert werden (Hirth-Verzahnung), d. h. wenn das Systemdatum $NT_HIRTH_INCR der betreffenden Achse ungleich Null ist und wenn diese Achse eine manuelle Rundachse ist. | | |

| Systemvariable | Bedeutung | | |
|---|---|---|---|
| $P_ORI_SOL | Werden bei einer Orientierungstransformation mit mehr als einer Orientierungsachse die Achswinkel berechnet, die zu einer vorgegebenen Orientierung führen sollen, gibt es im Allgemeinen mehr als eine Lösung. In der Systemvariablen $P_ORI_SOL ist die Anzahl der gültigen Lösungen zusammen mit zusätzlichen Statusinformationen enthalten.<br>Der Inhalt von $P_ORI_SOL ist folgendermaßen codiert: | | |
| | Werte < 0 | Allgemeine Fehlerzustände | |
| | | -1 | Für die aktive Transformation wurden noch keine Lösungen berechnet (fehlender Aufruf von ORISOLH). |
| | | -2 | Es ist keine Transformation aktiv, oder die aktive Transformation ist keine Orientierungstransformation (6-Achs-Transformation), die Positionen zu einer vorgegebenen Orientierungsprogrammierung liefern kann. |
| | | -4 | Die gewünschte Orientierung ist mit der gegebenen Kinematik nicht einstellbar. |
| | | -5 | Beim Aufruf der Funktion ORISOLH im Modus "Werkzeug ausrichten direkt" wurde keine Lösung gefunden. |
| | | -6 | Beim Aufruf der Funktion ORISOLH im Modus "Werkzeug ausrichten direkt" ist der Winkel γ zu groß. |
| | | -7 | Beim Aufruf der Funktion ORISOLH im Modus "Werkzeug ausrichten direkt" wurde ein Winkel vorgegeben, der aufgrund der Hirthverzahnung nicht einstellbar ist. |
| | | -8 | Die erste Orientierungsachse (Frameachse) darf nicht als Hirthachse parametriert sein. |
| | | -9 | Sowohl die zweite als auch die dritte Rundachse ist als Hirthachse parametriert. Es darf nur max. eine der beiden Achsen Hirthachse sein. |
| | | -10 | Es wurde keine Anpassung der Lösung(en) an die Hirthachsverzahnung gefunden. |
| | Werte > 0<br>1er-Stelle | Anzahl der mathematisch möglichen Lösungen ohne Berücksichtigung von Achsgrenzen und evtl. Fehlerbedingungen. | |
| | | 0 | Es existiert keine Lösung, d. h., die verlangte Orientierung ist nicht einstellbar.<br>Dieser Fall kann drei verschiedene Ursachen haben:<br>• Die verlangte Orientierung ist aufgrund der Maschinenkinematik (nicht rechtwinkelig angeordnete Orientierungsachsen) auch bei beliebigem Verfahrbereich der Orientierungsachsen prinzipiell nicht erreichbar. In diesem Fall sind die 10er- und 100er-Stelle von $P_ORI_SOL beide null, die den Orientierungsachsen zugeordneten Statusvariablen $P_ORI_STAT enthalten den Wert "-4".<br>• Die berechneten Lösungen sind nicht erreichbar, weil diese die Achsbegrenzungen verletzen würden. Die Positionen der Orientierungsachsen, die sich ohne Achsbegrenzungen ergeben würden, können in $P_ORI_POS gelesen werden.<br>• Beim Aufruf der Funktion ORISOLH im Modus "Werkzeug ausrichten direkt" wurden Achspositionen derart vorgegeben, dass entweder der Orientierungsvektor oder der Ori- |

| Systemvariable | Bedeutung | | |
|---|---|---|---|
| | | | entierungsnormalenvektor des Werkzeugs parallel zur ersten Orientierungsachse, deren Position berechnet werden soll, ausgerichtet ist. Die Position dieser Achse ist in diesen Fällen nicht definiert. |
| | | 1 | Es existiert eine Lösung.<br>Dieser Fall kann drei verschiedene Ursachen haben:<br>• Es gibt aufgrund der vorgegebenen Orientierung und der Maschinenkinematik auch ohne Berücksichtigung der Achsgrenzen nur eine Lösung (mathematisch gesehen handelt es sich um zwei zusammenfallende Lösungen). Dieser Fall tritt bei nicht rechtwinkeligen Kinematiken am Rand des Orientierungsbereichs auf. $P_ORI_POS enthält beide (identischen) Lösungen.<br>• Es gibt nur eine Lösung, weil eine zweite Lösung aufgrund verletzter Achsgrenzen ungültig ist. Die gültige Lösung ist dann immer die erste Lösung in $P_ORI_POS. Die zweite Lösung, die sich ohne Berücksichtigung der Achsgrenzen ergeben würde, ist in $P_ORI_POS ebenfalls lesbar.<br>• Beim Aufruf der Funktion ORISOLH im Modus "Werkzeug ausrichten direkt" ist das der Normalfall. Zu den vorgegebenen Achspositionen zweier Orientierungsachsen gibt es im Allgemeinen nur eine gültige Position der zu berechnenden fehlenden Orientierungsachse. |
| | | 2 | Es existieren zwei Lösungen. |
| | | 8 | Es existieren unendlich viele Lösungen d. h. die Position einer Orientierungsachse (der Polachse) ist beliebig. Von den beiden möglichen Stellungen der anderen Achse(n) ist jedoch eine aufgrund verletzter Achsgrenzen ausgeschlossen. |
| | | 9 | Es existieren unendlich viele Lösungen, d. h. die Position einer Orientierungsachse (der Polachse) ist unbestimmt. Die unbestimmte Achse kann aus der 100er-Stelle oder aus der Systemvariablen $P_ORI_STAT ermittelt werden. |
| | Werte > 0<br>10er-Stelle | Bitcodierte Anzeige für verletzte Achsgrenzen. Die genaue Fehlerursache kann aus der Systemvariablen $P_ORI_STAT ermittelt werden. | |
| | | Bit 0 (Wert 10): | Für mindestens eine Lösung ist mindestens eine Achsgrenze der 1. Orientierungsachse verletzt. |
| | | Bit 1 (Wert 20): | Für mindestens eine Lösung ist mindestens eine Achsgrenze der 2. Orientierungsachse verletzt. |
| | | Bit 2 (Wert 40): | Für mindestens eine Lösung ist mindestens eine Achsgrenze der 3. Orientierungsachse verletzt. |
| | Werte > 0<br>100er-Stelle | Bitcodierte Anzeige für nicht-definierte Achspositionen (kann nur auftreten, wenn es unendlich viele Lösungen gibt, d. h., wenn die 1er-Stelle gleich "9" ist). | |
| | | Bit 0 (Wert 100): | Die Position der 1. Orientierungsachse ist nicht definiert. |
| | | Bit 1 (Wert 200): | Die Position der 2. Orientierungsachse ist nicht definiert. |

| Systemvariable | Bedeutung | | |
|---|---|---|---|
| | | Bit 2 (Wert 400): | Die Position der 3. Orientierungsachse ist nicht definiert. |
| | Die Bezeichnungen 1., 2. und 3. Orientierungsachse beziehen sich auf die Definition der Achsen in $NT_ROT_AX_NAME. | | |

| Systemvariable | Bedeutung | | |
|---|---|---|---|
| $P_ORI_STAT [<n>] | Liefert für jede der maximal 3 Orientierungsachsen den Status nach Aufruf von ORISOLH. | | |
| | <n>: | Index der Orientierungsachse<br>(entspricht dem Index der betreffenden Orientierungsachse in $NT_ROT_AX_NAME) | |
| | | Wertebereich: | 0 ... 2 |
| | | | Die Reihenfolge der Orientierungsachsen (1 ... 3) bezieht sich auf die Definition der Achsen in $NT_ROT_AX_NAME. |
| | Der Inhalt von $P_ORI_STAT ist folgendermaßen codiert: | | |
| | Werte < 0 | Allgemeine Fehlerzustände | |
| | | -1 | Der Status ist nicht definiert (fehlender Aufruf von ORISOLH). |
| | | -2 | Es ist keine Transformation aktiv, oder die aktive Transformation ist keine Orientierungstransformation (6-Achs-Transformation), die Positionen zu einer vorgegebenen Orientierungsprogrammierung liefern kann. |
| | | -3 | Die Achse ist in der aktiven Transformation nicht enthalten. |
| | | -4 | Die Position der Achse kann nicht berechnet werden, weil mit der gegebenen Kinematik die verlangte Orientierung auch bei beliebig angenommenem Verfahrbereich der Achse nicht erreichbar ist. |
| | | -5 | Beim Aufruf der Funktion ORISOLH im Modus "Werkzeug ausrichten direkt" wurden Achspositionen derart vorgegeben, dass entweder der Orientierungsvektor oder der Orientierungsnormalenvektor des Werkzeugs parallel zur ersten Orientierungsachse, deren Position berechnet werden soll, ausgerichtet ist. Die Position dieser Achse ist in diesen Fällen nicht definiert. |
| | | -6 | Beim Aufruf der Funktion ORISOLH im Modus "Werkzeug ausrichten direkt" ist der Winkel γ zu groß. |
| | | -7 | Beim Aufruf der Funktion ORISOLH im Modus "Werkzeug ausrichten direkt" wurde ein Winkel vorgegeben, der aufgrund der Hirthverzahnung nicht einstellbar ist. |
| | | -8 | Die erste Orientierungsachse (Frameachse) darf nicht als Hirthachse parametriert sein. |
| | | -9 | Sowohl die zweite als auch die dritte Rundachse ist als Hirthachse parametriert. Es darf nur max. eine der beiden Achsen Hirthachse sein. |
| | | -10 | Es wurde keine Anpassung der Lösung(en) an die Hirthachsverzahnung gefunden. |
| | Werte > 0 | Bitcodierte Anzeige für verletzte Achsgrenzen der ersten Lösung. | |
| | 1er-Stelle | Bit 0 (Wert 1): | Die erste Lösung verletzt die untere Achsgrenze. |
| | | Bit 1 (Wert 2): | Die erste Lösung verletzt die obere Achsgrenze. |
| | Werte > 0 | Bitcodierte Anzeige für verletzte Achsgrenzen der zweiten Lösung. | |
| | 10er-Stelle | Bit 0 (Wert 10): | Die zweite Lösung verletzt die untere Achsgrenze. |
| | | Bit 1 (Wert 20): | Die zweite Lösung verletzt die obere Achsgrenze. |

| Systemvariable | Bedeutung | | |
|---|---|---|---|
| | Werte > 0 | Anzeige einer nicht-definierten Achsposition. | |
| | 100er-Stelle | Bit 0 (Wert 100): | Die Position der Orientierungsachse ist nicht definiert, d. h., die verlangte Orientierung wird mit jeder beliebigen Einstellung der Rundachse erreicht (Polstellung). Diese Information ist auch in der Systemvariablen $P_ORI_SOL enthalten. |
| | Von den Fehlernummern, die eine Verletzung der Achsgrenzen anzeigen, können mehrere gleichzeitig auftreten. Da bei der Verletzung einer Achsgrenze versucht wird, durch Addition bzw. Subtraktion von Vielfachen von 360° eine Position innerhalb der erlaubten Achsgrenzen zu erreichen, ist – falls dies nicht möglich ist – nicht eindeutig definiert, ob die untere oder die obere Achsgrenze verletzt wurde.<br>Gibt es für die verlangte Orientierung keine Lösung ($P_ORI_SOL = 0), ist der Status der in der Transformation enthaltenen Orientierungsachsen "0". | | |

**Hinweis**

**$NT_ROT_AX_NAME**

Mit dieser Systemvariablen wird auf maximal 3 Achsen verwiesen, die zur Einstellung der Orientierung dienen. Sie enthält die Namen der Kettenglieder ($NK_NAME), die diejenigen Maschinenachsen (Rundachsen) definieren, die die Orientierungsbewegungen ausführen müssen, die sich aus einer kinematischen Transformation ergeben. Die Reihenfolge, in der die maximal drei Rundachsen in dieser Systemvariablen enthalten sind, ist für die Maschinenkinematik bedeutungslos, da diese aus der Struktur der kinematischen Ketten abgeleitet wird. Da sie aber die Reihenfolge definiert, mit der andere Variablen auf die Rundachsen zugreifen, muss die Reihenfolge der Orientierungsachsen in $NT_ROT_AX_NAME mit der Kinematikbeschreibung übereinstimmen.

**Hinweis**

**Statusinformationen**

Die Statusinformationen, die z. B. anzeigen, dass eine Orientierung nicht erreichbar ist oder nur unter Verletzung der relevanten Achsgrenzen erreichbar wäre, führen nicht zu einem NC-Alarm. Es liegt in der Verantwortung des Anwenders, auf die genannten Bedingungen geeignet zu reagieren.

### 3.13.12.2 Modifikation der Korrekturdaten bei drehbaren Werkzeugen aktivieren (CUTMOD, CUTMODK)

Die Modifikation der Korrekturdaten bei drehbaren Werkzeugen wird im NC-Programm über den Sprachbefehl CUTMOD (in Verbindung mit orientierbaren Werkzeugträgern) bzw. CUTMODK (für Orientierungstransformationen, die mittels kinematischer Ketten definiert wurden) aktiviert.

**Hinweis**

Da orientierbare Werkzugträger und Orientierungstransformationen, die mittels kinematischer Ketten definiert wurden, nicht gleichzeitig aktiv sein können, kommt es nicht zu Konflikten zwischen beiden Varianten.

## Syntax

```
CUTMOD = <Value>
```

bzw.

```
CUTMODK = <Command>
```

## Bedeutung

| | | | |
|---|---|---|---|
| CUTMOD: | Funktionsaufruf in Verbindung mit orientierbaren Werkzeugträgern | | |
| <Value>: | Zugewiesener Wert | | |
| | Datentyp: | INT | |
| | Wert: | 0 | Die Funktion ist deaktiviert.<br>Die von den Systemvariablen $P_AD... gelieferten Werte sind gleich den korrespondierenden Werkzeugparametern. |
| | | > 0 | Die Funktion wird aktiviert, falls ein orientierbarer Werkzeugträger mit der angegebenen Nummer aktiv ist, d. h., die Aktivierung ist an einen bestimmten orientierbaren Werkzeugträger gebunden.<br>Die von den Systemvariablen $P_AD... gelieferten Werte sind gegenüber den korrespondierenden Werkzeugparametern abhängig von der aktiven Drehung gegebenenfalls modifiziert.<br>Die Deaktivierung des bezeichneten orientierbaren Werkzeugträgers deaktiviert die Funktion temporär, die Aktivierung eines anderen orientierbaren Werkzeugträgers deaktiviert sie permanent. Im ersten Fall wird die Funktion deshalb bei erneuter Anwahl des gleichen orientierbaren Werkzeugträgers wieder aktiviert, im zweiten Fall ist eine erneute Anwahl notwendig, auch dann, wenn zu einem späteren Zeitpunkt der orientierbare Werkzeugträger mit der angegebenen Nummer erneut aktiviert wird.<br>Die Funktion wird durch Reset nicht beeinflusst. |
| | | -1 | Die Funktion wird immer aktiviert, falls ein orientierbarer Werkzeugträger aktiv ist.<br>Beim Wechsel des Werkzeugträgers oder bei dessen Abwahl und einer späteren erneuten Anwahl muss CUTMOD nicht erneut gesetzt werden. |
| | | -2 | Die Funktion wird immer aktiviert, falls ein orientierbarer Werkzeugträger aktiv ist, dessen Nummer gleich der des aktuell aktiven orientierbaren Werkzeugträgers ist.<br>Ist kein orientierbarer Werkzeugträger aktiv, ist das gleichbedeutend mit `CUTMOD=0`.<br>Ist ein orientierbarer Werkzeugträger aktiv, ist das gleichbedeutend mit der unmittelbaren Angabe der aktuellen Werkzeugträgernummer. |
| | | < -2 | Werte kleiner 2 werden ignoriert, d. h., dieser Fall wird so behandelt, als wäre CUTMOD nicht programmiert.<br>**Hinweis:**<br>Dieser Wertebereich sollte nicht verwendet werden, weil er für eventuelle spätere Erweiterungen reserviert ist. |
| CUTMODK: | Funktionsaufruf in Verbindung mit Orientierungstransformationen, die mittels kinematischer Ketten definiert wurden | | |

| | | | |
|---|---|---|---|
| <Command>: | Zugewiesenes Kommando | | |
| | Datentyp: | STRING | |
| | Wert: | "NEW" | Es werden die für die "Modifikation der Korrekturdaten" relevanten Zustände einer aktiven mit kinematischen Ketten definierten Transformation, der Name der Transformation und der aktuelle Konturframe abgespeichert.<br>**Hinweis:**<br>Dieses Kommando ist nur zulässig, wenn eine geeignete Transformation (TRAORI_DYN, TRAORI_STAT oder TRAANG_K) aktiv ist. |
| | | "OFF" | Schaltet die aktive "Modifikation der Korrekturdaten" aus. Die zuvor mit "NEW“ abgelegten Daten bleiben erhalten.<br>**Hinweis:**<br>Dieses Kommando ist auch zulässig, wenn CUTMODK nicht aktiv ist. Es bleibt dann ohne Auswirkung. Ein eventuell vorhandener Datensatz zur "Modifikation der Korrekturdaten" bleibt erhalten. |
| | | "ON" | Mit diesem Kommando wird die "Modifikation der Korrekturdaten" mit einem zuvor mit dem Kommando "NEW" gespeicherten Datensatz erneut aktiviert.<br>Ist bei Ausführung dieses Kommandos eine Transformation mit dem Namen des gespeicherten Datensatzes aktiv, wird die "Modifikation der Korrekturdaten" sofort wirksam. Andernfalls wird die Aktivierung verzögert, bis eine aktive Transformation aktiviert wird. |
| | | "CLEAR" | Schaltet wie das Kommando "OFF" die "Modifikation der Korrekturdaten" aus und löscht zusätzlich den abgespeicherten Datensatz.<br>**Hinweis:**<br>Dieses Kommando ist auch zulässig, wenn CUTMODK nicht aktiv ist. |

**Hinweis**

**SD42984 $SC_CUTDIRMOD**

Die über den Befehl CUTMOD bzw. CUTMODK aktivierbare Funktion ersetzt die über das Settingdatum SD42984 $SC_CUTDIRMOD aktivierbare Funktion. Diese Funktion steht jedoch weiterhin unverändert zur Verfügung. Da es aber nicht sinnvoll ist, beide Funktionen parallel zu nutzen, kann sie nur aktiviert werden, wenn CUTMOD gleich Null und CUTMODK gleich dem Nullstring ist.

## Weitere Informationen

### Modifizierte Korrekturdaten lesen

Die modifizierten Korrekturdaten werden in den folgenden Systemvariablen und BTSS-Variablen zur Verfügung gestellt:

| Bedeutung | Systemvariable | BTSS-Variable |
|---|---|---|
| Schneidenlage | $P_AD[2] | cuttEdgeParam2 |
| Halterwinkel | $P_AD[10] | cuttEdgeParam10 |
| Schnittrichtung | $P_AD[11] | cuttEdgeParam11 |
| Freiwinkel | $P_AD[24] | cuttEdgeParam24 |

Die Daten sind gegenüber den korrespondierenden Werkzeugparametern ($TC_DP2[..., ...], ...) immer dann modifiziert, wenn die Funktion "Modifikation der Korrekturdaten bei drehbaren Werkzeugen" mit dem Befehl `CUTMOD` bzw. `CUTMODK` aktiviert wurde und das Werkzeug durch einen orientierbaren Werkzeugträger bzw. eine geeignete Orientierungstransformation gedreht wurde.

### Weitere funktionsrelevante Systemvariablen

| Systemvariable | Bedeutung |
|---|---|
| $P_CUTMOD_ANG / $AC_CUTMOD_ANG | Liefert den Winkel, um den ein Werkzeug in der aktiven Bearbeitungsebene gedreht wurde und der der Ermittlung modifizierter Schneidendaten bei den Funktionen CUTMOD bzw. CUTMODK zugrunde liegt. |
| $P_CUTMOD / $AC_CUTMOD | Liest den aktuell gültigen Wert, der zuletzt mit dem Befehl CUTMOD programmiert wurde (Nummer des Werkzeugträgers, für den die Modifikation der Korrekturdaten aktiviert werden soll).<br>War der letzte programmierte Wert `CUTMOD=-2` (Aktivierung mit dem aktuell aktiven orientierbaren Werkzeugträger), wird in der Systemvariablen nicht der Wert "-2", sondern die Nummer des zum Zeitpunkt der Programmierung aktiven orientierbaren Werkzeugträgers zurückgeliefert. |
| $P_CUTMODK / $AC_CUTMODK | Liest den Namen der Transformation, unter dem der aktuell gültige Datensatz zur "Modifikation der Korrekturdaten" angelegt wurde. |

| Systemvariable | Bedeutung |
|---|---|
| $P_CUT_INV /<br>$AC_CUT_INV | Liefert den Wert TRUE, wenn das Werkzeug so gedreht ist, dass die Spindeldrehrichtung invertiert werden muss. Dazu müssen in dem Satz, auf den sich die jeweilige Leseoperation bezieht, die folgenden vier Bedingungen erfüllt sein:<br>1. Es ist ein Dreh- oder Schleifwerkzeug aktiv (Werkzeugtypen 400 bis 599 und / oder SD42950 $SC_TOOL_LENGTH_TYPE = 2).<br>2. Die Modifikation der Korrekturdaten wurde mit dem Befehl `CUTMOD` bzw. `CUTMODK` aktiviert.<br>3. Es ist ein orientierbarer Werkzeugträger bzw. eine mit kinematischen Ketten definierte Orientierungstransformation aktiv, der/die durch den Befehl CUTMOD bzw. CUTMODK selektiert wurde.<br>4. Das Werkzeug wird durch den orientierbaren Werkzeugträger bzw. durch die kinematische Orientierungstransformation so gedreht, dass die resultierende Normale der Werkzeugschneide gegenüber der Ausgangslage um mehr als 90° (typischerweise 180°) gedreht ist.<br>Ist mindestens eine der genannten vier Bedingungen nicht erfüllt, liefert die Variable den Wert FALSE. Für Werkzeuge, deren Schneidenlage nicht definiert ist, ist der Wert der Variablen immer FALSE. |

| Systemvariable | Bedeutung | |
|---|---|---|
| $P_CUTMOD_ERR | Fehlerzustand nach dem letzten Aufruf der CUTMOD-Funktion<br>Die CUTMOD-Funktion kann auch implizit bei Werkzeugwechsel aufgerufen werden. Die Variable wird bei Reset auf null zurückgesetzt. Sie wird bei jedem Werkzeugwechsel zunächst zurückgesetzt und gegebenenfalls neu beschrieben.<br>Die Variable ist Bit-codiert. Die Bits haben folgenden Bedeutungen: | |
| | Bit 0: | Für das aktive Werkzeug ist keine gültige Schnittrichtung definiert. |
| | Bit 1: | Die Schneidenwinkel (Freiwinkel und Halterwinkel) des aktiven Werkzeugs sind beide null. |
| | Bit 2: | Der Freiwinkel des aktiven Werkzeugs hat einen unzulässigen Wert (< 0° oder > 180°). |
| | Bit 3: | Der Halterwinkel des aktiven Werkzeugs hat einen unzulässigen Wert (< 0° oder > 90°). |
| | Bit 4: | Der Plattenwinkel des aktiven Werkzeugs hat einen unzulässigen Wert (< 0° oder > 90°). |
| | Bit 5: | Die Kombination Schneidenlage - Halterwinkel des aktiven Werkzeugs ist unzulässig (bei den Schneidenlagen 1 bis 4 muss der Halterwinkel ≤ 90° sein, bei den Schneidenlagen 5 bis 8 muss er ≥ 90° sein). |
| | Bit 6: | Unzulässige Drehung des aktiven Werkzeugs.<br>Das Werkzeug wurde um ± 90° (mit einer Toleranz von etwa 1°) aus der aktiven Bearbeitungsebene herausgedreht. Dadurch ist die Schneidenlage in der Bearbeitungsebene nicht mehr definiert. |
| | Bit 7: | Die Schneidplatte liegt nicht in der Bearbeitungsebene, und der Winkel zwischen Schneidplatte und Bearbeitungsebene überschreitet die mit dem Settingdatum SD42998 $SC_CUTMOD_PLANE_TOL vorgegebene Obergrenze. |
| | Bit 8: | Die Schneidplatte liegt nicht in der Bearbeitungsebene. Der Betrag des Winkels α ist größer als 1°. Der Winkel α ist der Drehwinkel um die Koordinatenachse, die sowohl auf der Drehachse des Winkels β als auch auf der Drehachse des Winkels γ senkrecht steht (bei G18 die X-Achse). |

$P_...: Vorlaufvariablen

$AC_...: Hauptlaufvariablen

Alle Hauptlaufvariablen können in Synchronaktionen gelesen werden. Ein Lesezugriff aus dem Vorlauf generiert einen Vorlaufstopp.

### Ebenenwechsel

Für die Bestimmung von modifizierter Schneidenlage, Schnittrichtung und Halter- bzw. Freiwinkel ist die Betrachtung der Schneide in der jeweils aktiven Ebene (G17 - G19) maßgebend.

Enthält jedoch das Settingdatum SD42940 $SC_TOOL_LENGTH_CONST (Wechsel der Werkzeuglängenkomponenten bei Ebenenwechsel) einen gültigen Wert ungleich Null (plus oder minus 17, 18 oder 19), so bestimmt dessen Inhalt die Ebene, in der die relevanten Größen betrachtet werden.

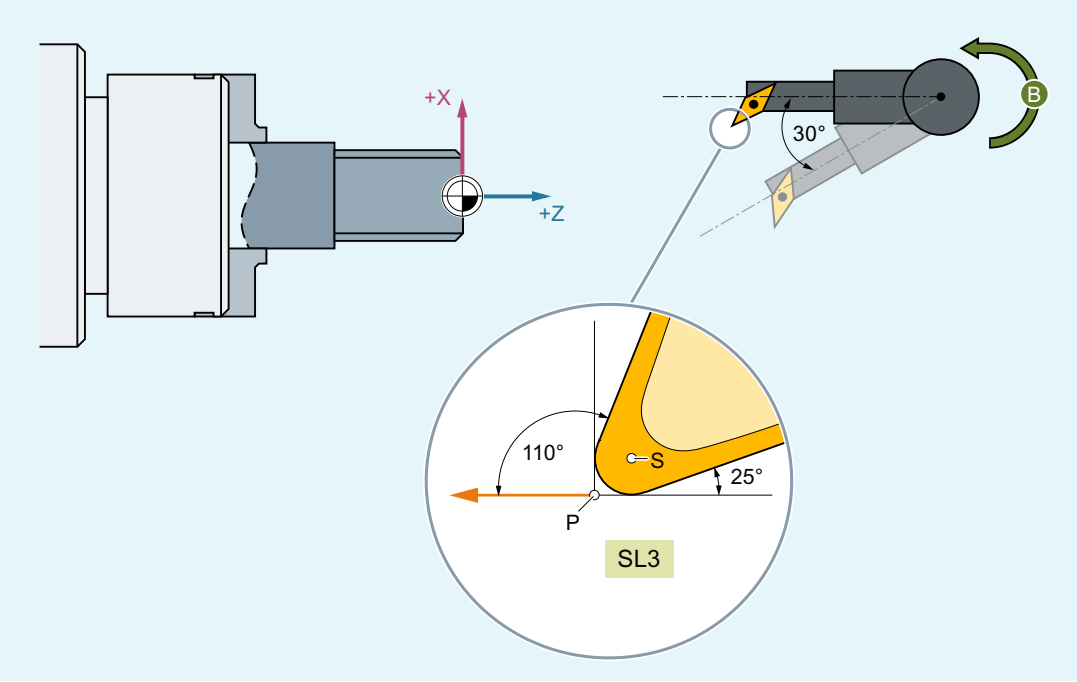
Diese Vorrangregel der Settingdaten über den G-Code kann durch Setzten von Bit 18 des Maschinendatums $MC_TOOL_PARAMETER_DEF_MASK außer Kraft gesetzt werden. D. h., wenn dieses Bit gesetzt ist, gilt immer die mit dem G-Befehl der Gruppe 6 definierte Ebene.

### Wirksamkeit der modifizierten Schneidendaten

Die modifizierte Schneidenlage und der modifizierte Schneidenbezugspunkt werden bei Programmierung auch für ein bereits aktives Werkzeug sofort wirksam. Eine Werkzeugneuanwahl ist dazu nicht notwendig.

## Beispiel

Bei einem Werkzeug mit der Schneidenlage 3 und einem orientierbaren Werkzeugträger, der das Werkzeug um die B-Achse drehen kann, soll mit Hilfe des CUTMOD-Befehls die Schneidenlage nach einer Werkzeugdrehung modifiziert werden.




S: Schneidenmittelpunkt
P: Schneidenbezugspunkt
SL: Schneidenlage

| Programmcode | Kommentar |
|---|---|
| N10 $TC_DP1[1,1]=500 | |
| N20 $TC_DP2[1,1]=3 | ; Schneidenlage |
| N30 $TC_DP3[1,1]=12 | |
| N40 $TC_DP4[1,1]=1 | |
| N50 $TC_DP6[1,1]=6 | |
| N60 $TC_DP10[1,1]=110 | ; Halterwinkel |
| N70 $TC_DP11[1,1]=3 | ; Schnittrichtung |
| N80 $TC_DP24[1,1]=25 | ; Freiwinkel |
| | |
| N90 $TC_CARR7[2]=0 $TC_CARR8[2]=1 $TC_CARR9[2]=0 | ; B-Achse |

```
Programmcode                                          Kommentar
N100 $TC_CARR10[2]=0 $TC_CARR11[2]=0                  ; C-Achse
$TC_CARR12[2]=1
N110 $TC_CARR13[2]=0
N120 $TC_CARR14[2]=0
N130 $TC_CARR21[2]=X
N140 $TC_CARR22[2]=X
N150 $TC_CARR23[2]="M"

N160 TCOABS CUTMOD=0
N170 G18 T1 D1 TCARR=2                                ; X        Y         Z
N180 X0 Y0 Z0 F10000                                  ; 12.000   0.000     1.000

N190 $TC_CARR13[2]=30
N200 TCARR=2
N210 X0 Y0 Z0                                         ; 10.892   0.000    -5.134
N220 G42 Z-10                                         ;  8.696   0.000   -17.330
N230 Z-20                                             ;  8.696   0.000   -21.330
N240 X10                                              ; 12.696   0.000   -21.330
N250 G40 X20 Z0                                       ; 30.892   0.000    -5.134

N260 CUTMOD=2 X0 Y0 Z0                                ;  8.696   0.000    -7.330
N270 G42 Z-10                                         ;  8.696   0.000   -17.330
N280 Z-20                                             ;  8.696   0.000   -21.330
N290 X10                                              ; 12.696   0.000   -21.330
N300 G40 X20 Z0                                       ; 28.696   0.000    -7.330

N310 M30
```

Die Zahlenwerte in den Kommentaren geben jeweils die Satzendpositionen in Maschinenkoordinaten (MKS) in der Reihenfolge X → Y → Z an.

### Erläuterungen

In Satz `N180` wird zunächst das Werkzeug bei `CUTMOD=0` und nicht gedrehtem orientierbaren Werkzeugträger angewählt. Da alle Offsetvektoren des orientierbaren Werkzeugträgers 0 sind, wird die Position angefahren, die den in `$TC_DP3[1,1]` und `$TC_DP4[1,1]` angegebenen Werkzeuglängen entspricht.

In Satz `N200` wird der orientierbare Werkzeugträger mit einer Drehung von 30° um die B-Achse aktiviert. Da die Schneidenlage wegen `CUTMOD=0` nicht modifiziert wird, ist nach wie vor der alte Schneidenbezugspunkt maßgebend. Deshalb wird in Satz `N210` die Position angefahren, die den alten Schneidenbezugspunkt im Nullpunkt beibehält (d. h. der Vektor (1, 12) wird in der Z/X-Ebene um 30° gedreht).

In Satz `N260` ist im Unterschied zu Satz `N200 CUTMOD=2` wirksam. Aufgrund der Drehung des orientierbaren Werkzeugträgers wird die modifizierte Schneidenlage 8. Daraus folgen auch abweichende Achspositionen.

In den Sätzen `N220` bzw. `N270` wird jeweils die Werkzeugradiuskorrektur (WRK) aktiviert. Die unterschiedliche Schneidenlage in beiden Programmstücken hat auf die Endpositionen der

Sätze, in denen die WRK aktiv ist, keinen Einfluss, die entsprechenden Positionen sind deshalb identisch. Erst in den Abwahlsätzen `N260` bzw. `N300` wirken sich die unterschiedlichen Schneidenlagen wieder aus.

## 3.13.13 Mit Werkzeugumgebungen arbeiten

**Übersicht Funktionen**

- Werkzeugumgebung speichern (TOOLENV) (Seite 821)
- Werkzeugumgebung löschen (DELTOOLENV) (Seite 824)
- T-, D- und DL-Nummer lesen (GETTENV) (Seite 825)
- Werkzeuglängen bzw. -längenkomponenten lesen (GETTCOR) (Seite 826)
- Werkzeugkomponenten ändern (SETTCOR) (Seite 833)

**Übersicht Systemvariablen**

- Informationen zu gespeicherten Werkzeugumgebungen lesen ($P_TOOLENVN, ($P_TOOLENV) (Seite 826)

### 3.13.13.1 Werkzeugumgebung speichern (TOOLENV)

Die Funktion TOOLENV dient dazu, alle aktuellen Zustände zu speichern, die für die Bewertung der im Speicher abgelegten Werkzeugdaten von Bedeutung sind.

Das sind im Einzelnen die folgenden Daten:

- Der aktive G-Befehl der Gruppe:
  - 6 (G17, G18, G19)
  - 56 (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)
- Die aktive Planachse
- Die Maschinendaten:
  - MD18112 $MN_MM_KIND_OF_SUMCORR (Eigenschaften der Summenkorrekturen im TO-Bereich)
  - MD20360 $MC_TOOL_PARAMETER_DEF_MASK (Definition der WZ-Parameter)

- Die Settingdaten:
  - SD42900 $SC_MIRROR_TOOL_LENGTH (Vorzeichenwechsel Werkzeuglänge beim Spiegeln)
  - SD42910 $SC_MIRROR_TOOL_WEAR (Vorzeichenwechsel Werkzeugverschleiß beim Spiegeln)
  - SD42920 $SC_WEAR_SIGN_CUTPOS (Vorzeichen des Verschleißes bei Werkzeugen mit Schneidenanlagen)
  - SD42930 $SC_WEAR_SIGN (Vorzeichen des Verschleißes)
  - SD42935 $SC_WEAR_TRANSFORM (Transformationen für Werkzeugkomponenten)
  - SD42940 $SC_LENGTH_CONST (Wechsel der Werkzeuglängenkomponenten bei Ebenenwechsel)
  - SD42942 $SC_TOOL_LENGTH_CONST_T (Wechsel der Werkzeuglängenkomponenten für Drehwerkzeuge bei Ebenenwechsel)
  - SD42950 $SC_TOOL_LENGTH_TYPE (Zuordnung der Werkzeuglängenkomponenten unabhängig vom Werkzeugtyp)
  - SD42954 $SC_TOOL_ORI_CONST_M (Wechsel der Werkzeugorientierungskomponenten für Fräswerkzeuge bei Ebenenwechsel)
  - SD42956 $SC_TOOL_ORI_CONST_T (Wechsel der Werkzeugorientierungskomponenten für Drehwerkzeuge bei Ebenenwechsel)
- Der Orientierungsanteil des aktuellen Gesamtframes (Drehungen und Spiegelungen, keine Nullpunktverschiebungen oder Skalierungen)
- Der Orientierungsanteil und die resultierende Länge des aktiven orientierbaren Werkzeugträgers
- Der Orientierungsanteil und die resultierende Länge einer aktiven Transformation

Außer den genannten Daten, die die Umgebung des Werkzeugs beschreiben, werden auch T-Nummer, D-Nummer und DL-Nummer des aktiven Werkzeugs mit abgelegt, damit später auf dieses Werkzeug in der gleichen Umgebung wie beim Aufruf von TOOLENV zugegriffen werden kann, ohne das Werkzeug erneut bezeichnen zu müssen.

## Syntax

```
<Status> = TOOLENV(<Name>)
```

## Bedeutung

| | | |
|---|---|---|
| `TOOLENV(...)`: | Vordefinierte Funktion zum Speichern einer Werkzeugumgebung | |
| | Alleine im Satz: | ja |

| | | | |
|---|---|---|---|
| <Status>: | Rückgabewert der Funktion. Negative Werte zeigen Fehlerzustände an. | | |
| | Datentyp: | | INT |
| | Wert: | 0 | Funktion OK |
| | | -1 | Kein Speicherplatz für Werkzeugumgebungen reserviert:<br>MD18116 $MN_MM_NUM_TOOL_ENV = 0<br>D. h., die Funktionalität "Werkzeugumgebungen" ist nicht vorhanden. |
| | | -2 | Keine freien Speicherplätze für Werkzeugumgebungen mehr vorhanden. |
| | | -3 | Nullstring als Name einer Werkzeugumgebung ist nicht zulässig. |
| | | -4 | Kein Parameter (<Name>) angegeben. |
| **Parameter** | | | |
| 1 <Name>: | Name, unter dem der aktuelle Datensatz abgelegt werden soll<br>Ist ein Datensatz mit dem gleichen Namen bereits vorhanden, wird dieser überschrieben. Der Status ist in diesem Fall "0". | | |
| | Datentyp: | | STRING |

## Weitere Informationen

### Basismaß/Adaptermaß - Werkzeuglängenkorrektur

Ob die Adapterlänge oder das Basismaß (schneidenspezifische Parameter $TC_DP21, $TC_DP22 und $TC_DP23) in die Berechnung der Werkzeuglänge eingeht, entscheidet bei aktiver Werkzeug-Magazinverwaltung (nur verfügbar mit der Option "Werkzeugverwaltung"!) der Wert des Maschinendatums:

MD18104 $MN_MM_NUM_TOOL_ADAPTER (WZ-Adapter im TO-Bereich)

Da eine Veränderung dieses Maschinendatums nur beim Hochlauf der Steuerung wirksam werden kann, wird es nicht in der Werkzeugumgebung gespeichert.

### Resultierende Länge von orientierbaren Werkzeugträgern und Transformationen

**Hinweis**

Sowohl bei orientierbaren Werkzeugträgern als auch bei Transformationen gibt es Systemvariablen bzw. Maschinendaten, die wie zusätzliche Werkzeuglängenkomponenten wirken, und die ganz oder teilweise den von diesen ausführbaren Drehungen unterworfen werden. Die zusätzlichen Werkzeuglängenanteile, die sich dadurch ergeben, müssen beim Aufruf von TOOLENV mit abgelegt werden, da sie einen Teil der Umgebung bilden, in der das Werkzeug eingesetzt wird.

### Adaptertransformation

Die Adaptertransformation ist eine Eigenschaft des Werkzeugadapters und damit des Gesamtwerkzeugs. Sie ist deshalb nicht Bestandteil einer Werkzeugumgebung, die auf ein anderes Werkzeug angewendet werden kann.

Durch das Abspeichern der gesamten für die Bestimmung der Werkzeuggesamtlänge erforderlichen Daten ist es möglich, zu einem späteren Zeitpunkt die effektive Länge eines

Werkzeugs zu berechnen, auch wenn es zu diesem Zeitpunkt nicht mehr aktiv ist bzw. wenn sich die Umgebungsbedingungen (z. B. G-Befehle oder Settingdaten) geändert haben. Ebenso kann die effektive Länge eines anderen Werkzeugs berechnet werden mit der Annahme, es würde unter den gleichen Bedingungen eingesetzt wie das Werkzeug, für das der Status abgespeichert wurde.

**Maximale Anzahl Datensätze für Werkzeugumgebungen**

Mit dem Maschinendatum MD18116 $MN_MM_NUM_TOOL_ENV wird festgelegt, wie viele Datensätze zur Beschreibung von Werkzeugumgebungen maximal abgelegt werden können. Die Daten liegen im TOA-Bereich. Sie bleiben auch beim Ausschalten der Steuerung erhalten.

Eine Datensicherung ist nicht möglich. Das bedeutet, dass diese Daten nicht zwischen verschiedenen Steuerungen übertragen werden können.

### 3.13.13.2 Werkzeugumgebung löschen (DELTOOLENV)

Mit der Funktion DELTOOLENV können Datensätze zur Beschreibung von Werkzeugumgebungen gelöscht werden. Löschen bedeutet, dass auf den unter einem bestimmten Namen abgelegten Datensatz nicht mehr zugegriffen werden kann (ein Zugriffsversuch führt zu einem Alarm).

**Hinweis**

Datensätze können nur mit der Funktion DELTOOLENV, durch einen INITIAL.INI-Download oder durch einen Kaltstart (NC-Hochlauf mit Standard-Maschinendaten) gelöscht werden. Es gibt keine weiteren automatischen Löschvorgänge.

## Syntax

```
<Status> = DELTOOLENV(<Name>)
<Status> = DELTOOLENV()
```

## Bedeutung

| | | | | |
|---|---|---|---|---|
| `DELTOOLENV(...)`: | | Vordefinierte Funktion zum Löschen einer Werkzeugumgebung | | |
| | | Alleine im Satz: | | ja |
| `<Status>`: | | Rückgabewert der Funktion. Negative Werte zeigen Fehlerzustände an. | | |
| | | Datentyp: | | INT |
| | | Wert: | 0 | Funktion OK |
| | | | -1 | Kein Speicherplatz für Werkzeugumgebungen reserviert:<br>MD18116 $MN_MM_NUM_TOOL_ENV = 0<br>D. h., die Funktionalität "Werkzeugumgebungen" ist nicht vorhanden. |
| | | | -2 | Ein Werkzeugumgebung mit dem angegebenen Namen existiert nicht. |
| **Parameter** | | | | |
| 1 | `<Name>`: | Name des zu löschenden Datensatzes | | |
| | | Datentyp: | | STRING |

| | |
|---|---|
| | |
| DELTOOLENV(): | DELTOOLENV() ohne Angabe eines Namens löscht alle Datensätze zur Beschreibung von Werkzeugumgebungen |

### 3.13.13.3 T-, D- und DL-Nummer lesen (GETTENV)

Die Funktion GETTENV dient dazu, die in einer Werkzeugumgebung abgelegte T-, D- und DL-Nummer zu lesen.

## Syntax

```
<Status> = GETTENV(<Name>, <TDDL>)
```

## Bedeutung

| | | | |
|---|---|---|---|
| GETTENV(...): | Vordefinierte Funktion zum Lesen von T-, D- und DL-Nummern in einem Datensatz zur Beschreibung einer Werkzeugumgebung | | |
| | Alleine im Satz: | | ja |
| <Status>: | Rückgabewert der Funktion. Negative Werte zeigen Fehlerzustände an. | | |
| | Datentyp: | | INT |
| | Wert: | 0 | Funktion OK |
| | | -1 | Kein Speicherplatz für Werkzeugumgebungen reserviert:<br>MD18116 $MN_MM_NUM_TOOL_ENV = 0<br>D. h., die Funktionalität "Werkzeugumgebungen" ist nicht vorhanden. |
| | | -2 | Ein Werkzeugumgebung mit dem angegebenen Namen existiert nicht. |
| **Parameter** | | | |
| 1 <Name>: | Name der Datensatzes, aus dem die T-, D- und DL-Nummer gelesen werden sollen | | |
| | Datentyp: | | STRING |
| 2 <TDDL>: | Das Feld dieses Ergebnisparameters enthält die T-, D- und DL-Nummer des Werkzeugs, dessen Werkzeugumgebung im angegebenen Datensatz abgelegt ist:<br>• <TDDL> [0]: T-Nummer<br>• <TDDL> [1]: D-Nummer<br>• <TDDL> [2]: DL-Nummer | | |
| | Datentyp: | | INT[3] |
| | | | |
| GETTENV(,<TDDL>),<br>GETTENV("",<TDDL>): | Es ist zulässig, beim Aufruf der Funktion GETTENV den ersten Parameter wegzulassen oder als ersten Parameter den Nullstring zu übergeben. In diesen beiden Sonderfällen werden in <TDDL> die T-, D- und DL-Nummern des **aktiven** Werkzeugs zurückgeliefert. | | |

### 3.13.13.4 Informationen zu gespeicherten Werkzeugumgebungen lesen ($P_TOOLENVN, ($P_TOOLENV)

Informationen zu gespeicherten Werkzeugumgebungen sind über folgende Systemvariablen lesbar:

| | | | | |
|---|---|---|---|---|
| `$P_TOOLENVN`: | Liefert die Anzahl der mittels TOOLENV definierten (und noch nicht gelöschten) Datensätze zur Beschreibung von Werkzeugumgebungen | | | |
| | Syntax: | `<n> = $P_TOOLENVN` | | |
| | Bedeutung: | `<n>`: | Anzahl der definierten Datensätze | |
| | | | Datentyp: | INT |
| | | | Wertebereich: | 0 ... MD18116 $MN_MM_NUM_TOOL_ENV |
| | Der Zugriff auf diese Systemvariable ist auch dann zulässig, wenn keine Werkzeugumgebungen möglich sind (MD18116 = 0). In diesem Fall ist der Rückgabewert "0". | | | |
| `$P_TOOLENV`: | Liefert den Namen des <i>-ten Datensatzes zur Beschreibung einer Werkzeugumgebung | | | |
| | Syntax: | `<Name> = $P_TOOLENV[<i>]` | | |
| | Bedeutung: | `<Name>`: | Name des Datensatzes mit der Nummer <i> | |
| | | | Datentyp: | STRING |
| | | `<i>`: | Nummer des Datensatzes | |
| | | | Datentyp: | INT |
| | | | Wertebereich: | 1 ... $P_TOOLENVN |
| | Die Zuordnung der Nummern zu den Datensätzen ist nicht fix, sondern kann sich durch das Löschen oder Neuanlegen von Datensätzen verändern. Die Nummerierung der Datensätze wird intern vorgenommen.<br>Verweist <i> auf einen nicht definierten Datensatz, wird der Nullstring zurückgeliefert.<br>Ist der Index <i> ungültig, d. h. <i> ist kleiner 1 oder größer als die maximale Anzahl Datensätze für Werkzeugumgebungen (MD18116 $MN_MM_NUM_TOOLENV), wird folgender Alarm ausgegeben:<br>Alarm 17020 "unerlaubter Array-Index1" | | | |

### 3.13.13.5 Werkzeuglängen bzw. -längenkomponenten lesen (GETTCOR)

Die Funktion GETTCOR dient dazu, Werkzeuglängen bzw. Werkzeuglängenkomponenten auszulesen.

Dabei kann durch Parametrierung angegeben werden, welche Komponenten berücksichtigt werden sollen, und unter welchen Einsatzbedingungen das Werkzeug betrachtet werden soll.

#### Syntax

```
<Status> = GETTCOR(<Len>[, <Comp>, <Stat>, <T>, <D>, <DL>])
```

## Bedeutung

| | | | |
|---|---|---|---|
| GETTCOR(...): | Vordefinierte Funktion zum Lesen von Werkzeuglängen bzw. Werkzeuglängenkomponenten | | |
| | Alleine im Satz: | | ja |
| <Status>: | Rückgabewert der Funktion. Negative Werte zeigen Fehlerzustände an. | | |
| | Datentyp: | | INT |
| | Wert: | 0 | Funktion OK |
| | | -1 | Kein Speicherplatz für Werkzeugumgebungen reserviert:<br>MD18116 $MN_MM_NUM_TOOL_ENV = 0<br>D. h., die Funktionalität "Werkzeugumgebungen" ist nicht vorhanden. |
| | | -2 | Eine Werkzeugumgebung mit dem unter <Stat> angegebenen Namen existiert nicht. |
| | | -3 | Ungültiger String im Parameter <Comp>.<br>Ursache dieses Fehlers können ungültige oder doppelt programmierte Zeichen sein. |
| | | -4 | Ungültige T-Nummer |
| | | -5 | Ungültige D-Nummer |
| | | -6 | Ungültige DL-Nummer |
| | | -7 | Zugriffsversuch auf nicht vorhandenes Speichermodul. |
| | | -8 | Zugriffsversuch auf eine nicht vorhandene Option (z. B. programmierbare Werkzeugorientierung, Werkzeugverwaltung). |
| | | -9 | Der String <Comp> enthält einen Doppelpunkt (Kennung für die Spezifikationen eines Koordinatensystems), es folgt jedoch kein gültiges Zeichen zur Bezeichnung des verlangten Koordinatensystems. |
| **Parameter** | | | |

| | | | |
|---|---|---|---|
| 1 | <Len>: | Ergebnisvektor | |
| | | Datentyp: | REAL[11] |
| | | Die Vektorkomponenten sind in folgender Reihenfolge angeordnet:<br>• <Len> [0]: Werkzeugtyp<br>• <Len> [1]: Schneidenlage<br>• <Len> [2]: Abszisse<br>• <Len> [3]: Ordinate<br>• <Len> [4]: Applikate<br>• <Len> [5]: Werkzeugradius<br>Als Bezugskoordinatensystem für die Längenkomponenten gilt das in <Comp> und <Stat> definierte Koordinatensystem. Ist in <Comp> kein Koordinatensystem definiert, werden die Werkzeuglängen im Maschinenkoordinatensystem dargestellt.<br>Die Zuordnung von Abszisse, Ordinate und Applikate zu den Geometrieachsen ist abhängig von der in der verwendeten Werkzeugumgebung aktiven Ebene. D. h., bei G17 ist die Abszisse parallel zu X, bei G18 zu Z usw.<br>Die Komponenten <Len>[6] bis <Len>[10] enthalten die zusätzlichen Parameter, die für die Geometriebeschreibung eines Werkzeugs angegeben werden können (z. B. $TC_DP7 bis $TC_DP11 für die Geometrie bzw. die entsprechenden Komponenten für Verschleiß bzw. Summen- und Einrichtekorrekturen).<br>Diese 5 zusätzlichen Elemente und der Werkzeugradius sind nur für die Komponenten E, G, S und W definiert. Deren Bewertung hängt nicht von <Stat> ab. Die entsprechenden Werte in <Len>[6] bis <Len>[10] können deshalb nur dann ungleich null sein, wenn an der Werkzeuglängenberechnung mindestens eine der vier genannten Komponenten beteiligt ist. Die übrigen Komponenten haben keinen Einfluss auf das Ergebnis. Die Maßangaben beziehen sich auf das Grundsystem der Steuerung (inch bzw. metrisch). | |

| | | | | | |
|---|---|---|---|---|---|
| 2 | <Comp>: | Werkzeuglängenkomponenten (optional) | | | |
| | | Datentyp: | STRING | | |
| | | Die Zeichenkette besteht aus zwei Teilstrings, die durch einen Doppelpunkt voneinander getrennt sind.<br>Allg. Form: "<SubStr_1> [: <SubStr_2]" | | | |
| | | <SubStr_1>: | Der **erste Teilstring** bezeichnet die Werkzeuglängenkomponenten, die bei der Werkzeuglängenberechnung berücksichtigt werden sollen.<br>Die Reihenfolge der Zeichen im Teilstring sowie deren Schreibweise (groß oder klein) ist beliebig. Zwischen den Zeichen können beliebig viele Leerzeichen oder Tabulatorzeichen (white spaces) eingefügt werden.<br>**Hinweis:**<br>Die Zeichen im Teilstring dürfen **nicht doppelt** programmiert werden! | | |
| | | | Zeichen: | – | Minuszeichen (nur als erstes Zeichen zulässig!)<br>Es wird die gesamte Werkzeuglänge abzüglich der Komponenten berechnet, die im nachfolgenden String spezifiziert sind. |
| | | | | C | Adaptermaß oder Basismaß (diejenige der beiden alternativ existierenden Komponenten, die für das verwendete Werkzeug aktiv ist) |
| | | | | E | Einrichtkorrekturen |
| | | | | G | Geometrie |
| | | | | K | Kinematische Transformation (wird nur bei generischer 3-, 4- und 5-Achs-Transformation ausgewertet!) |
| | | | | S | Summenkorrekturen |
| | | | | T | Orientierbarer Werkzeugträger |
| | | | | W | Verschleiß |
| | | | Ist der erste Teilstring (mit Ausnahme von white spaces) leer, so bedeutet das, dass die gesamte Werkzeuglänge unter Berücksichtigung aller Komponenten berechnet werden soll. Gleiches gilt auch dann, wenn der Parameter <Comp> nicht angegeben wird. | | |
| | | <Substr_2>: | Der optionale **zweite Teilstring** bezeichnet das Koordinatensystem, in dem die Werkzeuglänge ausgegeben werden soll.<br>Der zweite Teilstring besteht nur aus einem einzigen relevanten Zeichen. | | |
| | | | Zeichen: | A | Einstellbares Koordinatensystem (ACS) |
| | | | | B | Basiskoordinatensystem (BKS) |
| | | | | K | Werkzeugkoordinatensystem der kinematischen Transformation (KCS) |
| | | | | M | Maschinenkoordinatensystem (MKS) |
| | | | | T | Werkzeugkoordinatensystem (TCS) |
| | | | | W | Werkstückkoordinatensystem (WKS) |
| | | | Wird kein Koordinatensystem angegeben, erfolgt die Bewertung im Maschinenkoordinatensystem (MKS). Die gegebenen- | | |

| | | | |
|---|---|---|---|
| | | | falls zu berücksichtigenden Drehungen werden durch die in <Stat> definierte Werkzeugumgebung festgelegt. |
| 3 | <_Stat>: | Name des Datensatzes zur Beschreibung einer Werkzeugumgebung (optional) | |
| | | Datentyp: | STRING |
| | | Ist der Wert dieses Parameters der Nullstring ("") oder wird er nicht angeben, wird der aktuelle Zustand verwendet. Ist kein Werkzeug spezifiziert, wird das aktuelle Werkzeug verwendet. | |
| 4 | <T>: | Interne T-Nummer des Werkzeugs (optional) | |
| | | Datentyp: | INT |
| | | Ist dieser Parameter nicht angegeben oder ist sein Wert "0", wird das in <Stat> abgelegte Werkzeug verwendet.<br>Ist der Wert dieses Parameters "-1", wird die T-Nummer des aktiven Werkzeugs verwendet. Es ist auch zulässig, die Nummer des aktiven Werkzeugs explizit anzugeben.<br>**Hinweis:**<br>Ist <Stat> nicht angegeben, wird als Werkzeugumgebung der aktuelle Zustand verwendet. Da mit <T> = 0 auf die in der Werkzeugumgebung gespeicherte T-Nummer verwiesen wird, wird in dieser das aktive Werkzeug verwendet, d. h. die Angaben <T> = 0 und <T> = -1 sind in diesem Sonderfall gleichbedeutend. | |
| 5 | <D>: | Schneidennummer des Werkzeugs (optional) | |
| | | Datentyp: | INT |
| | | Ist dieser Parameter nicht angegeben oder ist sein Wert "0", richtet sich die verwendetet D-Nummer nach der Quelle der T-Nummer. Wird die T-Nummer aus der Werkzeugumgebung verwendet, wird auch die D-Nummer der Werkzeugumgebung gelesen, andernfalls die des aktuell aktiven Werkzeugs. | |
| 6 | <DL>: | Nummer der ortsabhängigen Korrektur (optional) | |
| | | Datentyp: | INT |
| | | Ist dieser Parameter nicht angegeben, richtet sich die verwendete DL-Nummer nach der Quelle der T-Nummer. Wird die T-Nummer aus der Werkzeugumgebung verwendet, wird auch die D-Nummer der Werkzeugumgebung gelesen, andernfalls die des aktuell aktiven Werkzeugs. | |

## Beispiele

| | |
|---|---|
| `GETTCOR(_LEN)` | Berechnet die Werkzeuglänge des aktuellen aktiven Werkzeugs im Maschinenkoordinatensystem unter Berücksichtigung aller Komponenten. |
| `GETTCOR(_LEN,"CGW:W")` | Es wird die Werkzeuglänge bestehend aus Adapter- bzw. Basismaß, Geometrie und Verschleiß für das aktive Werkzeug berechnet. Weitere Anteile wie z. B. orientierbarer Werkzeugträger oder kinematische Transformationen werden nicht berücksichtigt. Die Ausgabe erfolgt im Werkstückkoordinatensystem. |

| | |
|---|---|
| `GETTCOR (_LEN,"-K:B")` | Es wird die gesamte Werkzeuglänge des aktiven Werkzeugs ohne Berücksichtigung der Längenkomponenten einer eventuell aktiven kinematischen Transformation berechnet. Ausgabe im Basiskoordinatensystem. |
| `GETTCOR (_LEN,":M","Testenv1",,3)` | Es wird die gesamte Werkzeuglänge für das in der Werkzeugumgebung mit dem Namen "Testenv1" abgelegte Werkzeug im Maschinenkoordinatensystem berechnet. Die Berechnung erfolgt jedoch unabhängig von der abgelegten Schneidennummer für die Schneidennummer D3. |

## Weitere Informationen

### Adaptertransformation / Orientierbare Werkzeugträger / Kinematische Transformation

Die Drehungen und Komponentenvertauschungen, die gegebenenfalls von der Adaptertransformation, dem orientierbaren Werkzeugträger und einer kinematischen Transformation ausgeführt werden, sind Bestandteil der Werkzeugumgebung. Sie werden deshalb immer ausgeführt, auch dann, wenn die entsprechende Längenkomponente nicht berücksichtigt werden soll. Ist das nicht gewünscht, müssen Werkzeugumgebungen definiert werden, bei denen die entsprechenden Transformationen nicht aktiv sind. In vielen Fällen (nämlich immer dann, wenn an einer Maschine keine Transformationen bzw. orientierbare Werkzeugträger eingesetzt werden), erfüllen die abgespeicherten Datensätze für Werkzeugumgebungen diese Bedingungen automatisch, sodass sie vom Anwender nicht speziell beachtet werden müssen.

### Dreh- und Schleifwerkzeuge: Berechnung der Werkzeuglänge in Abhängigkeit von MD20360 $MC_TOOL_PARAMETER_DEF_MASK

Wie bei Dreh- und Schleifwerkzeugen der Verschleiß bzw. die Werkzeuglänge in einer eventuell vorhandenen Durchmesserachse zu bewerten ist, wird festgelegt im Maschinendatum:

MD20360 $MC_TOOL_PARAMETER_DEF_MASK (Definition der WZ-Parameter)

| Bit | Wert | |
|---|---|---|
| 0 | Bei Dreh- und Schleifwerkzeugen wird der **Verschleißparameter** der Planachse als Durchmesserwert eingerechnet: | |
| | = 0 (Default) | nein |
| | = 1 | ja |
| 1 | Bei Dreh- und Schleifwerkzeugen wird die **Werkzeuglängenkomponente** der Planachse als Durchmesserwert eingerechnet: | |
| | = 0 (Default) | nein |
| | = 1 | ja |

Sind die betreffenden Bits gesetzt, wird der zugehörige Eintrag mit dem Faktor 0,5 bewertet. Diese Bewertung schlägt sich auch in der von GETTCOR gelieferten Werkzeuglänge nieder.

**Beispiel:**

MD20360 $MC_TOOL_PARAMETER_DEF_MASK = 3

MD20100 $MC_DIAMETER_AX_DEF (Geometrieachse mit Planachsfunktion) = "X"

X ist Durchmesserachse (Standard-Drehmaschinenkonfiguration)

| **Programmcode** | **Kommentar** |
|---|---|
| N30 $TC_DP1[1,1]=500 | |
| N40 $TC_DP2[1,1]=2 | |
| N50 $TC_DP3[1,1]=3.0 | ; Geometrie L1 |
| N60 $TC_DP4[1,1]=4.0 | |
| N70 $TC_DP5[1,1]=5.0 | |
| N80 $TC_DP12[1,1]=12.0 | ; Verschleiß L1 |
| N90 $TC_DP13[1,1]=13.0 | |
| N100 $TC_DP14[1,1]=14.0 | |
| N110 T1 D1 G18 | |
| N120 R1=GETTCOR(_LEN,"GW") | |
| N130 R3=_LEN[2] | ; 17.0 (= 4.0 + 13.0) |
| N140 R4=_LEN[3] | ; 7.5 (= 0.5 * 3.0 + 0.5 * 12.0) |
| N150 R5=_LEN[4] | ; 19.0 (= 5.0 + 14.0) |
| N160 M30 | |

### Längenkomponenten von kinematischer Transformation und orientierbarem Werkzeugträger

Wird bei der Werkzeuglängenberechnung ein **orientierbarer Werkzeugträger** berücksichtigt, gehen folgende Vektoren in die Werkzeuglängenberechnung ein:

| Typ | Vektoren |
|---|---|
| M | l1 und l2 |
| T | l1, l2 und l3 |
| P | Werkzeuglänge wird durch den orientierbaren Werkzeugträger nicht beeinflusst. |

Bei der generischen **5-Achstransformation** gehen bei den Trafotypen 24 und 56 folgende Maschinendaten in die Werkzeuglängenberechnung ein:

| Trafotyp | Maschinendaten |
|---|---|
| 24 | MD24550/24650 $MC_TRAFO5_BASE_TOOL_1/2<br>MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2<br>MD24558/24658 $MC_TRAFO5_PART_OFFSET_1/2 |
| 56 | MD24550/24650 $MC_TRAFO5_BASE_TOOL_1/2<br>MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2 |

Der Trafotyp 56 (bewegliches Werkzeug und bewegliches Werkstück) entspricht dem Typ M beim orientierbaren Werkzeugträger.

Bei dieser 5-Achs-Transformation entspricht in den bisherigen Softwareständen der Vektor MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2 (Vektor des kinematischen Versatzes der 1./2. 5-Achstransformation im Kanal) der Summe der beiden Vektoren $l_1$ und $l_3$ beim orientierbaren Werkzeugträger vom Typ M.

Für die Transformation ist in beiden Fällen nur die Summe relevant. Die Art der Zusammensetzung aus den beiden Einzelkomponenten ist bedeutungslos. Bei der Berechnung der Werkzeuglänge ist es jedoch von Bedeutung, welcher Anteil dem Werkzeug und welcher dem Werkzeugtisch zuzuordnen ist. Deshalb wurde das Maschinendatum MD24558/24658 $MC_TRAFO5_JOINT_OFFSET_PART_1/2 (Vektor kinematischer Versatz

im Tisch) eingeführt. Es entspricht dem Vektor l3. Das Maschinendatum MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2 entspricht jetzt nicht mehr der Summe aus l1 und l3, sondern nur noch dem Vektor l1. Ist das Maschinendatum MD24558/24658 $MC_TRAFO5_JOINT_OFFSET_PART_1/2 gleich null, so ist das Verhalten identisch zum bisherigen Verhalten.

**Kompatibilität**

Die Funktion GETTCOR dient zusammen mit den Funktionen TOOLENV und SETTCOR unter anderem dazu, Teile der Funktionalität zu ersetzen, die bisher extern in den Messzyklen realisiert war.

In den Messzyklen wurde nur ein Teil der Parameter ausgewertet, die letztlich die effektive Werkzeuglänge bestimmen. Die genannten Funktionen können so parametriert werden, dass das Verhalten der Messzyklen bezüglich der Werkzeuglängenberechnung reproduziert werden kann.

### 3.13.13.6 Werkzeugkomponenten ändern (SETTCOR)

Die Funktion SETTCOR dient dazu, Werkzeugkomponenten unter Berücksichtigung aller Randbedingungen, die in die Bewertung der einzelnen Komponenten eingehen können, zu verändern.

**Hinweis**

Zur Terminologie: Wenn im Folgenden im Zusammenhang mit der Werkzeuglänge von Werkzeugkomponenten die Rede ist, sind die vektoriell betrachteten Komponenten gemeint, aus denen sich die gesamte Werkzeuglänge zusammensetzt (z. B. Geometrie oder Verschleiß). Eine solche Komponente setzt sich deshalb aus drei Einzelwerten (L1, L2, L3) zusammen, die im Folgenden als Koordinatenwerte bezeichnet werden.

Die Werkzeugkomponente "Geometrie" besteht somit beispielsweise aus den drei Koordinatenwerten $TC_DP3 bis $TC_DP5.

## Syntax

```
<Status> = SETTCOR(<CorVal>, <Comp>, [<CorComp>, <CorMode>, <GeoAx>,
<Stat>, <T>, <D>, <DL>])
```

## Bedeutung

| | | |
|---|---|---|
| `SETTCOR(...)`: | Vordefinierte Funktion zum Ändern von Werkzeugkomponenten | |
| | Alleine im Satz: | ja |

| | | | |
|---|---|---|---|
| <Status>: | Rückgabewert der Funktion. Negative Werte zeigen Fehlerzustände an. | | |
| | Datentyp: | | INT |
| | Wert: | 0 | Funktion OK |
| | | -1 | Kein Speicherplatz für Werkzeugumgebungen reserviert:<br>MD18116 $MN_MM_NUM_TOOL_ENV = 0<br>D. h., die Funktionalität "Werkzeugumgebungen" ist nicht vorhanden. |
| | | -2 | Eine Werkzeugumgebung mit dem unter <Stat> angegebenen Namen existiert nicht. |
| | | -3 | Ungültiger String im Parameter <Comp>.<br>Ursache dieses Fehlers können ungültige oder doppelt programmierte Zeichen sein. |
| | | -4 | Ungültige T-Nummer. |
| | | -5 | Ungültige D-Nummer. |
| | | -6 | Ungültige DL-Nummer. |
| | | -7 | Zugriffsversuch auf nicht vorhandenes Speichermodul. |
| | | -8 | Zugriffsversuch auf eine nicht vorhandene Option (z. B. programmierbare Werkzeugorientierung, Werkzeugverwaltung). |
| | | -9 | Unzulässiger Zahlenwert für den Parameter <CorComp>. |
| | | -10 | Unzulässiger Zahlenwert für den Parameter <CorMode>. |
| | | -11 | Die Inhalte der Parameter <Comp> und <CorComp> sind widersprüchlich. |
| | | -12 | Die Inhalte der Parameter <Comp> und <CorMode> sind widersprüchlich. |
| | | -13 | Der Inhalt des Parameters <GeoAx> bezeichnet keine Geometrieachse. |
| | | -14 | Schreibversuch auf nicht vorhandene Einrichtekorrektur. |
| **Parameter** | | | |
| 1 | <CorVal>: | Korrekturvektor<br>In dem durch <Stat> definierten Werkstückkoordinatensystem (WKS) gilt folgende Zuordnung:<br>• <CorVal> [0]: Abszisse<br>• <CorVal> [1]: Ordinate<br>• <CorVal> [2]: Applikate<br>Soll nur eine Werkzeugkomponente korrigiert werden (d. h. keine vektorielle Korrektur, siehe Parameter <CorMode>), steht der Korrekturwert immer in <CorVal>[0], gleichgültig auf welche Achse er wirkt. Der Inhalt der beiden übrigen Komponenten wird dann nicht ausgewertet.<br>Bezieht sich <CorVal> bzw. eine Komponente von <CorVal> auf die Planachse, so wird die Angabe als **Radiusmaß** bewertet. Das bedeutet, dass ein Werkzeug z. B. um das angegebene Maß "länger" wird, was entsprechend zu einer doppelt so großen Änderung des Werkstückdurchmessers führt.<br>Die Maßangaben beziehen sich auf das **Grundsystem** (inch bzw. metrisch) der Steuerung. | |
| | | Datentyp: | REAL[3] |

| | | | | | |
|---|---|---|---|---|---|
| 2 | <Comp>: | Werkzeugkomponente(n) | | | |
| | | Datentyp: | STRING | | |
| | | Die Zeichenkette besteht aus zwei Teilstrings, die durch einen Doppelpunkt voneinander getrennt sind.<br>Allg. Form: "<SubStr_1> [: <SubStr_2]" | | | |
| | | <SubStr_1>: | Der **erste Teilstring** muss immer vorhanden sein und kann entweder aus ein oder zwei Zeichen bestehen. Dabei steht das erste oder einzige Zeichen für die 1. Komponente ($Val_1$) und das zweite Zeichen für die 2. Komponente ($Val_2$), die entsprechend den nachfolgenden Parametern <CorComp> und <CorMode> bearbeitet werden. | | |
| | | | Zeichen: | C | Adaptermaß oder Basismaß (diejenige der beiden alternativ existierenden Komponenten, die für das verwendete Werkzeug aktiv ist) |
| | | | | E | Einrichtkorrekturen |
| | | | | G | Geometrie |
| | | | | S | Summenkorrekturen |
| | | | | W | Verschleiß |
| | | <Substr_2>: | Der **zweite Teilstring** ist optional. Er kann alternativ aus dem (einzigen) Buchstaben "W" oder "T" bestehen. | | |
| | | | Zeichen: | W | Ist der zweite Teilstring leer oder enthält er den Buchstaben "W", werden die Korrekturwerte so verrechnet, als seien sie im Werk**stück**koordinatensystem (WKS) vermessen worden. |
| | | | | T | Enthält der zweite Teilstring den Buchstaben "T", werden Korrekturwerte werden so verrechnet, als seien sie im Werk**zeug**koordinatensystem (**T**ool Coordinate System, TCS) vermessen worden. |
| | | Die Schreibweise der Zeichen im String (groß oder klein) ist beliebig. Es können beliebig viele Leerzeichen oder Tabulatorzeichen (white spaces) eingefügt werden. | | | |

| | | | | |
|---|---|---|---|---|
| 3 | <CorComp>: | Spezifiziert die Komponente(n) des Werkzeugdatensatzes, die beschrieben werden sollen (optional) | | |
| | | Datentyp: | | INT |
| | | Wert: | 0 | Der Korrekturwert <CorVal>[0] bezieht sich auf die im Parameter <GeoAx> übergebene Geometrieachse im Werkstückkoordinatensystem oder im Werkzeugkoordinatensystem (siehe dazu die Beschreibung des Parameters <Comp>). D. h., der Korrekturwert muss in die bezeichneten Werkzeugkomponenten so eingerechnet werden, dass sich unter Berücksichtigung aller Parameter, die die Werkzeuglängenberechnung beeinflussen können, als Ergebnis eine Änderung der gesamten Werkzeuglänge in der gegebenen Achsrichtung um den vorgegebenen Wert ergibt.<br>Diese Änderung soll erreicht werden durch die Korrektur der unter <Comp> angegebenen Komponente und der unter <CorMode> (siehe nachfolgenden Parameter) angegebenen symbolischen Rechenvorschrift. Die resultierende Korrektur kann sich deshalb in allen drei Achskomponenten auswirken. |
| | | | 1 | Wie "0", jedoch vektoriell. Der Inhalt des Vektors <CorVal> bezieht sich auf Abszisse, Ordinate und Applikate im Werkstückkoordinatensystem oder Werkzeugkoordinatensystem (siehe dazu die Beschreibung des Parameters <Comp>).<br>Der nachfolgende Parameter <GeoAx> wird nicht ausgewertet. |
| | | | 2 | Vektorielle Korrektur, d. h. L1, L2 und L3 können simultan verändert werden.<br>Im Gegensatz zu den Varianten "0" und "1" beziehen sich die in <CorVal> enthaltenen Korrekturwerte jedoch auf die Koordinaten der $Val_1$-Komponente (siehe nachfolgenden Parameter <CorMode>) des Werkzeugs.<br>Eine eventuell vorhandene Schrägstellung des Werkzeugs gegenüber dem Werkstückkoordinatensystem hat keinen Einfluss auf die Korrektur. |
| | | | 3 - 5 | Korrektur der Werkzeuglänge L1 bis L3 ($TC_DP3 bis $TC_DP5) bzw. der entsprechenden Werte bei Verschleiß, Einrichte- oder Summenkorrekturen.<br>Der Korrekturwert ist in <CorVal>[0] enthalten. Er ist in den Koordinaten der $Val_1$-Komponente (siehe nachfolgenden Parameter <CorMode>) des Werkzeugs gemessen. Eine eventuell vorhandene Schrägstellung des Werkzeugs gegenüber dem Werkstückkoordinatensystem hat keinen Einfluss auf die Korrektur. |
| | | | 6 | Korrektur des Werkzeugradius ($TC_DP6) bzw. der entsprechenden Werte bei Verschleiß, Einrichte- oder Summenkorrekturen. Die Bits 10 und 11 (Bewertung der Durchmesser- bzw. Durchmesserverschleißangabe wahlweise als Radius- oder Durchmesserangabe) im Maschinendatum MD20360 $MC_TOOL_PARAMETER_DEF_MASK werden berücksichtigt. |
| | | | 7 – 11 | Korrektur von $TC_DP7 bis $TC_DP11 bzw. der entsprechenden Werte bei Verschleiß, Einrichte- oder Summenkorrekturen. Diese Parameter werden wie der Werkzeugradius behandelt. |
| | | Ist dieser Parameter nicht angeben, ist sein Wert "0". | | |

| | | | | |
|---|---|---|---|---|
| 4 | <CorMode>: | Spezifiziert die Art der auszuführenden Schreiboperation (optional) | | |
| | | Datentyp: | | INT |
| | | Wert: | 0 | $Val_{1neu}$ = <CorVal> |
| | | | 1 | $Val_{1neu}$ = $Val_{1alt}$ + <CorVal> |
| | | | 2 | $Val_{1neu}$ = <CorVal><br>$Val_{2neu}$ = 0 |
| | | | 3 | $Val_{1neu}$ = $Val_{1alt}$ + $Val_{2alt}$ + <CorVal><br>$Val_{2neu}$ = 0 |
| | | Die Schreibweise $Val_{1alt}$ + $Val_{2alt}$ ist symbolisch zu verstehen. Werden die beiden Komponenten (aufgrund des Zustandes von <_Stat>) unterschiedlich bewertet, d. h. ist zwischen beiden Komponenten eine Drehung wirksam, so wird $Val_{2alt}$ vor der Addition so transformiert, dass die resultierende Werkzeuglänge nach dem Löschen von $Val_{2neu}$ und vor der Addition von <CorVal> unverändert bleibt.<br><CorVal> bezieht sich immer auf $Val_1$. <CorVal> ist ein Wert, der abhängig vom zweiten Teils des Parameters <Comp> im Werkstückkoordinatensystem (WKS) oder im Werkzeugkoordinatensystem (TCS) gemessen wird. Er ist deshalb gegenüber der Werkzeugkomponente, in die er eingerechnet werden soll, gegebenenfalls bereits transformiert. Er kann somit nicht unmittelbar mit dem abgespeicherten Wert verrechnet werden, sondern muss vor der Addition zu $Val_1$ bzw. $Val_2$ zurücktransformiert werden. Das kann zur Folge haben, dass sich die Korrektur in einer anderen als in der durch <CorComp> definierten Achse auswirkt, bzw. dass diese auf mehrere Achsen wirkt.<br>Für den Fall <CorComp> = 0, d. h., wenn <CorVal> keinen Vektor, sondern nur einen Einzelwert enthält, werden die beschriebene Operationen in den Koordinaten ausgeführt, in denen <CorVal> gemessen wurde (WKS / TCS). Insbesondere gilt das auch für das Nullsetzen von $Val_{2neu}$ in den Varianten 2 und 3. Dieses Ergebnis wird dann in die Koordinaten des Werkzeugs zurücktransformiert. Das kann zur Folge haben, dass keine der nullzusetzenden Koordinatenwerte (L1, L2, L3) Null wird bzw. dass Koordinatenwerte, die vorher Null waren, nun ungleich Null sind. Werden die entsprechenden Operationen jedoch sukzessive für alle drei Geometrieachsen ausgeführt, ist gewährleistet, dass auch alle drei Koordinatenwerte der zu löschenden Komponente Null werden. Ist das Werkzeug gegenüber dem Werkstückkoordinatensystem nicht gedreht, oder aber so gedreht, dass alle Werkzeugkomponenten parallel zu den Koordinatenachsen bleiben (Achsvertauschungen), ist gewährleistet, dass sich jeweils auch nur eine Werkzeugkoordinate ändert.<br>Die sukzessive Ausführung der gleichen Operation (<CorMode>) mit <CorComp> = 0 für alle drei Koordinatenachsen in beliebiger Reihenfolge ist identisch zu der einmaligen Ausführung der gleichen Operation mit <CorComp>=1.<br>Für die Parameterwerte "0" und "1" muss der Parameter <Comp> einen und für die Parameterwerte "2" und "3" zwei Zeichen enthalten.<br>Beispiel:<br><Comp> enthält den String "ES", <CorMode> den Wert "2"<br>⇒ $Einrichtkorrektur_{neu}$ = <CorVal>, $Summenkorrektur_{neu}$ = 0<br>Ist der Parameter <CorMode> nicht angeben, ist sein Wert "0". | | |

| | | | |
|---|---|---|---|
| 5 | <GeoAx>: | Gibt den Index der Geometrieachse an, in der der Korrekturwert <CorVal>[0] gemessen wurde (optional) | |
| | | Datentyp: | INT |
| | | Wertebereich: | 0 ... 2 |
| | | Die Indizes 0 bis 2 beziehen sich auf Abszisse, Ordinate und Applikate in der wirksamen Ebene (G17/G18/G19) der aktuellen Werkzeugumgebung.<br>Der Inhalt dieses Parameters wird nur dann ausgewertet, wenn der Parameter <CorComp> den Wert "0" hat. | |
| 6 | <Stat>: | Name des Datensatzes zur Beschreibung einer Werkzeugumgebung (optional) | |
| | | Datentyp: | STRING |
| | | Ist der Wert dieses Parameters der Nullstring ("") oder wird er nicht angeben, wird der aktuelle Zustand verwendet. Ist kein Werkzeug spezifiziert, wird das aktuelle Werkzeug verwendet. | |
| 7 | <T>: | Interne T-Nummer des Werkzeugs (optional) | |
| | | Datentyp: | INT |
| | | Ist dieser Parameter nicht angegeben oder ist sein Wert "0", wird das in <Stat> abgelegte Werkzeug verwendet.<br>Ist der Wert dieses Parameters "-1", wird die T-Nummer des aktiven Werkzeugs verwendet. Es ist auch zulässig, die Nummer des aktiven Werkzeugs explizit anzugeben.<br>**Hinweis:**<br>Ist <Stat> nicht angegeben, wird als Werkzeugumgebung der aktuelle Zustand verwendet. Da mit <T> = 0 auf die in der Werkzeugumgebung gespeicherte T-Nummer verwiesen wird, wird in dieser das aktive Werkzeug verwendet, d. h. die Angaben <T> = 0 und <T> = -1 sind in diesem Sonderfall gleichbedeutend. | |
| 8 | <D>: | Schneidennummer des Werkzeugs (optional) | |
| | | Datentyp: | INT |
| | | Ist dieser Parameter nicht angegeben oder ist sein Wert "0", richtet sich die verwendetet D-Nummer nach der Quelle der T-Nummer. Wird die T-Nummer aus der Werkzeugumgebung verwendet, wird auch die D-Nummer der Werkzeugumgebung gelesen, andernfalls die des aktuell aktiven Werkzeugs. | |
| 9 | <TL>: | Nummer der ortsabhängigen Korrektur (optional) | |
| | | Datentyp: | INT |
| | | Ist dieser Parameter nicht angegeben, richtet sich die verwendete DL-Nummer nach der Quelle der T-Nummer. Wird die T-Nummer aus der Werkzeugumgebung verwendet, wird auch die D-Nummer der Werkzeugumgebung gelesen, andernfalls die des aktuell aktiven Werkzeugs. Spezifizieren T, D und DL ein Werkzeug ohne ortsabhängige Korrekturen, dürfen im Parameter <Comp> keine Summen- oder Einrichtkorrekturen spezifiziert werden (Fehlercode in <Status>). | |

**Hinweis**

Nicht alle möglichen Kombinationen der drei Parameter <Comp>, <CorComp> und <CorMode> sind sinnvoll. Beispielsweise erfordert die Rechenvorschrift 3 in <CorComp> die Angabe von zwei Zeichen in <Comp>. Wird eine unzulässige Parameterkombination angegeben, wird im <Status> ein entsprechender Fehlercode zurückgegeben.

## Beispiele

### Beispiel 1

| Programmcode | Kommentar |
|---|---|
| N10 DEF REAL _CORVAL[3] | |
| N20 $TC_DP1[1,1]=120 | ; Fräswerkzeug |
| N30 $TC_DP3[1,1]=10.0 | ; Geometrie L1 |
| N40 $TC_DP12[1,1]=1.0 | ; Verschleiß L1 |
| N50 _CORVAL[0]=0.333 | |
| N60 T1 D1 G17 G0 | |
| N70 R1=**SETTCOR(_CORVAL,"G",0,0,2)** | |
| N80 T1 D1 X0 Y0 Z0 | ; ==> MKS-Position X0.000 Y0.000 **Z1.333** |
| N90 M30 | |

<CorComp> ist "0", deshalb wird der in Z-Richtung wirkende Koordinatenwert der Geometrie-Komponente durch den Korrekturwert 0,333 ersetzt.

Die resultierende Werkzeuggesamtlänge ist somit: L1 = 0,333 + 1,000 = 1,333

### Beispiel 2

| Programmcode | Kommentar |
|---|---|
| N10 DEF REAL _CORVAL[3] | |
| N20 $TC_DP1[1,1]=120 | ; Fräswerkzeug |
| N30 $TC_DP3[1,1]=10.0 | ; Geometrie L1 |
| N40 $TC_DP12[1,1]=1.0 | ; Verschleiß L1 |
| N50 _CORVAL[0]=0.333 | |
| N60 T1 D1 G17 G0 | |
| N70 R1=**SETTCOR(_CORVAL,"W",0,1,2)** | |
| N80 T1 D1 X0 Y0 Z0 | ; ==> MKS-Position X0.000 Y0.000 Z11.333 |
| N90 M30 | |

<CorComp> ist "1", deshalb wird der in Z-Richtung wirkende Korrekturwert von 0,333 zum Verschleißwert von 1,0 addiert.

Die resultierende Werkzeuggesamtlänge ist somit: L1 = 10,0 + 1,333 = 11,333

### Beispiel 3

| Programmcode | Kommentar |
|---|---|
| N10 DEF REAL _CORVAL[3] | |
| N20 $TC_DP1[1,1]=120 | ; Fräswerkzeug |
| N30 $TC_DP3[1,1]=10.0 | ; Geometrie L1 |
| N40 $TC_DP12[1,1]=1.0 | ; Verschleiß L1 |
| N50 _CORVAL[0]=0.333 | |
| N60 T1 D1 G17 G0 | |
| N70 R1=**SETTCOR(_CORVAL,"GW",0,2,2)** | |
| N80 T1 D1 X0 Y0 Z0 | ; ==> MKS-Position X0.000 Y0.000 Z0.333 |
| N90 M30 | |

<CorComp> ist "2", deshalb wird die in Z-Richtung wirkende Korrektur in die Geometrie-Komponente eingetragen (der alte Wert wird überschrieben), und der Verschleißwert wird gelöscht.

Die resultierende Werkzeuggesamtlänge ist somit: L1 = 0,333 + 0,0 = 0,333

### Beispiel 4

| Programmcode | Kommentar |
|---|---|
| N10 DEF REAL _CORVAL[3] | |
| N20 $TC_DP1[1,1]=120 | ; Fräswerkzeug |
| N30 $TC_DP3[1,1]=10.0 | ; Geometrie L1 |
| N40 $TC_DP12[1,1]=1.0 | ; Verschleiß L1 |
| N50 _CORVAL[0]=0.333 | |
| N60 T1 D1 G17 G0 | |
| N70 R1=**SETTCOR(_CORVAL,"GW",0,3,2)** | |
| N80 T1 D1 X0 Y0 Z0 | ; ==> MKS-Position X0.000 Y0.000 Z11.333 |
| N90 M30 | |

<CorComp> ist "3", deshalb werden Verschleißwert und Korrekturwert zur Geometrie-Komponente addiert und die Verschleißkomponente gelöscht.

Die resultierende Werkzeuggesamtlänge ist somit: L1 = 11,333 + 0,0 = 11,333.

### Beispiel 5

| Programmcode | Kommentar |
|---|---|
| N10 DEF REAL _CORVAL[3] | |
| N20 $TC_DP1[1,1]=120 | ; Fräswerkzeug |
| N30 $TC_DP3[1,1]=10.0 | ; Geometrie L1 |
| N40 $TC_DP12[1,1]=1.0 | ; Verschleiß L1 |
| N50 _CORVAL[0]=0.333 | |
| N60 T1 D1 G17 G0 | |
| N70 R1=**SETTCOR(_CORVAL,"GW",0,3,0)** | |
| N80 T1 D1 X0 Y0 Z0 | ; ==> MKS-Position X0.333 Y0.000 Z11.000 |
| N90 M30 | |

<CorComp> ist "3" wie im vorigen Beispiel, die Korrektur wirkt jetzt jedoch auf die Geometrieachse mit dem Index "0" (X-Achse), der wegen G17 bei einem Fräswerkzeug die Werkzeugkomponente L3 zugeordnet ist. Der Aufruf von SETTCOR beeinflusst deshalb die Werkzeugparameter $TC_DP3 und $TC_DP12 nicht. Der Korrekturwert wird stattdessen in $TC_DP5 eingetragen.

### Beispiel 6

| Programmcode | Kommentar |
|---|---|
| N10 DEF REAL _CORVAL[3] | |
| N20 $TC_DP1[1,1]=500 | ; Drehwerkzeug |
| N30 $TC_DP3[1,1]=10.0 | ; Geometrie L1 |
| N40 $TC_DP4[1,1]=15.0 | ; Geometrie L2 |
| N50 $TC_DP12[1,1]=10.0 | ; Verschleiß L1 |

| **Programmcode** | **Kommentar** |
|---|---|
| N60 $TC_DP13[1,1]=0.0 | ; Verschleiß L2 |
| N70 _CORVAL[0]=5.0 | |
| N80 ROT Y-30 | |
| N90 T1 D1 G18 G0 | |
| N100 R1=**SETTCOR(_CORVAL,"GW",0,3,1)** | |
| N110 T1 D1 X0 Y0 Z0 | ; ==> MKS-Position X24.330 Y0.000 Z17.500 |
| N120 M30 | |

Das Werkzeug ist ein Drehwerkzeug. In Satz N80 wird eine Framedrehung aktiviert, sodass das Basiskoordinatensystem (BKS) gegenüber dem Werkstückkoordinatensystem (WKS) gedreht ist. Der Korrekturwert (N70) wirkt im WKS auf die Geometrieachse mit dem Index "1", d. h., da G18 aktiv ist, auf die X-Achse. Da <CorMode> = 3 gilt, muss der Werkzeugverschleiß in Richtung der X-Achse des WKS nach dem Ausführen von N100 null werden.

Der Inhalt der relevanten Werkzeugparameter ist deshalb am Programmende:

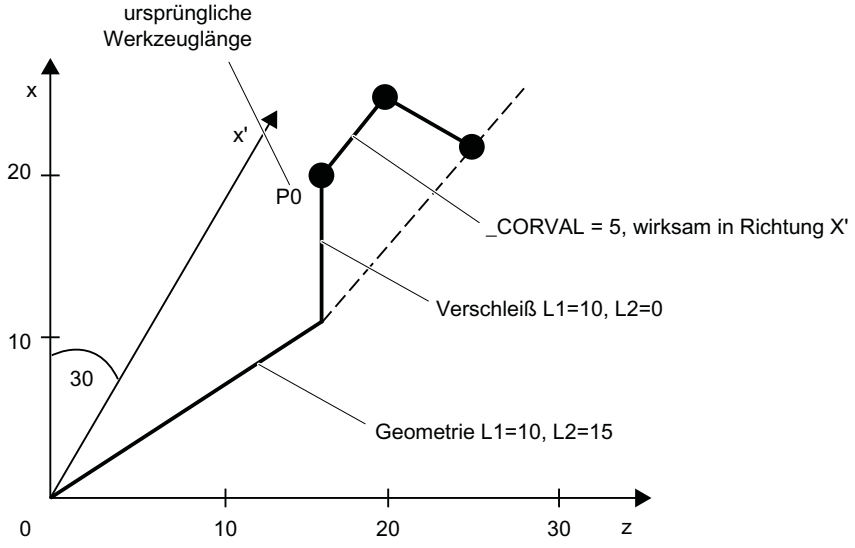
$TC_DP3[1,1] : 21.830 ; Geometrie L1

$TC_DP4[1,1] : 21.830 ; Geometrie L2

$TC_DP12[1,1] : 2.500 ; Verschleiß L1

$TC_DP13[1,1] : -4.330 ; Verschleiß L2

Die geometrischen Verhältnisse sind im folgenden Bild dargestellt. Der gesamte Verschleiß einschließlich _CORVAL wird auf die X'-Richtung im WKS projiziert. Das ergibt den Punkt P2. Die Koordinaten dieses Punkts (gemessen in X-Y-Koordinaten) werden in die Geometriekomponente des Werkzeugs eingetragen. Im Verschleiß bleibt der Differenzvektor $P_2$ - $P_1$ übrig. Damit hat der Verschleiß keine Komponente mehr in Richtung von _CORVAL.




Setzt man das Beispielprogramm nach N110 mit den folgenden Anweisungen fort, wird der restliche Verschleiß vollständig in die Geometrie übernommen, da die Korrektur jetzt in der Z'-Achse wirkt (Parameter <GeoAx> = 0):

```
N120 _CORVAL[0]=0.0
```

```
N130 R1=SETTCOR(_CORVAL,"GW",0,3,0)
N140 T1 D1 X0 Y0 Z0 ; ==> MKS-Position X24.330 Y0.000 Z17.500
```

Da der neue Korrekturwert "0" ist, darf sich die Werkzeuggesamtlänge und damit auch die in N140 angefahrene Position nicht verändern. Wäre _CORVAL in N120 ungleich "0", würde sich eine neue Werkzeuggesamtlänge und damit auch eine veränderte Position in N140 ergeben, der Verschleißanteil der Werkzeuglänge wäre jedoch in jedem Fall null, d. h., die gesamte Werkzeuglänge ist anschließend in jedem Fall im Geometrieanteil des Werkzeugs enthalten.

Das gleiche Ergebnis wie mit dem zweimaligen Aufruf der Funktion SETTCOR mit dem Parameter <CorComp> = 0 erreicht man auch durch einmaligen Aufruf mit <CorComp> = 1 (vektorielle Korrektur):

| **Programmcode** | **Kommentar** |
|---|---|
| N10 DEF REAL _CORVAL[3] | |
| N20 $TC_DP1[1,1]=500 | ; Drehwerkzeug |
| N30 $TC_DP3[1,1]=10.0 | ; Geometrie L1 |
| N40 $TC_DP4[1,1]=15.0 | ; Geometrie L2 |
| N50 $TC_DP12[1,1]=10.0 | ; Verschleiß L1 |
| N60 $TC_DP13[1,1]=0.0 | ; Verschleiß L2 |
| N70 _CORVAL[0]=0.0 | |
| N71 _CORVAL[1]=5.0 | |
| N72 _CORVAL[2]=0.0 | |
| N80 ROT Y-30 | |
| N90 T1 D1 G18 G0 | |
| N100 R1=**SETTCOR(_CORVAL,"GW",1,3,1)** | |
| N110 T1 D1 X0 Y0 Z0 | ; ==> MKS-Position X24.330 Y0.000 Z17.500 |
| N120 M30 | |

In diesem Fall sind alle Verschleißkomponenten des Werkzeugs sofort nach dem ersten Aufruf von SETTCOR in N100 gleich null.

### Beispiel 7

| **Programmcode** | **Kommentar** |
|---|---|
| N10 DEF REAL _CORVAL[3] | |
| N20 $TC_DP1[1,1]=500 | ; Drehwerkzeug |
| N30 $TC_DP3[1,1]=10.0 | ; Geometrie L1 |
| N40 $TC_DP4[1,1]=15.0 | ; Geometrie L2 |
| N50 $TC_DP12[1,1]=10.0 | ; Verschleiß L1 |
| N60 $TC_DP13[1,1]=0.0 | ; Verschleiß L2 |
| N70 _CORVAL[0]=5.0 | |
| N80 ROT Y-30 | |
| N90 T1 D1 G18 G0 | |
| N100 R1=**SETTCOR(_CORVAL,"GW",3,3)** | |
| N110 T1 D1 X0 Y0 Z0 | ; ==> MKS-Position X25.000 Y0.000 Z15.000 |
| N120 M30 | |

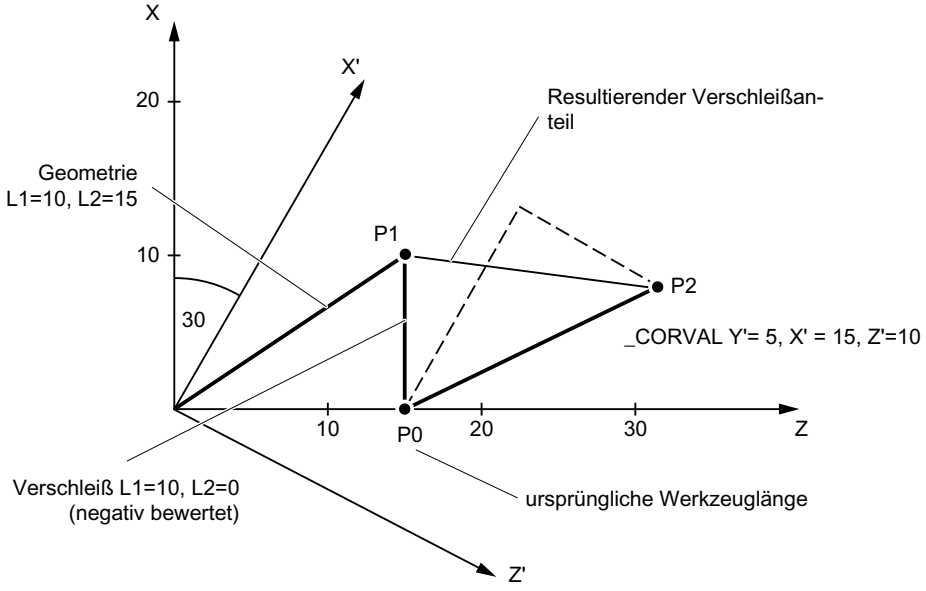
Gegenüber Beispiel 6 ist hier der Parameter <CorComp> = 3, deshalb kann die Angabe des Parameters <GeoAx> entfallen. Der in _CORVAL[0] enthaltene Wert wirkt jetzt unmittelbar auf

die Werkzeugkomponente L1, die Drehung in N80 hat auf das Ergebnis keinen Einfluss, die Verschleißkomponenten in $TC_DP12 wird zusammen mit _CORVAL[0] in den Geometrieanteil übernommen, sodass wegen $TC_DP13 die gesamte Werkzeuglänge bereits nach dem ersten Aufruf von SETTCOR in N100 im Geometrieteil des Werkzeugs steht.

### Beispiel 8

```
Programmcode                                Kommentar
N10 DEF REAL _CORVAL[3]
N20 $TC_DP1[1,1]=500                        ; Drehwerkzeug
N30 $TC_DP3[1,1]=10.0                       ; Geometrie L1
N40 $TC_DP4[1,1]=15.0                       ; Geometrie L2
N50 $TC_DP5[1,1]=20.0                       ; Geometrie L3
N60 $TC_DP12[1,1]=10.0                      ; Verschleiß L1
N70 $TC_DP13[1,1]=0.0                       ; Verschleiß L2
N80 $TC_DP14[1,1]=0.0                       ; Verschleiß L3
N90 $SC_WEAR_SIGN=TRUE
N100 _CORVAL[0]=10.0
N110 _CORVAL[1]=15.0
N120 _CORVAL[2]=5.0
N130 ROT Y-30
N140 T1 D1 G18 G0
N150 R1=SETTCOR(_CORVAL,"W",1,1)
N160 T1 D1 X0 Y0 Z0                         ; ==> MKS-Position X7.990 Y25.000 Z31.160
N170 M30
```

In N90 ist das Settingdatum SD42930 $SC_WEAR_SIGN gesetzt, d. h., der Verschleiß muss mit negativem Vorzeichen bewertet werden. Die Korrektur ist vektoriell (<CorComp> = 1), und der Korrekturvektor muss zum Verschleiß addiert werden (<CorMode> = 1). Die geometrischen Verhältnisse in der Z-X-Ebene sind im folgenden Bild dargestellt:



Wegen <CorMode> = 1 bleibt der Geometrieanteil des Werkzeugs unverändert. Der im WKS (Drehung um y-Achse) definierte Korrekturvektor muss so in den Verschleißanteil übernommen werden, dass die gesamte Werkzeuglänge in Bild 3 auf den Punkt $P_2$ verweist. Der resultierende Verschleißanteil des Werkzeugs ist deshalb durch die Distanz der beiden Punkte $P_1$ und $P_2$ gegeben.
Da der Verschleiß wegen des Settingdatums SD42930 $SC_WEAR_SIGN aber negativ bewertet wird, muss die so ermittelte Korrektur mit negativem Vorzeichen in den Korrekturspeicher eingetragen werden. Der Inhalt der relevanten Werkzeugparameter ist am Programmende deshalb:

$TC_DP3[1,1] : 10.000 ; Geometrie L1 (unverändert)

$TC_DP4[1,1] : 15.000 ; Geometrie L2 (unverändert)

$TC_DP5[1,1] : 10.000 ; Geometrie L3 (unverändert)

$TC_DP12[1,1] : 2.010 ; Verschleiß L1 (= 10 - 15 * cos(30) + 10 * sin(30))

$TC_DP13[1,1] : -16.160 ; Verschleiß L2 (= -15 * sin(30) - 10 * cos(30))

$TC_DP14[1,1] : -5.000 ; Verschleiß L3

An der L3-Komponente in Y-Richtung lässt sich die Wirkung des Settingdatums SD42930 $SC_WEAR_SIGN ohne die zusätzliche Komplizierung durch die Framedrehung erkennen.

## Weitere Informationen

### Dreh- / Schleifwerkzeuge: Berechnung der Werkzeuglänge in Abhängigkeit von MD20360 $MC_TOOL_PARAMETER_DEF_MASK

Wie bei Dreh- / Schleifwerkzeugen der Verschleiß bzw. die Werkzeuglänge in einer eventuell vorhandenen Durchmesserachse zu bewerten ist, wird festgelegt im Maschinendatum:

MD20360 $MC_TOOL_PARAMETER_DEF_MASK.<Bit> = <Wert>

| <Bit> | <Wert> | Bedeutung |
|---|---|---|
| 0 | 0 | Bei Dreh- / Schleifwerkzeugen wird der **Verschleißparameter** der Planachse im **Radiuswert** eingerechnet: |
| | 1 | Bei Dreh- / Schleifwerkzeugen wird der **Verschleißparameter** der Planachse als **Durchmesserwert** eingerechnet: |
| 1 | 0 | Bei Dreh- / Schleifwerkzeugen wird die **Werkzeuglängenkomponente** der Planachse als **Radiuswert** eingerechnet: |
| | 1 | Bei Dreh- / Schleifwerkzeugen wird die **Werkzeuglängenkomponente** der Planachse als **Durchmesserwert** eingerechnet: |

Sind die betreffenden Bits gesetzt, wird der zugehörige Eintrag mit dem Faktor 0,5 bewertet. Die Korrektur mittels SETTCOR wird so ausgeführt, dass die gesamte effektive Werkzeuglängenänderung gleich dem in <CorVal> übergebenen Wert ist. Wird eine Länge auf Grund des Maschinendatums MD20360 $MC_TOOL_PARAMETER_DEF_MASK bei der Längenberechnung mit dem Faktor 0,5 bewertet, muss die Korrektur dieser Komponente deshalb mit dem doppelten übergebenen Wert erfolgen.

### Beispiel

MD20360 $MC_TOOL_PARAMETER_DEF_MASK = 2 (Werkzeuglänge muss in der Durchmesserachse mit dem Faktor 0,5 bewertet werden)

Achse X ist die Durchmesserachse.

```
Programmcode                                Kommentar
N10 DEF REAL _LEN[11]
N20 DEF REAL _CORVAL[3]
N30 $TC_DP1[1,1]=500                        ; Werkzeugtyp
N40 $TC_DP2[1,1]=2                          ; Schneidenlage
N50 $TC_DP3[1,1]=3.                         ; Geometrie - Länge 1
N60 $TC_DP4[1,1]=4.                         ; Geometrie - Länge 2
N70 $TC_DP5[1,1]=5.                         ; Geometrie - Länge 3
N80 _CORVAL[0]=1.
N90 _CORVAL[1]=1.
N100 _CORVAL[2]=1.
N110 T1 D1 G18 G0 X0 Y0 Z0                  ; ==> MKS-Position X1.5 Y5 Z4
N120 R1=SETTCOR(_CORVAL,"G",1,1)
N130 T1 D1 X0 Y0 Z0                         ; ==> MKS-Position X2.5 Y6 Z5
N140 R3=$TC_DP3[1,1]                        ; = 5. = (3.000 + 2.*1.000)
N150 R4=$TC_DP4[1,1]                        ; = 5. = (4.000 + 1.000)
N160 R5=$TC_DP5[1,1]                        ; = 6. = (5.000 + 1.000)
N170 M30
```

Die Korrektur der Werkzeuglänge soll in jeder Achse 1 mm sein (N80 bis N100). In den Längen L2 und L3 wird deshalb zur Originallänge jeweils 1 mm addiert. Zur Originalwerkzeuglänge in L1 wird dagegen der doppelte Korrekturwert (2 mm) addiert, damit sich die Gesamtwerkzeuglänge wie verlangt um 1 mm ändert. Im Vergleich der angefahrenen Positionen in den Sätzen N110 und N130 erkennt man, dass sich jede Achsposition um 1 mm verändert hat.

## 3.13.14 Zuordnung der Werkzeuglängen L1, L2, L3 zu den Koordinatenachsen lesen (LENTOAX)

Die Funktion LENTOAX liefert Informationen über die Zuordnung der Werkzeuglängen L1, L2 und L3 des **aktiven** Werkzeugs zu Abszisse, Ordinate und Applikate. Die Zuordnung von Abszisse, Ordinate und Applikate zu den Geometrieachsen wird durch Frames und die aktive Ebene (G17 - G19) beeinflusst.

Betrachtet wird dabei nur der Geometrieanteil eines Werkzeugs ($TC_DP3[<t>,<d>] bis $TC_DP5[<t>,<d>]), d. h., eine gegebenenfalls davon abweichende Achszuordnung weiterer Komponenten (z. B. Verschleiß) hat auf das Ergebnis keinen Einfluss.

### Syntax

```
<Status> = LENTOAX(<AxInd>, <Matrix>[, <Coord>])
```

## Prinzip

| | | | |
|---|---|---|---|
| LENTOAX(...): | Vordefinierte Funktion zum Lesen der Zuordnung der Werkzeuglängen L1, L2 und L3 des aktiven Werkzeugs zu den Koordinatenachsen | | |
| | Alleine im Satz: | | ja |
| <Status>: | Rückgabewert der Funktion. Negative Werte zeigen Fehlerzustände an. | | |
| | Datentyp: | | INT |
| | Wert: | 0 | Funktion OK<br>Information in <AxInd> reicht zur Beschreibung aus (alle Werkzeuglängenkomponenten sind parallel zu Geometrieachsen). |
| | | 1 | Funktion ist OK, zu einer korrekten Beschreibung muss aber der Inhalt von <Matrix> ausgewertet werden (die Werkzeuglängenkomponenten sind nicht parallel zu den Geometrieachsen). |
| | | -1 | Ungültiger String im Parameter <Coord>. |
| | | -2 | Kein Werkzeug aktiv. |
| **Parameter** | | | |
| 1 <AxInd>: | Sind die Werkzeuglängenkomponenten parallel zu den Geometrieachsen, werden die Achsindizes, die den Längenkomponenten L1 bis L3 zugeordnet sind, im Feld <AxInd> zurückgeliefert:<br>• <AxInd> [0]: Abszisse<br>• <AxInd> [1]: Ordinate<br>• <AxInd> [2]: Applikate | | |
| | Datentyp: | | INT[3] |
| | Wert: | 0 | Keine Zuordnung vorhanden (Achse existiert nicht) |
| | | 1 ... 3<br>bzw.<br>-1 ... -3 | Nummer der Länge, die in der entsprechende Koordinatenachse wirkt.<br>Das Vorzeichen ist negativ, wenn die Werkzeuglängenkomponente in negative Koordinatenrichtung zeigt. |
| | Sind nicht alle Längenkomponenten parallel bzw. antiparallel zu den Geometrieachsen, wird in <AxInd> jeweils der Index der Achse zurückgegeben, die den größten Anteil einer Werkzeuglängenkomponente enthält. In diesem Fall (falls die Funktion nicht aus einem anderen Grund einen Fehler liefert) ist der Rückgabewert <Status> = 1. Die Abbildung der Werkzeuglängenkomponenten L1 bis L3 auf die Geometrieachsen 1 bis 3 wird dann durch den Inhalt des 2. Parameters <Matrix> vollständig beschrieben. | | |
| 2 <Matrix>: | Matrix, die den Vektor der Werkzeuglängen (L1=1, L2=1, L3=1) in den Vektor der Koordinatenachsen (Abszisse, Ordinate, Applikate) abbildet, d. h., den Spalten sind die Werkzeuglängenkomponenten in der Reihenfolge L1, L2, L3 zugeordnet, den Zeilen die Achsen in der Reihenfolge Abszisse, Ordinate, Applikate. | | |
| | Datentyp: | | REAL |
| | In der Matrix sind immer alle Elemente gültig, auch dann, wenn die zu einer Koordinatenachse gehörende Geometrieachse nicht vorhanden ist, d. h. wenn der entsprechende Eintrag in <AxInd> null ist. | | |

| | | | | |
|---|---|---|---|---|
| 3 | <Coord>: | Koordinatensystem, für das die Zuordnung gilt (optional) | | |
| | | Datentyp: | | STRING |
| | | Zeichen: | MCS<br>M | Abbildung der Werkzeuglänge in das Maschinenkoordinatensystem |
| | | | BCS<br>B | Abbildung der Werkzeuglänge in das Basiskoordinatensystem |
| | | | WCS<br>W | Abbildung der Werkzeuglänge in das Werkstückkoordinatensystem (Default) |
| | | | KCS<br>K | Abbildung der Werkzeuglänge in das Werkzeugkoordinatensystem der kinematischen Transformation |
| | | | TCS<br>T | Abbildung der Werkzeuglänge in das Werkzeugkoordinatensystem |
| | | Die Schreibweise der Zeichen im String (groß oder klein) ist beliebig.<br>Wird der Parameter <Coord> nicht angegeben, wird das WKS verwendet (Default). | | |

**Hinweis**

Im TCS sind immer alle Werkzeuglängenkomponenten parallel oder antiparallel zu den Achsen.

Antiparallel können die Komponenten nur dann sein, wenn Spiegeln aktiv ist und das folgende Settingdatum gesetzt ist:

SD42900 $SC_MIRROR_TOOL_LENGTH (Vorzeichenwechsel Werkzeuglänge beim Spiegeln)

## Beispiel

Standardfall Fräswerkzeug bei G17.

L1 wirkt in Z (Applikate), L2 wirkt in Y (Ordinate), L3 wirkt in X (Abszisse).

Aufruf der Funktion in der Form:
```
<Status>=LENTOAX(<AxInd>,<Matrix>,"WCS")
```

Der Ergebnisparameter <AxInd> enthält dann die Werte:

<AxInd>[0] = 3

<AxInd>[1] = 2

<AxInd>[2] = 1

oder kurz: ( 3, 2, 1)

Die zugehörige Matrix (<Matrix>) ist in diesem Fall:

$$\text{<Matrix>} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Ein Wechsel von G17 nach G18 oder G19 ändert am Ergebnis nichts, da die Zuordnung der Längenkomponenten zu den Geometrieachsen sich in gleicher Weise ändert wie die Zuordnung von Abszisse, Ordinate und Applikate.

Es wird nun bei aktivem G17 eine Framedrehung um Z von 60 Grad programmiert, z. B.:
```
ROT Z60
```

Die Richtung der Applikate (Z-Richtung) bleibt unverändert, der Hauptanteil von L2 liegt nun in Richtung der neuen X-Achse, der Hauptanteil von L1 in Richtung der negativen Y-Achse. Der Rückgabewert (<Status>) ist deshalb "1", <AxInd> enthält die Werte (2, -3, 1).

Die zugehörige Matrix (<Matrix>) ist in diesem Fall:

$$\text{<Matrix>} = \begin{pmatrix} 0 & \sin 60° & \cos 60° \\ 0 & \cos 60° & -\sin 60° \\ 1 & 0 & 0 \end{pmatrix}$$

# 3.14 Bahnverhalten

## 3.14.1 Vorschubverlauf (FNORM, FLIN, FCUB, FPO)

Zur flexibleren Vorgabe des Vorschubverlaufs wird die Vorschubprogrammierung nach DIN 66025 um lineare und kubische Verläufe erweitert.

Die kubischen Verläufe können direkt oder als interpolierende Splines programmiert werden. Hierdurch lassen sich - abhängig von der Krümmung des zu bearbeitenden Werkstücks - kontinuierlich glatte Geschwindigkeitsverläufe programmieren.

Diese Geschwindigkeitsverläufe ermöglichen ruckfreie Beschleunigungsänderungen und hierdurch Fertigung gleichmäßiger Werkstückoberflächen.

### Syntax

```
F… FNORM
F… FLIN
F… FCUB
F=FPO(…,…,…)
```

### Bedeutung

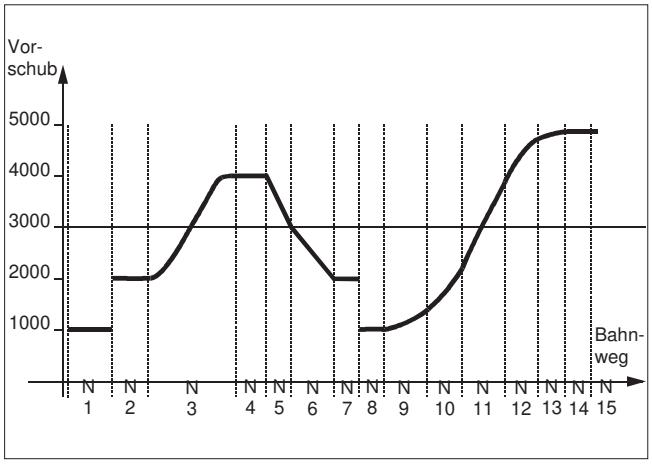
| | |
|---|---|
| FNORM: | Grundeinstellung. Der Vorschubwert wird über den Bahnweg des Satzes vorgegeben und gilt danach als modaler Wert. |
| FLIN: | Bahngeschwindigkeitsprofil **linear:**<br>Der Vorschubwert wird vom aktuellen Wert am Satzanfang bis zum Satzende über den Bahnweg linear eingefahren und gilt danach als modaler Wert. Dieses Verhalten kann mit G93 und G94 kombiniert werden. |
| FCUB: | Bahngeschwindigkeitsprofil **kubisch:**<br>Die satzweise programmierten F-Werte werden - bezogen auf den Satzendpunkt - durch einen Spline verbunden. Der Spline beginnt und endet tangential zur vorhergehenden bzw. nachfolgenden Vorschubangabe und wirkt mit G93 und G94.<br>Fehlt in einem Satz die F-Adresse, so wird hierfür der zuletzt programmierte F-Wert verwendet. |
| F=FPO… : | Bahngeschwindigkeitsprofil **über Polynom:**<br>Die F-Adresse bezeichnet den Vorschubverlauf über ein Polynom vom aktuellen Wert bis zum Satzende. Der Endwert gilt danach als modaler Wert. |

#### Vorschuboptimierung bei gekrümmten Bahnstücken

Vorschub-Polynom F=FPO und Vorschubspline FCUB sollten immer mit konstanter Schnittgeschwindigkeit CFC abgefahren werden. Hierdurch lässt sich ein beschleunigungsstetiges Sollvorschubprofil erzeugen.

## Beispiel: Verschiedene Vorschubprofile

In diesem Beispiel finden Sie die Programmierung und grafische Darstellung verschiedener Vorschubprofile.

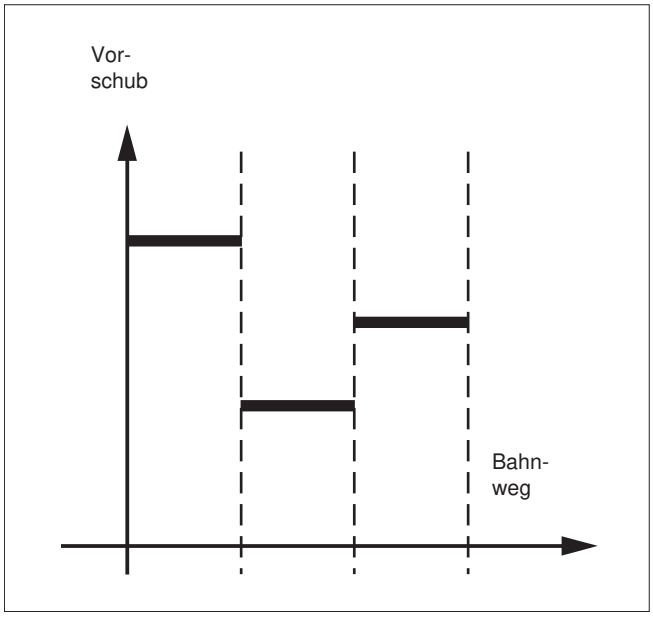



| **Programmcode** | **Kommentar** |
|---|---|
| N1 F1000 FNORM G1 X8 G91 G64 | ; Konstantes Vorschubprofil, Kettenmaßangabe |
| N2 F2000 X7 | ; Sprunghafte Sollgeschwindigkeitsänderung |
| N3 F=FPO(4000, 6000, -4000) | ; Vorschubprofil über Polynom mit Vorschub 4000 am Satzende. |
| N4 X6 | ; Polynomvorschub 4000 gilt als modaler Wert. |
| N5 F3000 FLIN X5 | ; Lineares Vorschubprofil |
| N6 F2000 X8 | ; Lineares Vorschubprofil |
| N7 X5 | ; Linearer Vorschub gilt als modaler Wert |
| N8 F1000 FNORM X5 | ; Konstantes Vorschubprofil mit sprunghafter Beschleunigungsänderung. |
| N9 F1400 FCUB X8 | ; Alle folgenden satzweise programmierten F-Werte werden mit Splines verbunden. |
| N10 F2200 X6 | |
| N11 F3900 X7 | |
| N12 F4600 X7 | |
| N13 F4900 X5 | ; Splineprofil ausschalten. |
| N14 FNORM X5 | |
| N15 X20 | |

## Weitere Informationen

### FNORM

Die Vorschubadresse F bezeichnet den Bahnvorschub als konstanten Wert nach DIN 66025.

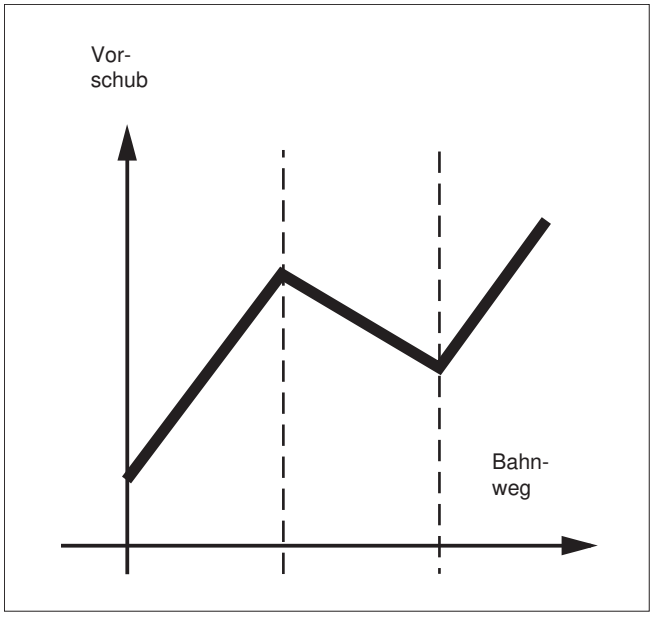



**FLIN**

Der Vorschubverlauf wird vom aktuellen Vorschubwert zum programmierten F-Wert linear bis Satzende eingefahren.

Beispiel:

```
N30 F1400 FLIN X50
```



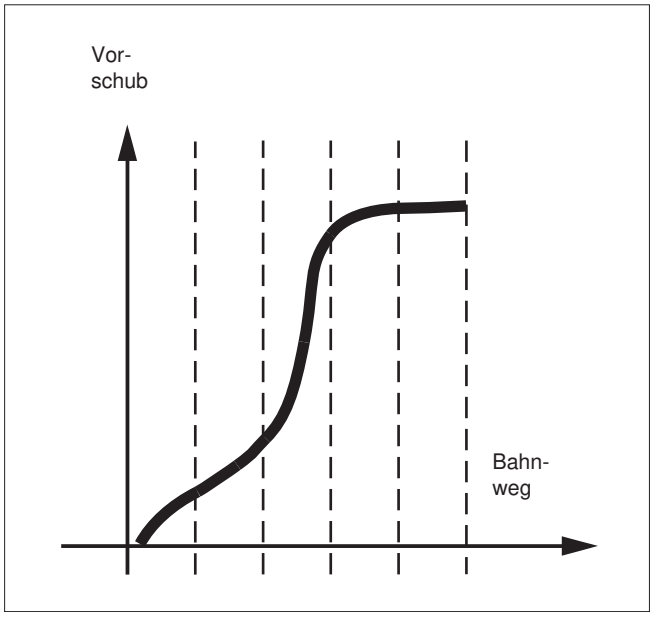

**FCUB**

Der Vorschub wird vom aktuellen Vorschubwert zum programmierten F-Wert bis Satzende im kubischen Verlauf eingefahren. Die Steuerung verbindet alle mit aktivem FCUB satzweise

programmierten Vorschubwerte durch Splines. Die Vorschubwerte dienen hier als Stützpunkte zur Berechnung der Splineinterpolation.

Beispiel:

```
N50 F1400 FCUB X50
N60 F2000 X47
N70 F3800 X52
```




### F=FPO(...,...,...)

Der Vorschubverlauf wird über ein Polynom direkt programmiert. Die Angabe der Polynomkoeffizienten erfolgt analog zur Polynominterpolation.

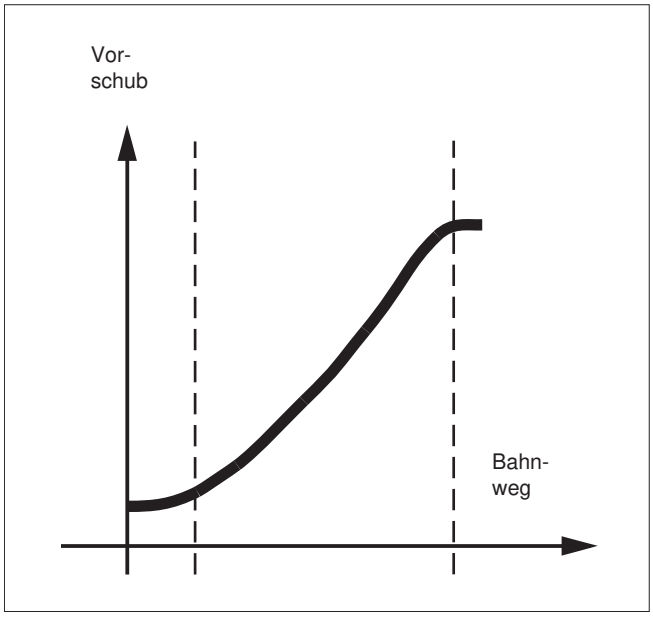
Beispiel:

```
F=FPO(endfeed, quadf, cubf)
```

endfeed, quadf und cubf sind vorher definierte Variable.

| | |
|---|---|
| endfeed: | Vorschub am Satzende |
| quadf: | Quadratischer Polynomkoeffizient |
| cubf: | Kubischer Polynomkoeffizient |

Bei aktivem FCUB schließt der Spline am Satzanfang und Satzende tangential an den über FPO festgelegten Verlauf an.




## Randbedingungen

- Unabhängig vom programmierten Vorschubverlauf gelten die Funktionen zur Programmierung des Bahnfahrverhaltens.
- Der programmierbare Vorschubverlauf gilt grundsätzlich absolut - unabhängig von G90 oder G91.
- Der Vorschubverlauf FLIN und FCUB wirkt mit G93 und G94, **nicht** bei G95, G96/G961 und G97/G971.
- Bei aktivem Kompressor COMP... und Zusammenfassung mehrerer Sätze zu einem Splinesegment gilt Folgendes:

| | |
|---|---|
| `FNORM`: | Für das Splinesegment gilt das F-Wort des letzten zugehörigen Satzes. |
| `FLIN`: | Für das Splinesegment gilt das F-Wort des letzten zugehörigen Satzes.<br>Der programmierte F-Wert gilt zum Ende des Segments und wird dann linear angefahren. |
| `FCUB`: | Der erzeugte Vorschubspline weicht maximal um den im Maschinendatum MD20172 $MC_COMPRESS_VELO_TOL definierten Wert von den programmierten Endpunkten ab. |
| `F=FPO(…,…,…)`: | Diese Sätze werden nicht komprimiert. |

## 3.14.2 Beschleunigungsverhalten

### 3.14.2.1 Beschleunigungsmodus (BRISK, BRISKA, SOFT, SOFTA, DRIVE, DRIVEA)

Zur Programmierung des Beschleunigungsmodus stehen folgende Teileprogrammbefehle zur Verfügung:

- "BRISK, BRISKA"
  Die Einzelachsen bzw. die Bahnachsen verfahren mit maximaler Beschleunigung bis zum Erreichen der programmierten Vorschubgeschwindigkeit (**Beschleunigung ohne Ruckbegrenzung**).
- "SOFT, SOFTA"
  Die Einzelachsen bzw. die Bahnachsen verfahren mit stetiger Beschleunigung bis zum Erreichen der programmierten Vorschubgeschwindigkeit (**Beschleunigung mit Ruckbegrenzung**).
- "DRIVE, DRIVEA"
  Die Einzelachsen bzw. die Bahnachsen verfahren mit maximaler Beschleunigung bis zu einer projektierten Geschwindigkeitsgrenze (MD-Einstellung!). Danach erfolgt eine Beschleunigungsreduktion (MD-Einstellung!) bis zum Erreichen der programmierten Vorschubgeschwindigkeit.

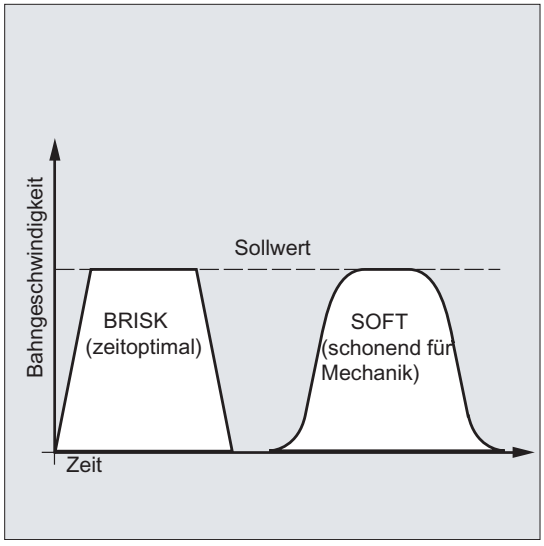



Bild 3-12 Verlauf der Bahngeschwindigkeit bei `BRISK` und `SOFT`

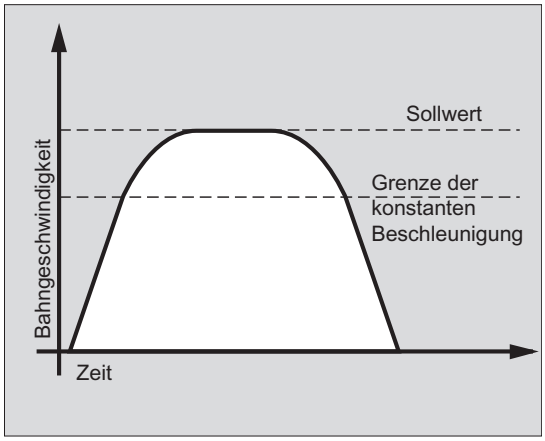



Bild 3-13 Verlauf der Bahngeschwindigkeit bei DRIVE

## Syntax

```
BRISK
BRISKA(<Achse1>,<Achse2>,…)
SOFT
SOFTA(<Achse1>,<Achse2>,…)
DRIVE
DRIVEA(<Achse1>,<Achse2>,…)
```

## Bedeutung

| | |
|---|---|
| BRISK: | Befehl zum Einschalten der "Beschleunigung ohne Ruckbegrenzung" für die Bahnachsen. |
| BRISKA: | Befehl zum Einschalten der "Beschleunigung ohne Ruckbegrenzung" für Einzelachsbewegungen (JOG, JOG/INC, Positionierachse, Pendelachse, etc.). |
| SOFT: | Befehl zum Einschalten der "Beschleunigung mit Ruckbegrenzung" für die Bahnachsen. |
| SOFTA: | Befehl zum Einschalten der "Beschleunigung mit Ruckbegrenzung" für Einzelachsbewegungen (JOG, JOG/INC, Positionierachse, Pendelachse, etc.). |
| DRIVE: | Befehl zum Einschalten der reduzierten Beschleunigung oberhalb einer projektierten Geschwindigkeitsgrenze (MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT) für die Bahnachsen. |
| DRIVEA: | Befehl zum Einschalten der reduzierten Beschleunigung oberhalb einer projektierten Geschwindigkeitsgrenze (MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT) für Einzelachsbewegungen (JOG, JOG/INC, Positionierachse, Pendelachse, etc.). |
| (<Achse1>,<Achse2>,…): | Einzelachsen, für die der aufgerufene Beschleunigungsmodus gelten soll. |

## Randbedingungen

### Wechsel des Beschleunigungsmodus während Bearbeitung

Wenn in einem Teileprogramm der Beschleunigungsmodus während der Bearbeitung gewechselt wird (BRISK ↔ SOFT), dann erfolgt auch bei Bahnsteuerbetrieb am Übergang ein Satzwechsel mit Genauhalt am Satzende.

## Beispiele

### Beispiel 1: SOFT und BRISKA

```
Programmcode
N10 G1 X… Y… F900 SOFT
N20 BRISKA(AX5,AX6)
...
```

### Beispiel 2: DRIVE und DRIVEA

```
Programmcode
N05 DRIVE
N10 G1 X… Y… F1000
N20 DRIVEA (AX4, AX6)
...
```

## 3.14.2.2 Beeinflussung der Beschleunigung bei Folgeachsen (VELOLIMA, ACCLIMA, JERKLIMA)

Bei Achskopplungen (Seite 908) (Tangentiale Nachführung, Mitschleppen, Leitwertkopplung, Elektronisches Getriebe) werden Folgeachsen/-spindeln abhängig von einer oder mehreren Leitachsen/-spindeln verfahren.

Die Dynamikbegrenzungen der Folgeachsen/-spindeln können mit den Funktionen VELOLIMA, ACCLIMA und JERKLIMA aus dem Teileprogramm oder aus Synchronaktionen beeinflusst werden, auch bei bereits aktiver Achskopplung.

### Hinweis

Die Funktion JERKLIMA ist nicht für alle Kopplungsarten verfügbar.

## Syntax

```
VELOLIMA(<Achse>)=<Wert>
ACCLIMA(<Achse>)=<Wert>
JERKLIMA(<Achse>)=<Wert>
```

## Bedeutung

| | |
|---|---|
| VELOLIMA: | Befehl zur Korrektur der parametrierten Maximal**geschwindigkeit** |
| ACCLIMA: | Befehl zur Korrektur der parametrierten Maximal**beschleunigung** |
| JERKLIMA: | Befehl zur Korrektur des parametrierten Maximal**ruck**s |
| <Achse>: | Folgeachse, deren Dynamikbegrenzungen korrigiert werden sollen |
| <Wert>: | Prozentualer Korrekturwert |

## Beispiele

### Beispiel 1: Korrektur der Dynamikbegrenzungen für eine Folgeachse (AX4)

```
Programmcode                Kommentar
...
VELOLIMA[AX4]=75            ; Begrenzungskorrektur auf 75% der im Maschinendatum hin-
                              terlegten axialen Maximalgeschwindigkeit.
ACCLIMA[AX4]=50             ; Begrenzungskorrektur auf 50% der im Maschinendatum hin-
                              terlegten axialen Maximalbeschleunigung.
JERKLIMA[AX4]=50            ; Begrenzungskorrektur auf 50% des im Maschinendatum hin-
                              terlegten axialen Maximalrucks bei Bahnbewegung.
...
```

### Beispiel 2: Elektronisches Getriebe

Achse 4 wird über eine Kopplung "Elektronisches Getriebe" an Achse X gekoppelt. Das Beschleunigungsvermögen der Folgeachse wird auf 70 % der maximalen Beschleunigung begrenzt. Die maximal zulässige Geschwindigkeit wird auf 50 % der maximalen Geschwindigkeit begrenzt. Nach erfolgter Einschaltung der Kopplung wird die maximal zulässige Geschwindigkeit wieder auf 100 % gesetzt.

```
Programmcode                          Kommentar
...
N120 ACCLIMA[AX4]=70                  ; Reduzierte maximale Beschleunigung.
N130 VELOLIMA[AX4]=50                 ; Reduzierte maximale Geschwindigkeit.
...
N150 EGON(AX4,"FINE",X,1,2)           ; Einschalten der EG-Kopplung.
...
N200 VELOLIMA[AX4]=100                ; Volle Maximalgeschwindigkeit.
...
```

### Beispiel 3: Leitwertkopplung per statische Synchronaktion beeinflussen

Achse 4 wird mittels Leitwertkopplung an X gekoppelt. Das Beschleunigungsverhalten wird per statische Synchronaktion 2 ab Position 100 auf 80 % begrenzt.

```
Programmcode                                    Kommentar
...
N120 IDS=2 WHENEVER $AA_IM[AX4] >               ; Synchronaktion
100 DO ACCLIMA[AX4]=80
```

| Programmcode | Kommentar |
|---|---|
| N130 LEADON(AX4, X, 2) | ; Leitwertkopplung ein |
| ... | |

### 3.14.2.3 Aktivierung von Technologie-spezifischen Dynamikwerten (DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN, DYNFINISH, DYNPREC)

Mit den Befehlen der G-Gruppe 59 "Dynamikmodus für Bahninterpolation" kann für unterschiedliche technologische Bearbeitungsschritte die dazu passende Dynamik aktiviert werden.

Dynamikwerte und G-Befehle sind projektierbar und damit von Maschinendateneinstellungen abhängig.

**Weitere Informationen:** Funktionshandbuch Basisfunktionen

## Syntax

**Dynamikwerte aktivieren:**
```
DYNNORM/DYNPOS/DYNROUGH/DYNSEMIFIN/DYNFINISH/DYNPREC
```

**Hinweis**

Die Dynamikwerte werden bereits in dem Satz wirksam, in dem der zugehörige G-Befehl programmiert wird. Es folgt kein Bearbeitungsstopp.

**Bestimmtes Feldelement lesen oder schreiben:**
```
R<m>=$MA...[n,X]
$MA...[n,X]=<Wert>
```

## Bedeutung

| | |
|---|---|
| DYNNORM: | **Normale Dynamik** aktivieren |
| DYNPOS: | Dynamik für **Positionierbetrieb, Gewindebohren** aktivieren |
| DYNROUGH: | Dynamik für **Schruppen** aktivieren |
| DYNSEMIFIN: | Dynamik für **Vorschlichten** aktivieren |
| DYNFINISH: | Dynamik für **Schlichten** aktivieren |
| DYNPREC: | Dynamik für **Feinschlichten** aktivieren |
| | |
| R<m>: | Rechenparameter mit Nummer <m> |
| $MA...[n,X]: | Maschinendatum mit dynamikbestimmendem Feldelement |

| | | |
|---|---|---|
| <n>: | Feldindex | |
| | Wertebereich: | 0 ... 5 |
| | 0 | Normale Dynamik (DYNNORM) |
| | 1 | Dynamik für Positionierbetrieb (DYNPOS) |
| | 2 | Dynamik für Schruppen (DYNROUGH) |
| | 3 | Dynamik für Vorschlichten (DYNSEMIFIN) |
| | 4 | Dynamik für Schlichten (DYNFINISH) |
| | 5 | Dynamik für Feinschlichten (DYNPREC) |
| <X>: | Achsadresse | |
| <Wert>: | Dynamikwert | |

## Beispiele

### Beispiel 1: Dynamikwerte aktivieren

| Programmcode | Kommentar |
|---|---|
| DYNNORM G1 X10 | ; Grundstellung |
| DYNPOS G1 X10 Y20 Z30 F… | ; Positionierbetrieb, Gewindebohren |
| DYNROUGH G1 X10 Y20 Z30 F10000 | ; Schruppen |
| DYNSEMIFIN G1 X10 Y20 Z30 F2000 | ; Vorschlichten |
| DYNFINISH G1 X10 Y20 Z30 F1000 | ; Schlichten |
| DYNPREC G1 X10 Y20 Z30 F600 | ; Feinschlichten |

### Beispiel 2: Bestimmtes Feldelement lesen oder schreiben

Maximale Beschleunigung für das Schruppen, Achse X.

| Programmcode | Kommentar |
|---|---|
| R1=$MA_MAX_AX_ACCEL[2,X] | ; Lesen |
| $MA_MAX_AX_ACCEL[2,X]=5 | ; Schreiben |

## 3.14.3 Fahren mit Vorsteuerung (FFWON, FFWOF)

Durch die Vorsteuerung wird der geschwindigkeitsabhängige Nachlaufweg beim Bahnfahren gegen Null reduziert. Fahren mit Vorsteuerung ermöglicht höhere Bahngenauigkeit und damit bessere Fertigungsergebnisse.

## Syntax

```
FFWON
FFWOF
```

### Bedeutung

| | |
|---|---|
| FFWON: | Befehl zum **Ein**schalten der Vorsteuerung |
| FFWOF: | Befehl zum **Aus**schalten der Vorsteuerung |

**Hinweis**

Über Maschinendaten wird die Art der Vorsteuerung festgelegt und welche Bahnachsen vorgesteuert verfahren werden sollen.

Standard: Geschwindigkeitsabhängige Vorsteuerung

Option: Beschleunigungsabhängige Vorsteuerung

### Beispiel

```
Programmcode
N10 FFWON
N20 G1 X… Y… F900 SOFT
```

## 3.14.4 Programmierbare Konturgenauigkeit (CPRECON, CPRECOF)

Die Funktion "Programmierbare Konturgenauigkeit" reduziert den Bahnfehler an gekrümmten Konturen durch automatische Anpassung der Geschwindigkeit.

Die einzuhaltende Konturgenauigkeit wird abhängig von der Projektierung der Maschine (MD20470 $MC_MC_CPREC_WITH_FFW; siehe Angaben des Maschinenherstellers) entweder über das Settingdatum $SC_CONTPREC oder über die programmierte Konturtoleranz CTOL vorgegeben. Je kleiner der Wert und je kleiner der $K_V$-Faktor der Geometrieachsen, umso stärker wird der Bahnvorschub auf gekrümmten Konturen abgesenkt.

Die Funktion "Programmierbare Konturgenauigkeit" wird über die Anweisungen CPRECON und CPRECOF im NC-Programm ein- bzw. ausgeschaltet.

### Syntax

```
CPRECON
...
CPRECOF
```

### Bedeutung

| | | |
|---|---|---|
| CPRECON: | G-Befehlsaufruf: "Programmierbare Konturgenauigkeit" **ein**schalten | |
| | Wirksamkeit: | modal |
| CPRECOF: | G-Befehlsaufruf: "Programmierbare Konturgenauigkeit" **aus**schalten | |
| | Wirksamkeit: | modal |

CPRECON und CPRECOF bilden zusammen die G-Funktionsgruppe 39 (Programmierbare Konturgenauigkeit).

**Hinweis**

Über das Settingdatum $SC_MINFEED (Mindestbahnvorschub bei CPRECON) kann der Anwender eine Mindestgeschwindigkeit für den Bahnvorschub vorgeben.

Der Vorschub wird nicht unter diesen Wert begrenzt, es sei denn, ein niedrigerer F-Wert wurde programmiert oder die dynamischen Begrenzungen der Achsen erzwingen eine niedrigere Bahngeschwindigkeit.

**Hinweis**

Die Funktion "Programmierbare Konturgenauigkeit" betrachtet nur die Geometrieachsen der Bahn. Auf die Geschwindigkeiten von Positionierachsen hat sie keinen Einfluss.

## Beispiel

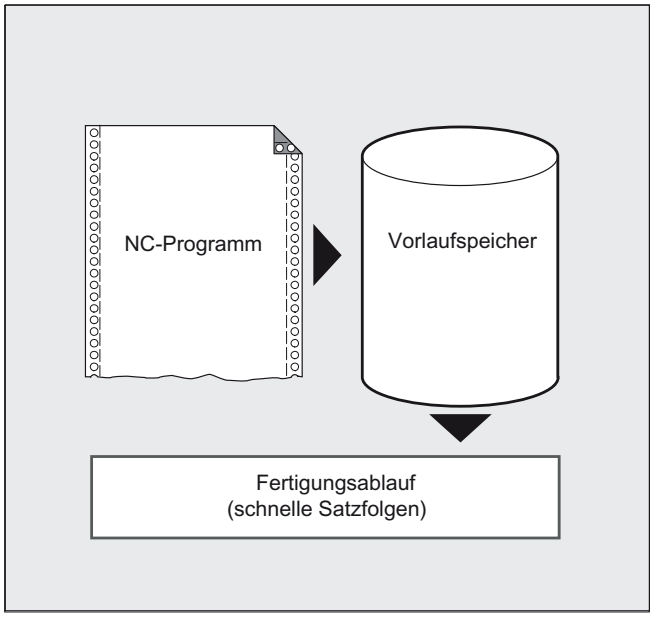
| **Programmcode** | **Kommentar** |
|---|---|
| N10 G0 X0 Y0 | |
| N20 CPRECON | ; Einschalten der "Programmierbaren Konturgenauigkeit". |
| N30 G1 G64 X100 F10000 | ; Bearbeitung mit 10 m/min im Bahnsteuerbetrieb. |
| N40 G3 Y20 J10 | ; Automatische Vorschubbegrenzung im Kreissatz. |
| N50 G1 X0 | ; Vorschub wieder ohne Begrenzung (10 m/min). |
| ... | |
| N100 CPRECOF | ; Ausschalten der "Programmierbaren Konturgenauigkeit". |
| N110 G0 ... | |

## Siehe auch

Kontur-/Orientierungtoleranz programmieren (CTOL, OTOL, ATOL) (Seite 880)

## 3.14.5 Programmablauf mit Vorlaufspeicher (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE)

Je nach Ausbaustufe verfügt die Steuerung über eine bestimmte Menge sog. Vorlaufspeicher, die fertig aufbereitete Sätze vor der Abarbeitung speichern und im Fertigungsablauf als schnelle Satzfolgen ausgeben. Hierdurch lassen sich kurze Wege mit hohen Geschwindigkeiten abfahren. Soweit die Restzeit der Steuerung es zulässt, wird der Vorlaufspeicher grundsätzlich gefüllt.




### Bearbeitungsabschnitt kennzeichnen

Der Bearbeitungsabschnitt, der im Vorlaufspeicher zwischengespeichert werden soll, wird im Teileprogramm am Anfang mit "STOPFIFO" und am Ende mit "STARTFIFO" gekennzeichnet. Die Abarbeitung der aufbereiteten und zwischengespeicherten Sätze beginnt erst nach dem Befehl "STARTFIFO" oder wenn der Vorlaufspeicher voll ist.

### Automatische Vorlaufspeichersteuerung

Die automatische Vorlaufspeichersteuerung wird mit dem Befehl "FIFOCTRL" aufgerufen. "FIFOCTRL" wirkt zunächst genauso wie "STOPFIFO". Bei jeder Programmierung wird gewartet, bis der Vorlaufspeicher voll ist, dann beginnt die Abarbeitung. Unterschiedlich ist dagegen das Verhalten beim Leerlaufen des Vorlaufspeichers: mit "FIFOCTRL" wird ab einem Füllstand von 2/3 die Bahngeschwindigkeit zunehmend reduziert, um ein komplettes Leerlaufen und ein Abbremsen bis zum Stillstand zu verhindern.

### Vorlaufstopp

Die Satzaufbereitung und -zwischenspeicherung wird angehalten, wenn im Satz der Befehl "STOPRE" programmiert ist. Der nachfolgende Satz wird erst dann ausgeführt, wenn alle

vorher aufbereiteten und gespeicherten Sätze vollständig abgearbeitet sind. Der vorherige Satz wird im Genauhalt angehalten (wie G9).

| ACHTUNG |
| --- |
| **Programmabbruch** |
| Bei eingeschalteter Werkzeugkorrektur und bei Spline-Interpolationen sollte kein "STOPRE" programmiert werden, da sonst zusammengehörige Satzfolgen unterbrochen werden. |

## Syntax

Tabelle 3-3 Bearbeitungsabschnitt kennzeichnen:

```
STOPFIFO
...
STARTFIFO
```

Tabelle 3-4 Automatische Vorlaufspeichersteuerung:

```
...
FIFOCTRL
...
```

Tabelle 3-5 Vorlaufstopp:

```
...
STOPRE
...
```

**Hinweis**

Die Befehle "STOPFIFO", "STARTFIFO", "FIFOCTRL" und "STOPRE" müssen im eigenen Satz programmiert werden.

### Bedeutung

| | |
|---|---|
| STOPFIFO: | "STOPFIFO" kennzeichnet den Beginn eines Bearbeitungsabschnitts, der im Vorlaufspeicher zwischengespeichert werden soll. Mit "STOPFIFO" wird die Bearbeitung angehalten und der Vorlaufspeicher gefüllt, bis:<br>• "STARTFIFO" oder "STOPRE" erkannt wird<br>oder<br>• der Vorlaufspeicher voll ist<br>oder<br>• das Programmende erreicht ist. |
| STARTFIFO: | Mit "STARTFIFO" startet die schnelle Abarbeitung des Bearbeitungsabschnitts, parallel dazu erfolgt das Auffüllen des Vorlaufspeichers |
| FIFOCTRL: | Einschalten der automatischen Vorlaufspeichersteuerung |
| STOPRE: | Vorlauf stoppen |

**Hinweis**

Das Auffüllen des Vorlaufspeichers wird nicht ausgeführt bzw. unterbrochen, wenn der Bearbeitungsabschnitt Befehle enthält, die einen ungepufferten Betrieb erzwingen (Referenzpunktfahren, Messfunktionen, …).

**Hinweis**

Beim Zugriff auf Zustandsdaten der Maschine ($SA...) erzeugt die Steuerung internen Vorlaufstopp.

### Beispiel: Vorlauf stoppen

```
Programmcode                                  Kommentar
...
N30 MEAW=1 G1 F1000 X100 Y100 Z50             ; Messsatz mit Messtaster des ersten
                                              Messeingangs und Geradeninterpolation.
N40 STOPRE                                    ; Vorlaufstopp.
...
```

## 3.14.6 Stop-Delay-Bereiche definieren (DELAYFSTON, DELAYFSTOF)

Zur Definition eines bedingt unterbrechbaren Bereichs im Teileprogramm (Stop-Delay-Bereich) dienen die vordefinierten Prozeduren DELAYFSTON und DELAYFSTOF.

**Hinweis**

In Synchronaktionen sind DELAYFSTON und DELAYFSTOF **nicht** zulässig!

## Syntax

```
DELAYFSTON
...
DELAYFSTOF
```

## Bedeutung

| | | |
|---|---|---|
| DELAYFSTON: | Beginn eines Stop-Delay-Bereichs definieren | |
| | Alleine im Satz: | ja |
| DELAYFSTOF: | Ende eines Stop-Delay-Bereichs definieren | |
| | Alleine im Satz: | ja |

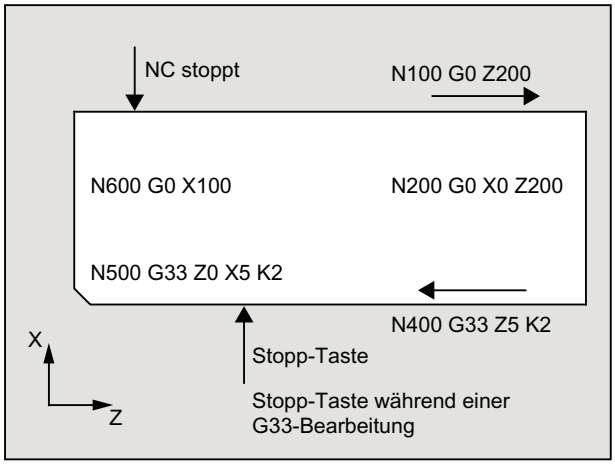
## Programmierbeispiel

In einer Schleife wird folgender Programmblock wiederholt:

**Programmcode**

```
...
N99 MY_LOOP:
N100 G0 Z200
N200 G0 X0 Z200
N300 DELAYFSTON
N400 G33 Z5 K2 M3 S1000
N500 G33 Z0 X5 K3
N600 G0 X100
N700 DELAYFSTOF
N800 GOTOB MY_LOOP
...
```

Im folgenden Bild ist erkennbar, dass der Anwender im Stop-Delay-Bereich "Stopp" drückt, und die NC beginnt den Bremsvorgang außerhalb des Stop-Delay-Bereichs, d. h. im Satz N100. Damit kommt die NC im vorderen Bereich von N100 zum Halten.



## Weitere Informationen

### Unterprogrammende

Mit dem Ende des Unterprogramms, in dem DELAYFSTON gerufen wurde, wird implizit DELAYFSTOF aktiviert.

### Schachtelung

Ruft Unterprogramm 1 in einem Stop-Delay-Bereich Unterprogramm 2, so ist Unterprogramm 2 komplett ein Stop-Delay-Bereich. Insbesondere ist DELAYFSTOF in Unterprogramm 2 wirkungslos.

Beispiel:

```
Programmcode                    Kommentar
N10010 DELAYFSTON               ; Sätze mit N10xxx Programmebene 1.
N10020 R1 = R1 + 1
N10030 G4 F1                    ; Stop-Delay-Bereich beginnt.
...
N10040 Unterprogramm2
...
...                             ; Interpretation des Unterprogramms 2.
N20010 DELAYFSTON               ; Unwirksam, wiederholter Beginn, 2. Ebene.
...
N20020 DELAYFSTOF               ; Unwirksam, Ende in anderer Ebene.
N20030 RET
N10050 DELAYFSTOF               ; Stop-Delay-Bereichsende in gleicher Ebene.
...
N10060 R2 = R2 + 2
N10070 G4 F1                    ; Stop-Delay-Bereich endet. Stopps wirken ab jetzt un-
                                mittelbar.
```

### Systemvariablen

Ob sich die Teileprogrammbearbeitung zum aktuellen Zeitpunkt in einem Stop-Delay-Bereich befindet, kann mittels folgender Systemvariablen abgefragt werden:

- im Teileprogramm mit $P_DELAYFST
- in Synchronaktionen mit $AC_DELAYFST

| Wert | Bedeutung |
|---|---|
| 0 | Stop-Delay-Bereich nicht aktiv |
| 1 | Stop-Delay-Bereich aktiv |

## 3.14.7 Programmstelle für SERUPRO verhindern (IPTRLOCK, IPTRUNLOCK)

Für bestimmte komplizierte mechanische Situationen an der Maschine ist es erforderlich, den Satzsuchlauf SERUPRO zu verhindern.

Mit einem programmierbaren Unterbrechungszeiger besteht eine Eingriffsmöglichkeit, beim "Suchen auf der Unterbrechungsstelle", vor der suchunfähigen Stelle aufzusetzen.

Es können auch suchunfähige Bereiche in Teileprogrammbereichen definiert werden, in denen die NC noch nicht wieder einsteigen kann. Mit dem Programmabbruch vermerkt der NC den zuletzt verarbeiteten Satz, auf den über die Bedienoberfläche HMI gesucht werden kann.

## Syntax

```
IPTRLOCK
IPTRUNLOCK
```

Die Befehle stehen allein in einer Teleprogrammzeile und ermöglichen einen programmierbaren Unterbrechungszeiger

## Bedeutung

| | |
|---|---|
| `IPTRLOCK`: | Beginn des suchunfähigen Programmabschnitts |
| `IPTRUNLOCK`: | Ende des suchunfähigen Programmabschnitts |

Beide Befehle sind nur in Teileprogrammen, **nicht** jedoch in Synchronaktionen zulässig.

## Beispiel

Verschachtelung suchunfähiger Programmabschnitte in zwei Programmebenen mit impliziten "IPTRUNLOCK". Das implizite "IPTRUNLOCK" in Unterprogramm 1 beendet den suchunfähigen Bereich.

| **Programmcode** | **Kommentar** |
|---|---|
| `N10010 IPTRLOCK()` | |
| `N10020 R1 = R1 + 1` | |
| `N10030 G4 F1` | `; Haltesatz, der suchunfähige Programmabschnitt beginnt.` |
| `...` | |
| `N10040 Unterprogramm2` | |
| `...` | `; Interpretation des Unterprogramms 2.` |
| `N20010 IPTRLOCK ()` | `; Unwirksam, wiederholter Beginn.` |
| `...` | |
| `N20020 IPTRUNLOCK ()` | `; Unwirksam, Ende in anderer Ebene.` |
| `N20030 RET` | |
| `...` | |
| `N10060 R2 = R2 + 2` | |
| `N10070 RET` | `; Ende des suchunfähigen Programmabschnitts.` |
| `N100 G4 F2` | `; Hauptprogramm wird fortgesetzt.` |

Eine Unterbrechung auf 100 liefert dann wieder der Unterbrechungszeiger.

## Weitere Informationen

### Suchunfähige Bereiche erfassen und suchen

Die suchunfähigen Programmabschnitte werden mit dem Sprachbefehlen "IPTRLOCK" und "IPTRUNLOCK" gekennzeichnet.

Der Befehl "IPTRLOCK" friert den Unterbrechungszeiger auf ein im Hauptlauf ausführbaren Einzelsatz (SB1) ein. Dieser Satz wird im Folgenden als Haltesatz bezeichnet. Tritt nach "IPTRLOCK" ein Programmabbruch ein, so kann auf der Bedienoberfläche HMI nach diesen sogenannten Haltesatz gesucht werden.

### Auf den aktuellen Satz wieder aufsetzen

Der Unterbrechungszeiger wird mit "IPTRUNLOCK" für den nachfolgenden Programmabschnitt auf den aktuellen Satz zum Unterbrechungspunkt gesetzt werden.

Nach einem gefundenen Suchziel kann mit dem selben Haltesatz ein neues Suchziel wiederholt werden.

Ein vom Benutzer editierter Unterbrechungszeiger, muss über HMI wieder entfernt werden.

### Regeln bei Schachtelung

Folgende Punkte regeln das Zusammenspiel der Sprachbefehle "IPTRLOCK" und "IPTRUNLOCK" mit Verschachtelungen und dem Unterprogrammende:

1. Mit dem Ende des Unterprogramms, in dem "IPTRLOCK" gerufen wurde, wird implizit "IPTRUNLOCK" aktiviert.
2. "IPTRLOCK" in einem suchunfähigen Bereich bleibt ohne Wirkung.
3. Ruft Unterprogramm1 in einem suchunfähigen Bereich Unterprogramm2, so bleibt Unterprogramm2 komplett suchunfähig. Insbesondere ist "IPTRUNLOCK" in Unterprogramm2 wirkungslos.

**Weitere Informationen:** Funktionshandbuch Basisfunktionen

### Systemvariable

Ein suchunfähiger Bereich kann mit "$P_IPTRLOCK" im Teileprogramm erkannt werden.

### Automatischer Unterbrechungszeiger

Die Funktion automatischer Unterbrechungszeiger legt automatisch eine vorher festgelegte Kopplungsart als suchunfähig fest. Mittels Maschinendatum wird für

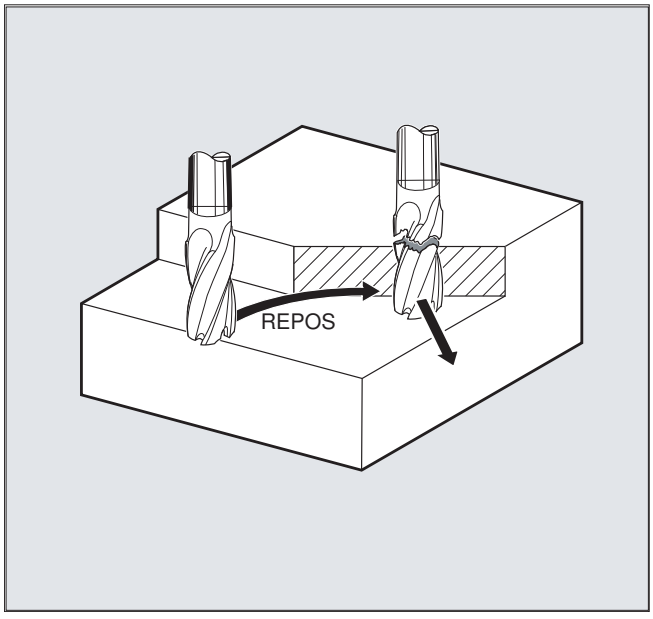
- Elektronisches Getriebe bei "EGON"
- Axiale Leitwertkopplung bei "LEADON"

der automatische Unterbrechungszeiger aktiviert. Überschneiden sich der programmierte und der über Maschinendatum aktivierbare automatische Unterbrechungszeiger, so wird der größtmögliche suchunfähige Bereich gebildet.

## 3.14.8 Wiederanfahren an Kontur (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)

Wenn Sie während der Bearbeitung das laufende Programm unterbrechen und das Werkzeug freifahren – z.B. wegen Werkzeugbruchs oder zum Vermessen des Werkstücks – können Sie anschließend die Kontur an einem wählbaren Punkt programmgesteuert wieder anfahren.

Der Befehl `REPOS` wirkt in einem ASUP wie ein Unterprogramm-Rücksprung (z.B. M17). Nachfolgende Sätze werden nicht mehr ausgeführt. Zur Unterbrechung des Programmlaufs siehe auch "Interruptroutine (ASUP) (Seite 547)".




### Syntax

```
REPOSA RMIBL DISPR=…
REPOSA RMBBL
REPOSA RMEBL
REPOSA RMNBL
REPOSL RMIBL DISPR=…
REPOSL RMBBL
REPOSL RMEBL
REPOSL RMNBL
REPOSQ RMIBL DISPR=… DISR=…
REPOSQ RMBBL DISR=…
REPOSQ RMEBL DISR=…
REPOSQA DISR=…
REPOSH RMIBL DISPR=… DISR=…
REPOSH RMBBL DISR=…
REPOSH RMEBL DISR=…
REPOSHA DISR=…
```

## Bedeutung

### Auswahl des Anfahrwegs

| | |
|---|---|
| REPOSA: | Wiederanfahren an die Kontur mit den Geometrieachsen auf einer Geraden.<br>Alle anderen Kanalachsen werden ebenfalls repositioniert. |
| REPOSL: | Wiederanfahren an die Kontur mit den Geometrieachsen auf einer Geraden .<br>Alle anderen Kanalachsen müssen explizit programmiert werden. |
| REPOSQ DISR=… : | Wiederanfahren an die Kontur mit den Geometrieachsen auf einem Viertelkreis mit Radius DISR.<br>Alle anderen Kanalachsen müssen explizit programmiert werden. |
| REPOSQA DISR=… : | Wiederanfahren an die Kontur mit den Geometrieachsen auf einem Viertelkreis mit Radius DISR.<br>Alle anderen Kanalachsen werden ebenfalls repositioniert. |
| REPOSH DISR=… : | Wiederanfahren an die Kontur mit den Geometrieachsen auf einem Halbkreis mit Durchmesser DISR.<br>Alle anderen Kanalachsen müssen explizit programmiert werden. |
| REPOSHA DISR=… : | Wiederanfahren an die Kontur mit den Geometrieachsen auf einem Halbkreis mit Radius DISR.<br>Alle anderen Kanalachsen werden ebenfalls repositioniert. |

### Auswahl des Wiederanfahrpunkts

| | |
|---|---|
| RMIBL: | Unterbrechungspunkt anfahren |
| RMIBL DISPR=…: | Eintrittspunkt im Abstand DISPR in mm/inch vor Unterbrechungspunkt |
| RMBBL: | Satzanfangspunkt anfahren |
| RMEBL: | Satzendpunkt anfahren |
| RMEBL DISPR=… : | Satzendpunkt anfahren im Abstand DISPR vor Endpunkt |
| RMNBL: | An den nächstliegenden Bahnpunkt anfahren |
| A0 B0 C0 : | Achsen, in denen angefahren werden soll |

### Hinweis

### Kompatibilität

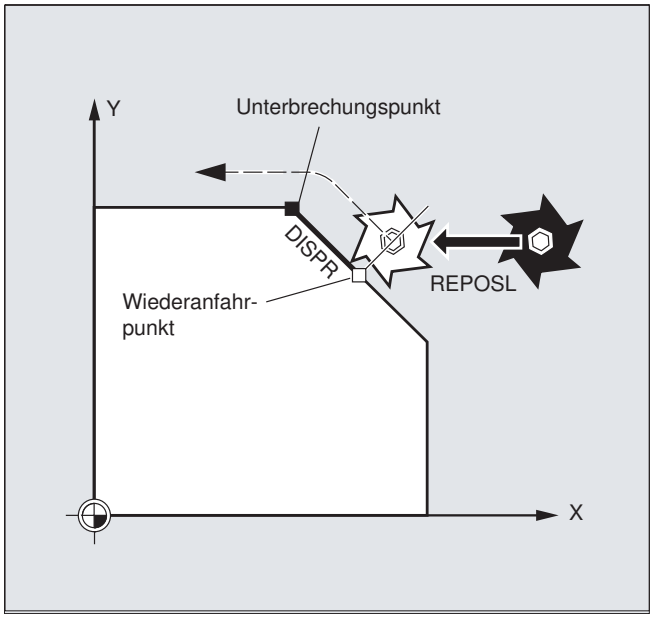
Um kompatibel mit älteren Softwareständen zu bleiben, können Sie den REPOS-Anfahrmodus auch weiterhin über die modalen Befehle RMI, RMB, RME und RMN programmieren. Bei Verwendung innerhalb eines ASUP sollte dieses mit dem Attribut SAVE in der PROC-Anweisung versehen werden. Andernfalls wirkt der im ASUP verwendete modale REPOS-Anfahrmodus, wenn er von der Voreinstellung RMI abweicht, auch bei nachfolgenden REPOS-Vorgänge.

## Wiederanfahren an die Kontur auf einer Geraden, REPOSA, REPOSL

Das Werkzeug fährt den Wiederanfahrpunkt direkt auf einer Geraden an.

### Beispiel

```
REPOSL RMIBL DISPR=6 F400
```

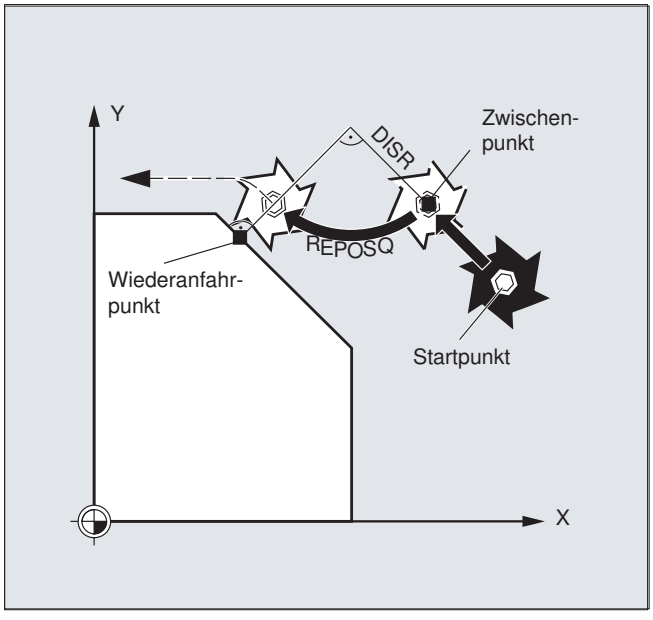



## Wiederanfahren an die Kontur mit Viertelkreis, REPOSQ, REPOSQA

Das Werkzeug fährt den Wiederanfahrpunkt auf einem Viertelkreis mit Radius `DISR=...` an. Den notwendigen Zwischenpunkt zwischen Start- und Wiederanfahrpunkt berechnet die Steuerung automatisch.

**Beispiel**

```
REPOSQ RMIBL DISR=10 F400
```

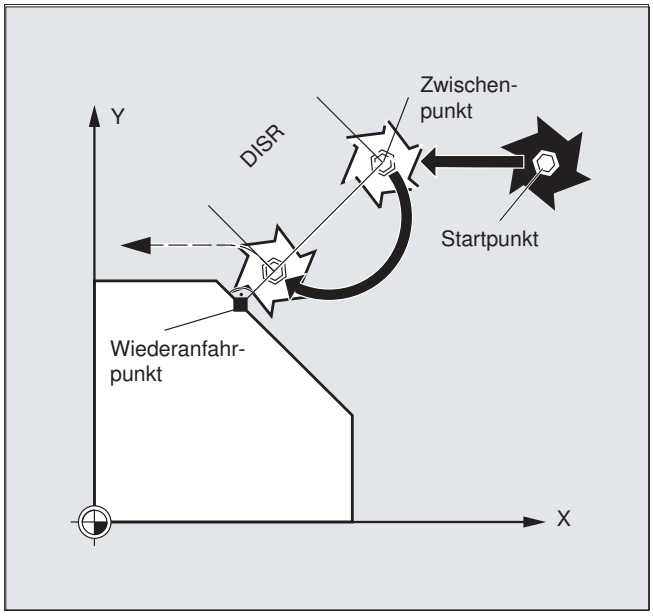


## Wiederanfahren an die Kontur mit Halbkreis, REPOSH, REPOSHA

Das Werkzeug fährt den Wiederanfahrpunkt auf einem Halbkreis mit Durchmesser `DISR=...` an. Den notwendigen Zwischenpunkt zwischen Start- und Wiederanfahrpunkt berechnet die Steuerung automatisch.

**Beispiel**

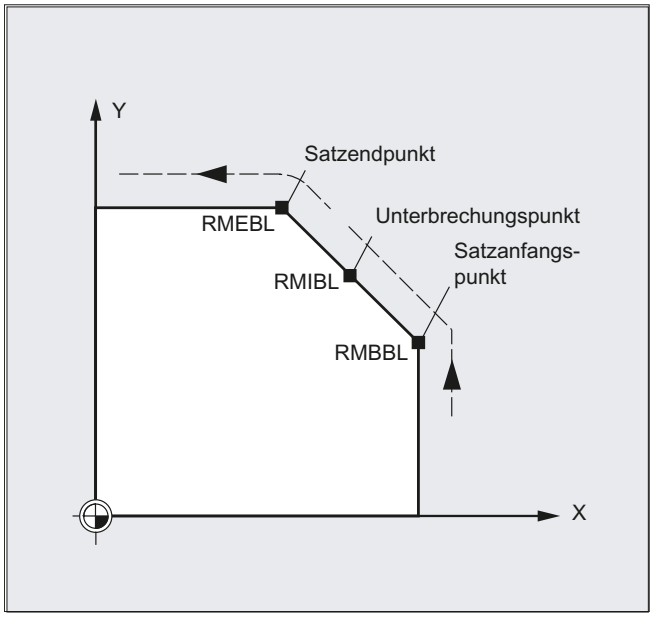
```
REPOSH RMIBL DISR=20 F400
```




## Wiederanfahrpunkt festlegen (nicht für SERUPRO Anfahren mit RMNBL)

Bezogen auf den NC-Satz, in dem der Programm-Ablauf unterbrochen wurde, können Sie zwischen drei Wiederanfahrpunkten wählen:

- RMIBL, Unterbrechungspunkt
- RMBBL, Satzanfangspunkt bzw. letzter Endpunkt
- RMEBL, Satzendpunkt




Mit RMIBL DISPR=... bzw. mit RMEBL DISPR=... können Sie einen Wiederanfahrpunkt festlegen, der vor dem Unterbrechungspunkt bzw. vor dem Satzendpunkt liegt.

Mit DISPR=... beschreiben Sie den Konturweg in mm/inch, um den der Wiederanfahrpunkt vor dem Unterbrechungs- bzw. Endpunkt liegt. Dieser Punkt kann - auch für größere Werte - maximal im Satzanfangspunkt liegen.

Wird kein DISPR=... programmiert, gilt DISPR=0 und damit der Unterbrechungspunkt (bei RMIBL) bzw. der Satzendpunkt (bei RMEBL).

## Vorzeichen von DISPR

Das Vorzeichen von DISPR wird ausgewertet. Bei positivem Vorzeichen ist das Verhalten wie bisher.

Bei negativem Vorzeichen wird hinter dem Unterbrechungspunkt bzw. bei RMBBL hinter dem Startpunkt wieder aufgesetzt.

Der Abstand Unterbrechungspunkt-Aufsetzpunkt ergibt sich aus dem Betrag von DISPR. Dieser Punkt kann auch für betragsmäßig größere Werte maximal im Satzendpunkt liegen.

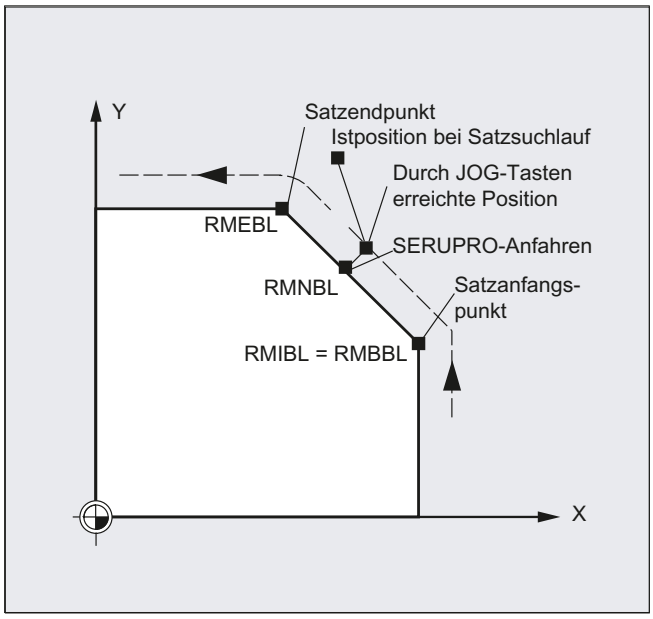
**Anwendungsbeispiel:**

Durch einen Sensor wird die Annäherung an eine Spannpratze erkannt. Es wird ein ASUP ausgelöst, mit dem die Spannpratze umfahren wird.

Anschließend wird mit negativem DISPR auf einen Punkt hinter der Spannpratze repositioniert und das Programm fortgesetzt.

## SERUPRO-Anfahren mit RMNBL

Wird bei der Bearbeitung an einer beliebigen Stelle ein Abbruch erzwungen, dann wird mit SERUPRO-Anfahren unter RMNBL der kürzeste Weg von der Abbruchstelle angefahren, um anschließend nur den Restweg abzuarbeiten. Dazu startet der Anwender ein SERUPRO-Vorgang auf den Unterbrechungssatz und positioniert mit den JOG-Tasten vor die schadhafte Stelle des Zielsatzes.



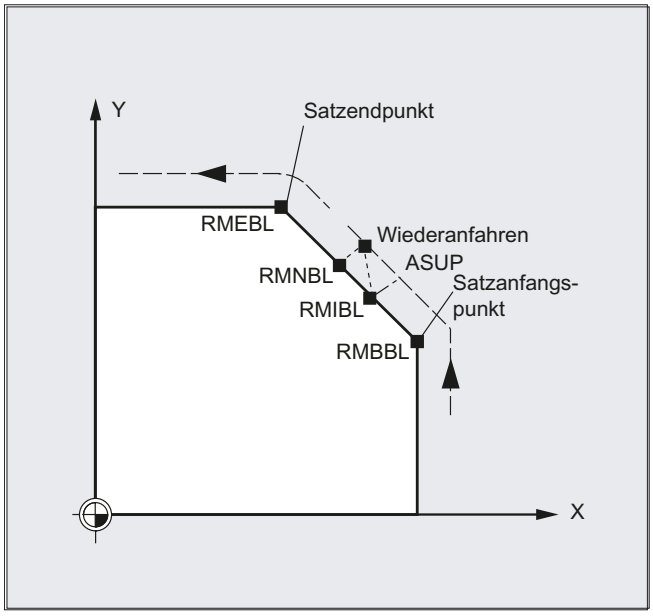

**Hinweis**

**SERUPRO**

Für SERUPRO ist RMIBL und RMBBL identisch. RMNBL ist nicht nur auf SERUPRO beschränkt, sondern allgemein gültig.

## Anfahren vom nächstliegenden Bahnpunkt RMNBL

Zum Interpretationszeitpunkt von REPOSA wird nach einer Unterbrechung der Wiederanfahrsatz mit RMNBL nicht noch einmal komplett begonnen, sondern nur der Restweg abgearbeitet. Es wird der nächstliegende Bahnpunkt des unterbrochenen Satzes angefahren.




### Status für den gültigen REPOS-Mode

Der gültige REPOS-Mode des unterbrochenen Satzes kann über Synchronaktionen mit der Variablen $AC_REPOS_PATH_MODE gelesen werden:

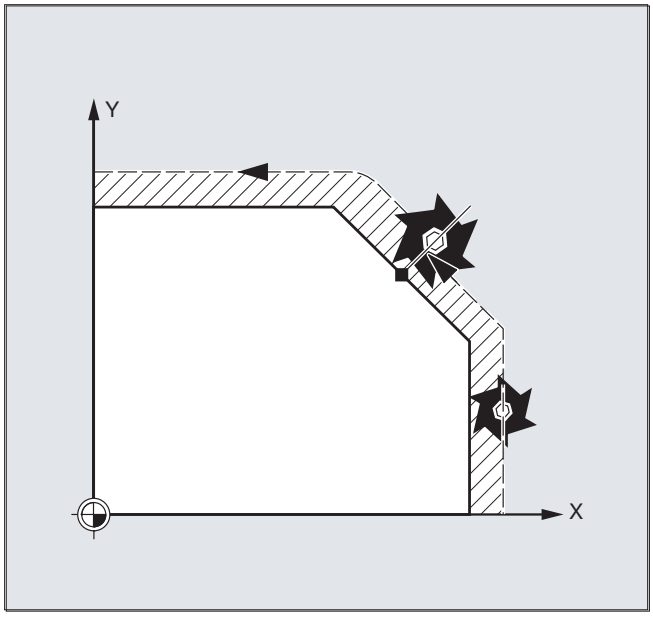
0 Anfahren nicht definiert
1 RMBBL: Anfahren auf den Beginn
2 RMIBL: Anfahren auf den Unterbrechungspunkt
3 RMEBL: Anfahren auf den Satzendpunkt
4 RMNBL: Anfahren auf den nächstliegenden Bahnpunkt des unterbrochenen Satzes

## Anfahren mit neuem Werkzeug

Falls Sie den Programmablauf wegen Werkzeugbruch gestoppt haben:

Mit Programmierung der neuen D-Nummer wird das Programm ab Wiederanfahrpunkt mit den geänderten Werkzeugkorrekturwerten fortgesetzt.

Bei geänderten Werkzeugkorrekturwerten kann der Unterbrechungspunkt möglicherweise nicht mehr angefahren werden. In diesem Fall wird der dem Unterbrechungspunkt nächstgelegene Punkt auf der neuen Kontur angefahren (gegebenenfalls um DISPR modifiziert).




## Kontur anfahren

Die Bewegung, mit der das Werkzeug wieder an die Kontur heranfährt, ist programmierbar. Die Adressen der zu verfahrenden Achsen geben Sie mit Wert Null an.

Mit den Befehlen REPOSA, REPOSQA und REPOSHA werden automatisch alle Achsen repositioniert. Es ist keine Achsangabe notwendig.

Bei Programmierung von REPOSL, REPOSQ und REPOSH fahren alle Geometrieachsen automatisch, also auch ohne Angabe im Befehl, an. Alle anderen Achsen müssen im Befehl angegeben werden.

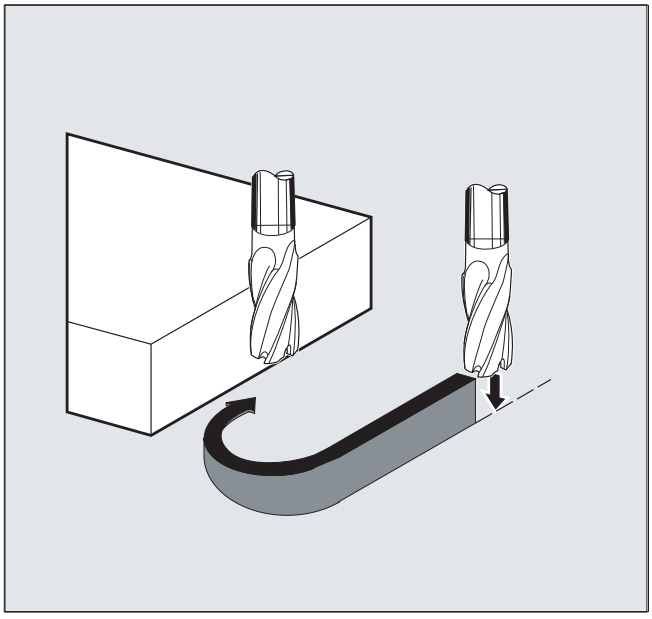
**Für die Kreisbewegungen REPOSH und REPOSQ gilt:**

Der Kreis wird in der angegebenen Arbeitsebene G17 bis G19 gefahren.

Falls Sie im Anfahrsatz die dritte Geometrieachse (Zustellrichtung) angeben, wird der Wiederanfahrpunkt für den Fall, dass Werkzeugposition und programmierte Position in Zustellrichtung nicht übereinstimmen, auf einer Schraubenlinie angefahren.

In folgenden Fällen wird automatisch auf lineares Anfahren REPOSL umgeschaltet:

- Sie haben keinen Wert für DISR angegeben.
- Es gibt keine definierte Anfahrrichtung (Programmunterbrechung in einem Satz ohne Verfahrinformation).
- Bei Anfahrrichtung senkrecht zur aktuellen Arbeitsebene.



## 3.14.9 Beeinflussung der Bewegungsführung

### 3.14.9.1 Maximalen Achsruck anpassen (JERKLIM)

Mit dem Befehl JERKLIM kann im Teileprogramm der per Maschinendatum eingestellte maximal mögliche Ruck einer Achse bei Bahnbewegung in kritischen Programmabschnitten reduziert oder überhöht werden.

#### Voraussetzung

Der Beschleunigungsmodus SOFT muss aktiv sein.

#### Wirksamkeit

JERKLIM wirkt:

- in den AUTOMATIK-Betriebsarten
- nur auf Bahnachsen

## Syntax

```
JERKLIM[<Ax>]=<Value>
```

## Bedeutung

| | | |
|---|---|---|
| JERKLIM: | Ruckgrenzwert anpassen | |
| <Ax>: | Maschinenachse, deren Ruckgrenzwert angepasst werden soll | |
| <Wert>: | Prozentualer Korrekturwert<br>Der Korrekturwert bezieht sich auf den projektierten maximalen Achsruck bei Bahnbewegung (MD32431 $MA_MAX_AX_JERK). | |
| | Wertebereich: | 1 ... 200 |
| | Der Wert 100 bewirkt keine Beeinflussung des Rucks. | |

**Hinweis**

**Verhalten bei Teileprogrammende und Kanal-Reset**

Das Verhalten von JERKLIM bei Teileprogrammende und Kanal-Reset ist abhängig von der Einstellung von Bit 0 im Maschinendatum MD32320 $MA_DYN_LIMIT_RESET_MASK.

## Beispiel

| **Programmcode** | **Kommentar** |
|---|---|
| N1000 G0 X0 Y0 F10000 SOFT G64 | |
| N1100 G1 X20 RNDM=5 ACC[X]=20 ACC[Y]=30 | |
| N1200 G1 Y20 JERKLIM[Y]=200 | ; Der Achsschlitten in Y-Richtung kann mit max. 200% des für die Achse zulässigen Rucks beschleunigt/verzögert werden. |
| N1300 G1 X0 JERKLIM[X]=2 | ; Der Achsschlitten in X-Richtung soll nur mit max. 2% des für die Achse zulässigen Rucks beschleunigt/verzögert werden. |
| N1400 G1 Y0 | |
| M30 | |

### 3.14.9.2 Maximale Achsgeschwindigkeit oder Spindeldrehzahl anpassen (VELOLIM)

Mit dem Befehl VELOLIM kann im Teileprogramm die per Maschinendatum eingestellte maximal mögliche Geschwindigkeit einer Achse bzw. die maximal mögliche getriebestufenabhängige Drehzahl einer Spindel reduziert werden.

## Wirksamkeit

VELOLIM wirkt:

- in den AUTOMATIK-Betriebsarten
- auf Bahn- und Positionierachsen
- auf Spindeln im Spindel-/Achsbetrieb

## Syntax

```
VELOLIM[<Ax>]=<Value>
```

## Bedeutung

| | | |
|---|---|---|
| `VELOLIM`: | Geschwindigkeits- oder Drehzahlgrenzwert anpassen | |
| `<Ax>`: | Achse oder Spindel, deren Geschwindigkeits- oder Drehzahlgrenzwert angepasst werden soll<br>Über MD30455 $MA_MISC_FUNCTION_MASK Bit 6 kann eingestellt werden, ob VELOLIM unabhängig von der aktuellen Verwendung als Spindel oder Achse wirkt (Bit 6 = 1) oder getrennt für jede Betriebsart programmierbar sein soll (Bit 6 = 0). Ist eine getrennte Wirkung projektiert, dann wird die Auswahl über den Bezeichner bei der Programmierung getroffen:<br>• Spindelbezeichner für Spindelbetriebsarten: `S<n>`<br>• Achsbezeichner für den Achsbetrieb, z. B.: `C` | |
| `<Value>`: | Prozentualer Korrekturwert<br>Bei Achsen bzw. bei Spindeln im Achsbetrieb mit der Einstellung MD30455 Bit 6 = 0 bezieht sich der Korrekturwert auf die projektierte maximale Achsgeschwindigkeit (MD32000 $MA_MAX_AX_VELO).<br>Bei Spindeln im Spindel- oder Achsbetrieb und der Einstellung MD30455 Bit 6 = 1 bezieht sich der Korrekturwert auf die Maximaldrehzahl der aktiven Getriebestufe (MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[<n>]). | |
| | Wertebereich: | 1 ... 100 |
| | Der Wert 100 bewirkt keine Beeinflussung der Geschwindigkeit bzw. Drehzahl. | |

**Hinweis**

**Verhalten bei Teileprogrammende und Kanal-Reset**

Das Verhalten von VELOLIM bei Teileprogrammende und Kanal-Reset ist abhängig von der Einstellung von Bit 0 im Maschinendatum MD32320 $MA_DYN_LIMIT_RESET_MASK.

**Hinweis**

**Drehzahlbegrenzung im Spindelbetrieb**

Eine Drehzahlbegrenzung durch VELOLIM (kleiner 100 %) kann im Spindelbetrieb über folgende Systemvariablen erkannt werden:

- $AC_SMAXVELO (maximal mögliche Spindeldrehzahl)
- $AC_SMAXVELO_INFO (Kennung für die drehzahlbegrenzende Ursache)

## Beispiele

### Beispiel 1: Geschwindigkeitsbegrenzung Maschinenachse

| Programmcode | Kommentar |
|---|---|
| ... | |
| N70 VELOLIM[X]=80 | ; Der Achsschlitten in X-Richtung soll nur mit maximal 80% der für die Achse zulässigen Geschwindigkeit verfahren werden. |
| ... | |

### Beispiel 2: Drehzahlbegrenzung Spindel 1 (AX5)

Projektierung:

- MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[ 1, AX5 ] = 1000
  (Maximaldrehzahl der Getriebstufe 1 = 1000 U/min)
- MD30455 $MA_MISC_FUNCTION_MASK[ AX5 ], Bit 6 = 1
  (Programmierung von VELOLIM wirkt gemeinsam für Spindel- und Achsbetrieb unabhängig vom programmierten Bezeichner)

Programmierung:

| Programmcode | Kommentar |
|---|---|
| N05 VELOLIM[S1]=90 | ; Begrenzung der Maximaldrehzahl von Spindel 1 auf 90% von 1000 U/min. |
| ... | |
| N50 VELOLIM[C]=45 | ; Begrenzung der Maximaldrehzahl von Spindel 1 auf 45 % von 1000 U/min, C sei der Achsbezeichner von S1. |
| ... | |

## 3.14.10 Kontur-/Orientierungtoleranz programmieren (CTOL, OTOL, ATOL)

Mit den Adressen CTOL, OTOL und ATOL können die über Maschinen- und Settingdaten parametrierten Bearbeitungstoleranzen für Kompressor-Funktionen, Überschleifen und Orientierungsglättung im Teileprogramm angepasst werden.

Die programmierten Toleranzwerte gelten, bis sie neu programmiert oder durch Zuweisung eines negativen Werts gelöscht werden. Sie werden ferner gelöscht bei Programmende bzw. Reset. Nach dem Löschen sind wieder die parametrierten Toleranzwerte wirksam.

## Syntax

```
CTOL=<Value>
OTOL=<Value>
ATOL[<Axis>]=<Value>
```

## Bedeutung

| | | | | |
|---|---|---|---|---|
| CTOL: | Adresse zum Programmieren der **Konturtoleranz** | | | |
| | Anwendungsbereich: | • alle Kompressor-Funktionen<br>• alle Überschleifarten außer G641 und G644 | | |
| | Vorlaufstopp: | nein | | |
| | Wirksamkeit: | modal | | |
| | <Value>: | Der Wert für die Konturtoleranz ist eine Längenangabe. | | |
| | | Typ: | REAL | |
| | | Einheit: | Inch/mm (abhängig von der aktuellen Einstellung der Maßangabe) | |
| | | Wertebereich: | ≥ 0: | Toleranzwert |
| | | | < 0: | Löschen des programmierten Toleranzwerts<br>⇒ Es gilt wieder der im Maschinen- bzw. Settingdatum parametrierte Toleranzwert. |
| OTOL: | Adresse zum Programmieren der **Orientierungstoleranz** | | | |
| | Anwendungsbereich: | • alle Kompressor-Funktionen<br>• Orientierungsglättung ORISON<br>• alle Überschleifarten außer G641, G644 und OSD | | |
| | Vorlaufstopp: | nein | | |
| | Wirksamkeit: | modal | | |
| | <Value>: | Der Wert für die Orientierungstoleranz ist eine Winkelangabe. | | |
| | | Typ: | REAL | |
| | | Einheit: | Grad | |
| | | Wertebereich: | ≥ 0: | Toleranzwert |
| | | | < 0: | Löschen des programmierten Toleranzwerts<br>⇒ Es gilt wieder der im Maschinen- bzw. Settingdatum parametrierte Toleranzwert. |

| | | | | |
|---|---|---|---|---|
| ATOL: | Adresse zum Programmieren einer **achsspezifischen Toleranz** | | | |
| | Anwendungsbereich: | • alle Kompressor-Funktionen<br>• Orientierungsglättung ORISON<br>• alle Überschleifarten außer G641, G644 und OSD | | |
| | Vorlaufstopp: | nein | | |
| | Wirksamkeit: | modal | | |
| | <Axis>: | Name der Kanalachse, für welche die programmierte Toleranz wirken soll | | |
| | <Value>: | Der Wert für die Achstoleranz ist je nach Achstyp (Linear- oder Rundachse) eine Längen- oder Winkelangabe. | | |
| | | Typ: | REAL | |
| | | Einheit: | für Linearachsen: | Inch/mm (abhängig von der aktuellen Einstellung der Maßangabe) |
| | | | für Rundachsen: | Grad |
| | | Wertebereich: | ≥ 0: | Toleranzwert |
| | | | < 0: | Löschen des programmierten Toleranzwerts<br>⇒ Es gilt wieder der im Maschinen- bzw. Settingdatum parametrierte Toleranzwert. |

**Hinweis**

Die mit CTOL und OTOL programmierten kanalspezifischen Toleranzwerte sind höherprior als die mit ATOL programmierten achsspezifischen Toleranzwerte.

**Hinweis**

**Skalierende Frames**

Skalierende Frames wirken auf die programmierten Toleranzen in gleicher Weise wie auf die Achspositionen, d. h. die relative Toleranz bleibt gleich.

## Beispiel

```
Programmcode                        Kommentar
COMPCAD G645 G1 F10000              ; Kompressor-Funktion COMPCAD aktivieren.
X... Y... Z...                      ; Hier wirken die Maschinen-und Settingdaten.
X... Y... Z...
X... Y... Z...
CTOL=0.02                           ; Ab hier wirkt eine Konturtoleranz von 0,02 mm.
X... Y... Z...
X... Y... Z...
X... Y... Z...
ASCALE X0.25 Y0.25 Z0.25            ; Ab hier wirkt eine Konturtoleranz von 0,005 mm.
X... Y... Z...
```

| **Programmcode** | **Kommentar** |
|---|---|
| X... Y... Z... | |
| X... Y... Z... | |
| **CTOL=-1** | ; Ab hier wirken wieder Maschinen- und Settingdaten. |
| X... Y... Z... | |
| X... Y... Z... | |
| X... Y... Z... | |

## Systemvariablen

### Lesen mit Vorlaufstopp

Über folgende Systemvariablen sind im Teileprogramm und Synchronaktion die aktuell wirksamen Toleranzen lesbar:

- $AC_CTOL
  Kanalspezifische Konturtoleranz, die bei der Aufbereitung des aktuellen Hauptlaufsatzes wirksam war.
  Wenn keine Konturtoleranz wirksam ist, liefert $AC_CTOL die Wurzel aus der Summe der Quadrate der Toleranzen der Geometrieachsen.
- $AC_OTOL
  Kanalspezifische Orientierungstoleranz, die bei der Aufbereitung des aktuellen Hauptlaufsatzes wirksam war.
  Wenn keine Orientierungstoleranz wirksam ist, liefert $AC_OTOL während einer aktiven Orientierungstransformation die Wurzel aus der Summe der Quadrate der Toleranzen der Orientierungsachsen, ansonsten den Wert "-1".
- $AA_ATOL[<Achse>]
  Achsspezifische Konturtoleranz, die bei der Aufbereitung des aktuellen Hauptlaufsatzes wirksam war.
  Wenn eine Konturtoleranz aktiv ist, liefert $AA_ATOL[<Geometrieachse>] die Konturtoleranz geteilt durch die Wurzel aus der Anzahl der Geometrieachsen.
  Wenn eine Orientierungstoleranz und eine Orientierungstransformation aktiv ist, liefert $AA_ATOL[<Orientierungsachse>] die Orientierungstoleranz geteilt durch die Wurzel aus der Anzahl der Orientierungsachsen.

### Hinweis

Wenn keine Toleranzwerte programmiert wurden, dann sind die $A-Variablen nicht differenziert genug, um die Toleranzen der einzelnen Funktionen zu unterscheiden.

Solche Fälle können auftreten, wenn die Maschinen- und Settingdaten unterschiedliche Toleranzen für Kompressor-Funktionen, Überschleifen und Orientierungsglättung einstellen. Die Systemvariablen liefern dann den größten Wert, der bei den gerade aktiven Funktionen auftritt. Wenn z. B. eine Kompressor-Funktion mit Orientierungstoleranz 0,1° und eine Orientierungsglättung ORISON mit 1° aktiv ist, liefert $AC_OTOL den Wert "1". Wird die Orientierungsglättung ausgeschaltet, liefert $AC_OTOL den Wert "0,1".

**Lesen ohne Vorlaufstopp**

Über folgende Systemvariablen sind im Teileprogramm die aktuell wirksamen Toleranzen lesbar:

- $P_CTOL
  Aktuell wirksame kanalspezifische Konturtoleranz.
- $P_OTOL
  Aktuell wirksame kanalspezifische Orientierungstoleranz.
- $PA_ATOL
  Aktuell wirksame achsspezifische Konturtoleranz.

### Randbedingungen

Die mit CTOL, OTOL und ATOL programmierten Toleranzen wirken auch auf Funktionen, die indirekt von diesen Toleranzen abhängen:

- Die Begrenzung des Sehnenfehlers bei der Sollwertberechnung
- Die Grundfunktionen des Freiformflächenmodus

Folgende Überschleif-Funktionen werden durch die Programmierung von CTOL, OTOL und ATOL **nicht** beeinflusst:

- Überschleifen der Orientierung mit OSD
  OSD verwendet keine Toleranz, sondern eine Distanz zum Satzübergang.
- Überschleifen mit G644
  G644 dient nicht der Bearbeitung, sondern der Optimierung von Werkzeugwechseln und anderen Bewegungen ohne Bearbeitung.
- Überschleifen mit G645
  G645 verhält sich fast immer wie G642 und verwendet somit die programmierten Toleranzen. Nur an tangentenstetigen Satzübergängen mit einem Krümmungssprung, z. B. einem tangentialen Kreis-Gerade-Übergang, wird der Toleranzwert aus dem Maschinendatum MD33120 $MA_PATH_TRANS_POS_TOL verwendet. Der Überschleifweg kann sich an diesen Stellen nämlich auch auf der Außenseite der programmierten Kontur befinden, wo viele Anwendungen weniger tolerant sind. Außerdem genügt in der Regel eine kleine, fest eingestellte Toleranz zum Ausgleich der Krümmungsänderungen, über die sich der NC-Programmierer keine Gedanken machen muss.

## 3.14.11 Satzwechselverhalten bei aktiver Kopplung (CPBC)

Mit dem Befehl CPBC wird das Satzwechselkriterium vorgegeben das erfüllt sein muss, damit im Teileprogramm bei aktiver Kopplung ein Satzwechsel durchgeführt wird.

### Syntax

```
CPBC[<Folgeachse>] = <Kriterium>
```

## Bedeutung

| | | |
|---|---|---|
| CPBC: | Satzwechselkriterium bei aktiver Kopplung | |
| <Folgeachse>: | Achsbezeichner der Folgeachse | |
| <Kriterium>: | Satzwechselkriterium | |
| | Typ: | STRING |
| | **Wert** | **Bedeutung:** Satzwechsel erfolgt |
| | "NOC" | unabhängig vom Kopplungszustand |
| | "IPOSTOP" | bei sollwertseitigem Synchronlauf |
| | "COARSE" | bei istwertseitigem Synchronlauf "Grob" |
| | "FINE" | bei istwertseitigem Synchronlauf "Fein" |

## Beispiel

**Programmcode**

```
; Satzwechsel erfolgt bei:
;  - Kopplung zu Folgeachse X2 == aktiv
;  - sollwertseitiger Synchronlauf == aktiv
CPBC[X2]="IPOSTOP"
```

# 3.15 Achsfunktionen

## 3.15.1 Achstausch, Spindeltausch (RELEASE, GET, GETD)

Eine oder mehrere Achsen bzw. Spindeln können immer nur in einem Kanal interpoliert werden. Muss eine Achse wechselweise in zwei verschiedenen Kanälen arbeiten (z. B. Palettenwechsler), so muss sie zunächst im aktuellen Kanal freigegeben und dann in den anderen Kanal übernommen werden. Die Achse wird zwischen den Kanälen getauscht.

**Achstauscherweiterungen**

Eine Achse/Spindel kann mit Vorlaufstopp und Synchronisation zwischen Vorlauf und Hauptlauf oder alternativ auch ohne Vorlaufstopp getauscht werden. Außerdem ist ein Achstausch auch möglich über:

- Achscontainer-Drehung AXCTSWE bzw. AXCTWED mittels impliziten `GET/GETD`.
- Frame mit Rotation, wenn diese Achse hierüber mit anderen Achsen verknüpft ist.
- Synchronaktionen, siehe Bewegungssynchronaktionen, "Achstausch `RELEASE`, `GET`".

**Maschinenhersteller**

Bitte beachten Sie die Angaben des Maschinenherstellers. Über projektierbare Maschinendaten muss eine Achse für den Achstausch in allen Kanälen eindeutig definiert sein und das Achstauschverhalten ist auch über Maschinedaten veränderbar einstellbar.

### Syntax

`RELEASE(Achsname, Achsname, ...)` oder `RELEASE(S1)`

`GET(Achsname, Achsname, ...)` oder `GET(S2)`

`GETD(Achsname, Achsname, ...)` oder `GETD(S3)`

Mit GETD (GET Directly) wird eine Achse direkt aus einem anderen Kanal geholt. Das bedeutet, dass zu diesem GETD kein passendes RELEASE in einem anderen Kanal programmiert sein muss. Es bedeutet aber auch, dass jetzt eine andere Kanalkommunikation aufgebaut werden muss (z. B. Waitmarken).

### Bedeutung

| | |
|---|---|
| `RELEASE (Achsname, Achsname, …):` | Freigeben der Achse(n) |
| `GET (Achsname, Achsname, …):` | Übernehmen der Achse(n) |
| `GETD (Achsname, Achsname, …):` | Direktes Übernehmen der Achse(n) |
| `Achsname:` | Achszuordnung im System: AX1, AX2, … oder Angabe der Maschinenachsnamen |
| `RELEASE(S1):` | Freigeben der Spindel S1, S2, … |
| `GET(S2):` | Übernehmen der Spindel S1, S2, … |
| `GETD(S3):` | Direktes Übernehmen der Spindel S1, S2, … |

**GET-Anforderung ohne Vorlaufstopp**

Wird nach einer GET-Anforderung **ohne** Vorlaufstopp die Achse mit `RELEASE(Achse)` oder `WAITP(Achse)` wieder freigegeben, so führt ein nachfolgender `GET` zu einem `GET` **mit** Vorlaufstopp.

**⚠ VORSICHT**

**Achszuordnung geändert**

Eine mit GET übernommene Achse bzw. Spindel bleibt auch nach einem Tasten- oder Programm-RESET diesem Kanal zugeordnet.

Bei neuem Programmstart muss die Zuordnung der getauschten Achsen bzw. Spindeln programmtechnisch erfolgen, falls die Achse in ihrem Grundkanal benötigt wird.

Bei POWER ON wird sie dem im Maschinendatum hinterlegten Kanal zugeordnet.

## Beispiele

### Beispiel 1: Achstausch zwischen zwei Kanälen

Von 6 Achsen werden in Kanal 1 zur Bearbeitung benutzt: 1., 2., 3. und 4. Achse.
5. und 6. Achse werden in Kanal 2 zum Werkstückwechsel benutzt.

Achse 2 soll zwischen beiden Kanälen getauscht werden können und nach POWER ON dem Kanal 1 zugeordnet sein.

Programm "MAIN" in Kanal 1:

| **Programmcode** | **Kommentar** |
|---|---|
| `INIT (2,"TAUSCH2")` | `; Programm TAUSCH2 im Kanal 2 anwählen.` |
| `N… START (2)` | `; Programm in Kanal 2 starten.` |
| `N… GET (AX2)` | `; Achse AX2 übernehmen.` |
| `...` | |
| `N… RELEASE (AX2)` | `; Achse AX2 freigeben.` |
| `N… WAITM (1,1,2)` | `; Warten auf WAIT-Marke in Kanal 1 und 2 zur Synchronisation in den beiden Kanälen.` |
| `...` | `; Weiterer Ablauf nach Achstausch.` |
| `N… M30` | |

Programm "TAUSCH2" in Kanal 2:

| **Programmierung** | **Kommentar** |
|---|---|
| `N… RELEASE(AX2)` | |
| `N160 WAITM(1,1,2)` | `; Warten auf WAIT-Marke in Kanal 1 und 2 zur Synchronisation in den beiden Kanälen.` |
| `N150 GET(AX2)` | `; Achse AX2 übernehmen.` |
| `...` | `; Weiterer Ablauf nach Achstausch.` |
| `N… M30` | |

### Beispiel 2: Achstausch ohne Synchronisierung

Wenn die Achse nicht synchronisiert werden muss, wird durch GET kein Vorlaufstopp erzeugt.

| Programmierung | Kommentar |
|---|---|
| N01 G0 X0 | |
| N02 RELEASE(AX5) | |
| N03 G64 X10 | |
| N04 X20 | |
| N05 GET(AX5) | ; Wenn keine Synchronisation nötig, wird dies kein ausführbarer Satz. |
| N06 G01 F5000 | ; Kein ausführbarer Satz. |
| N07 X20 | ; Kein ausführbarer Satz, da X-Position wie in N04. |
| N08 X30 | ; Erster ausführbarer Satz nach N05. |
| ... | |

### Beispiel 3: Aktivierung eines Achstausches ohne Vorlaufstopp

Voraussetzung: Der Achstausch ohne Vorlaufstopp muss über ein Maschinendatum projektiert werden.

| Programmierung | Kommentar |
|---|---|
| N010 M4 S100 | |
| N011 G4 F2 | |
| N020 M5 | |
| N021 SPOS=0 | |
| N022 POS[B]=1 | |
| N023 WAITP(B) | ; Achse B wird zur neutralen Achse. |
| N030 X1 F10 | |
| N031 X100 F500 | |
| N032 X200 | |
| N040 M3 S500 | ; Achse löst kein Vorlaufstopp/REORG aus. |
| N041 G4 F2 | |
| N050 M5 | |
| N099 M30 | |

Wird die Spindel bzw. Achse B unmittelbar nach dem Satz N023 als **PLC-Achse** z. B. auf 180 Grad und zurück auf 1 Grad verfahren, dann wird diese Achse wieder zur neutralen Achse und löst im Satz `N40` keinen Vorlaufstopp auf.

## Weitere Informationen

### Voraussetzungen für den Achstausch

- Die Achse muss über Maschinendaten in allen Kanälen definiert sein, die Achse verwenden wollen.
- Über das **achs**spezifische Maschinendatum muss festgelegt sein, welchem Kanal die Achse nach POWER ON zugeordnet werden soll.

### Beschreibung

#### Achse freigeben: RELEASE

Bei der Achsfreigabe ist zu beachten:

1. Die Achse darf an keiner Transformation beteiligt sein.
2. Bei Achskopplungen (Tangentialsteuerung), müssen alle Achsen des Verbands freigegeben werden.
3. Eine konkurrierende Positionierachse kann in diesem Zustand nicht getauscht werden.
4. Bei einer Gantry-Masterachse werden auch alle Folgeachsen getauscht.
5. Bei Achskopplungen (Mitschleppen, Leitwertkopplung, Elektronisches Getriebe) kann nur die Leitachse des Verbandes freigegeben werden.

#### Achse übernehmen: GET

Mit diesem Befehl wird der eigentliche Achstausch durchgeführt. Die Verantwortung für die Achse liegt vollständig bei dem Kanal, in dem der Befehl programmiert wurde.

#### Auswirkungen von GET:

Achstausch mit Synchronisierung:

Eine Achse muss immer dann synchronisiert werden, wenn sie zwischenzeitlich in einem anderen Kanal oder der PLC zugeordnet war, und vor dem GET keine Synchronisierung durch "WAITP", G74 oder Restweglöschen stattgefunden hat.

- Ein Vorlaufstopp erfolgt (wie bei STOPRE).
- Die Bearbeitung wird so lange unterbrochen, bis der Tausch vollständig ausgeführt ist.

#### Automatisches "GET"

Wenn eine Achse prinzipiell im Kanal verfügbar, jedoch derzeitig nicht als "Kanal-Achse" vorhanden ist, wird automatisch ein GET ausgeführt. Falls die Achse(n) schon synchronisiert ist (sind), wird kein Vorlaufstopp erzeugt.

#### Achstauschverhalten veränderbar einstellen

Der Abgabezeitpunkt von Achsen lässt sich über ein Maschinendatum wie folgt einstellen:

- Automatischer Achstausch findet zwischen zwei Kanälen auch dann statt, wenn die Achse durch WAITP in einen neutralen Zustand gebracht wurde (Verhalten wie bisher)
- Bei der Anforderung einer Achs-Containerdrehung werden alle dem ausführenden Kanal zuordenbaren Achsen des Achs-Containers mittels impliziten GET bzw. GETD in den Kanal geholt. Ein anschließender Achstausch ist erst nach dem Abschluss der Achs-Containerdrehung wieder erlaubt.

- Nach einem eingeschobenen Zwischensatz im Hauptlauf wird geprüft, ob ein Reorganisieren erforderlich ist oder nicht. Nur wenn die Achszustände dieses Satzes mit den aktuellen Achszuständen **nicht** übereinstimmen, ist ein Reorganisieren erforderlich.
- Statt eines GET-Satzes mit Vorlaufstopp und Synchronisation zwischen Vorlauf und Hauptlauf kann ein Achstausch auch ohne Vorlaufstopp erfolgen. Es wird dann nur ein Zwischensatz mit der GET-Anforderung erzeugt. Im Hauptlauf wird bei Abarbeitung dieses Satzes überprüft, ob die Zustände der Achse im Satz mit den aktuellen Achszuständen übereinstimmen.

## 3.15.2 Achse einem anderen Kanal übergeben (AXTOCHAN)

Mit dem Sprachbefehl `AXTOCHAN` kann eine Achse angefordert werden, um diese Achse einem anderen Kanal zu übergeben. Die Achse kann sowohl vom NC-Teileprogramm als auch aus einer Synchronaktion heraus in den entsprechenden Kanal gebracht werden.

### Syntax

```
AXTOCHAN(Achsname,Kanalnummer[,Achsname,Kanalnummer[,...]])
```

### Bedeutung

| Element | Beschreibung |
|---|---|
| `AXTOCHAN`: | Achse für einen bestimmten Kanal anfordern |
| `Achsname`: | Achszuordnung im System: X, Y, … oder Angabe der beteiligten Maschinenachsnamen. Der auszuführende Kanal muss nicht der eigene Kanal sein und es muss auch nicht der Kanal sein, der aktuell das Interpolationsrecht für die Achse besitzt |
| `Kanalnummer`: | Nummer des Kanals, dem die Achse zugeordnet werden soll |

**Hinweis**

**Konkurrierende Positionierachse und ausschließlich PLC kontrollierte Achse**

Eine PLC-Achse kann als konkurrierende Positionierachse den Kanal nicht wechseln. Eine ausschließlich von der PLC kontrollierte Achse kann nicht dem NC-Programm zugeordnet werden.

**Weitere Informationen:** Funktionshandbuch Achsen und Spindeln

### Beispiel

**AXTOCHAN im NC-Programm**

Die Achsen X und Y sind im 1. Kanal und im 2. Kanal bekannt. Aktuell hat der Kanal 1 das Interpolationsrecht und im Kanal 1 wird folgendes Programm gestartet:

| Programmcode | Kommentar |
|---|---|
| `N110 AXTOCHAN(Y,2)` | `; Y-Achse in den 2. Kanal schieben.` |

| Programmcode | Kommentar |
|---|---|
| N111 M0 | |
| N120 AXTOCHAN(Y,1) | ; Y-Achse wieder zurückholen (neutral). |
| N121 M0 | |
| N130 AXTOCHAN(Y,2,X,2) | ; Y-Achse und X-Achse in den 2. Kanal schieben (Achsen neutral). |
| N131 M0 | |
| N140 AXTOCHAN(Y,2) | ; Y-Achse in den 2. Kanal schieben (NC-Programm). |
| N141 M0 | |

## Weitere Informationen

### AXTOCHAN im NC-Programm

Dabei wird nur bei einer Anforderung der Achse für das NC-Programm im eigenen Kanal ein GET durchgeführt und damit auch auf die tatsächliche Zustandsänderung gewartet. Wird die Achse für einen anderen Kanal angefordert oder soll sie zur neutralen Achse im eigenen Kanal werden, dann nur wird die Anforderung entsprechend abgesetzt.

### AXTOCHAN aus einer Synchronaktion

Wird eine Achse für den eigenen Kanal angefordert so wird AXTOCHAN aus einer Synchronaktion auf ein GET aus einer Synchronaktion abgebildet. In diesem Fall wird die Achse bei der ersten Anforderung für den eigenen Kanal zur neutralen Achse. Bei der zweiten Anforderung wird die Achse dem NC-Programm analog zur GET-Anforderung im NC-Programm zugeordnet. Zur GET-Anforderung aus einer Synchronaktion siehe Kapitel "Bewegungssynchronaktionen".

## 3.15.3 Achsfunktionen (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL)

"AXNAME" wird z. B. bei der Erstellung allgemeingültiger Zyklen verwendet, wenn die Namen der Achsen nicht bekannt sind.

"AX" wird für die indirekte Programmierung von Geometrie- und Synchronachsen verwendet. Der Achsbezeichner wird dabei in einer Variablen vom Typ AXIS hinterlegt oder von einem Befehl wie "AXNAME" oder "SPI" geliefert.

"SPI" wird verwendet, wenn Achsfunktionen für eine Spindel, z. B. Synchronspindel, programmiert werden.

"AXTOSPI" wird verwendet, um einen Achsbezeichner in einen Spindelindex zu wandeln (Umkehrfunktion zu "SPI").

"AXSTRING" wird verwendet, um einen Achsbezeichner (Datentyp AXIS) in einen String zu wandeln (Umkehrfunktion zu "AXNAME").

"ISAXIS" wird in allgemeingültigen Zyklen verwendet, um sicherzustellen, dass eine bestimmte Geometrieachse vorhanden ist und damit ein nachfolgender Aufruf von $P_AXNX nicht mit Fehler abgebrochen wird.

"MODAXVAL" wird verwendet, um bei Modulo-Rundachsen die Modulo-Position zu ermitteln.

## Syntax

```
AXNAME("String")
AX[AXNAME("String")]
SPI(n)
```

`AXTOSPI(A)` oder `AXTOSPI(B)` oder `AXTOSPI(C)`

```
AXSTRING(SPI(n))
ISAXIS(<Geometrieachsnummer>)
<Modulo-Position>=MODAXVAL(<Achse>,<Achsposition>)
```

## Bedeutung

| | |
|---|---|
| `AXNAME`: | Konvertiert einen Eingangsstring in Achsbezeichner; der Eingangsstring muss gültigen Achsnamen enthalten. |
| `AX`: | Variabler Achsbezeichner |
| `SPI`: | Konvertiert Spindelnummer in Achsbezeichner; der Übergabeparameter muss eine gültige Spindelnummer enthalten. |
| `n`: | Spindelnummer |
| `AXTOSPI`: | Wandelt einen Achsbezeichner in einen Spindelindex vom Typ Integer um. "AXTOSPI" entspricht der Umkehrfunktion zu "SPI". |
| `X, Y, Z`: | Achsbezeichner vom Typ AXIS als Variable oder Konstante |
| `AXSTRING`: | Es wird der String mit zugeordneter Spindelnummer ausgegeben. |
| `ISAXIS`: | Prüft, ob die angegebene Geometrieachse vorhanden ist. |
| `MODAXVAL`: | Ermittelt bei Modulo-Rundachsen die Modulo-Position; diese entspricht dem Modulo-Rest bezogen auf den parametrierten Modulo-Bereich (beträgt in der Standardeinstellung 0 bis 360 Grad; über MD30340 MODULO_RANGE_START und MD30330 $MA_MODULO_RANGE können Beginn und Größe des Modulo-Bereichs verändert werden). |

### Hinweis

**SPI-Erweiterungen**

Die Achsfunktion SPI(n) ist auch für das Lesen und Schreiben von Framekomponenten einsetzbar. Damit können Frames z. B. mit der Syntax `$P_PFRAME[SPI(1),TR]=2.22` geschrieben werden.

Durch die zusätzliche Programmierung von Achspositionen über die Adresse `AX[SPI(1)]=<Achsposition>` kann eine Achse verfahren werden. Voraussetzung dafür ist, dass sich die Spindel im Positionier- oder Achsbetrieb befindet.

## Beispiele

### Beispiel 1: AXNAME, AX, ISAXIS

| **Programmcode** | **Kommentar** |
|---|---|
| `OVRA[AXNAME("Planachse")]=10` | `; Override für Planachse` |

| Programmcode | Kommentar |
|---|---|
| AX[AXNAME("Planachse")]=50.2 | ; Endposition für Planachse |
| OVRA[SPI(1)]=70 | ; Override für Spindel 1 |
| AX[SPI(1)]=180 | ; Endposition für Spindel 1 |
| IF ISAXIS(1)==FALSE GOTOF WEITER | ; Abszisse vorhanden? |
| AX[$P_AXN1]=100 | ; Abszisse verfahren |
| WEITER: | |

### Beispiel 2: AXSTRING

Bei der Programmierung mit AXSTRING[SPI(n)] wird nicht mehr der Achsindex der Achse, der die Spindel zugeordnet ist, als Spindelnummer ausgegeben, sondern es wird der String "Sn" ausgegeben.

| Programmcode | Kommentar |
|---|---|
| AXSTRING[SPI(2)] | ; Es wird der String "S2" ausgegeben. |

### Beispiel 3: MODAXVAL

Die Modulo-Position der Modulo-Rundachse A soll ermittelt werden.

Ausgangswert für die Berechnung ist die Achsposition 372.55.

Der parametrierte Modulo-Bereich beträgt 0 bis 360 Grad:

MD30340 MODULO_RANGE_START = 0

MD30330 $MA_MODULO_RANGE = 360

| Programmcode | Kommentar |
|---|---|
| R10=MODAXVAL(A,372.55) | ; Berechnete Modulo-Position R10 = 12.55. |

### Beispiel 4: MODAXVAL

Wenn sich der programmierte Achsbezeichner nicht auf eine Modulo-Rundachse bezieht, dann wird der zu wandelnde Wert (<Achsposition>) unverändert zurückgegeben.

| Programmcode | Kommentar |
|---|---|
| R11=MODAXVAL(X,372.55) | ; X ist Linearachse; R11 = 372.55. |

## 3.15.4 Umschaltbare Geometrieachsen (GEOAX)

Mit der Funktion "Umschaltbare Geometrieachsen" können die über Maschinendaten parametrierten Geometrieachsen durch andere Kanalachsen ersetzt werden.

### Syntax

```
GEOAX(<n>,<Kanalachse>,<n>,<Kanalachse>,<n>,<Kanalachse>)
GEOAX()
```

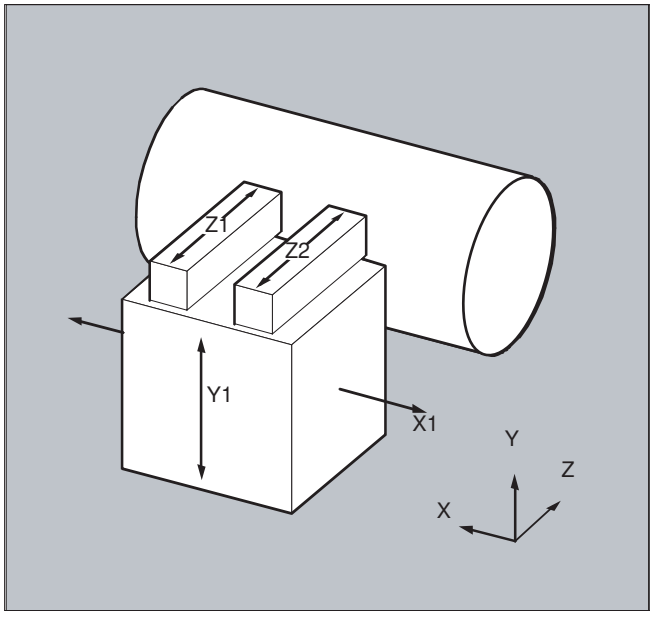
## Bedeutung

| | |
|---|---|
| `GEOAX(...)` | Funktion zum Umschalten von Geometrieachsen.<br>**Hinweis:**<br>`GEOAX()` ohne Parameterangabe aktiviert wieder die in den Maschinendaten parametrierte Grundkonfiguration der Geometrieachsen. |
| `<n>` | Nummer der Geometrieachse, die durch die angegebene Kanalachse ersetzt wird.<br>Wertebereich: 0, 1, 2, 3<br>**Hinweis:**<br>0: Die angegebene Kanalachse wird ersatzlos aus dem Geometrieachsverbund entfernt<br>1: 1. Geometrieachse ≙ Koordinatenachse X (Abszisse) des WKS<br>2: 2. Geometrieachse ≙ Koordinatenachse Y (Ordinate) des WKS<br>3: 3. Geometrieachse ≙ Koordinatenachse Z (Applikate) des WKS |
| `<Kanalachse>` | Name der Kanalachse, die in den Geometrieachsverbund aufgenommen werden soll. |

## Beispiele

### Beispiel 1: Zwei Achsen wechselweise als Geometrieachse schalten

Ein Werkzeugschlitten kann über die Kanalachsen X1, Y1, Z1, Z2 verfahren werden:




Die Geometrieachsen sind so projektiert, dass nach dem Einschalten zunächst Z1 als 3. Geometrieachse unter dem Geometrieachsnamen "Z" wirksam ist und zusammen mit X1 und Y1 den Geometrieachsverbund bildet.

Im Teileprogramm sollen nun die Achsen Z1 und Z2 wechselweise als Geometrieachse Z zum Einsatz kommen:

```
Programmcode                    Kommentar
...
N100 GEOAX(3,Z2)                ; Als 3. Geometrieachse (Z) fungiert Kanalachse Z2.
N110 G1 ...
N120 GEOAX(3,Z1)                ; Als 3. Geometrieachse (Z) fungiert Kanalachse Z1.
...
```

### Beispiel 2: Umschalten der Geometrieachsen bei 6 Kanalachsen

Eine Maschine besitzt 6 Kanalachsen mit den Namen XX, YY, ZZ, U, V, W.

Die Grundeinstellung der Geometrieachskonfiguration über Maschinendaten ist:

Kanalachse XX = 1. Geometrieachse (X-Achse)

Kanalachse YY = 2. Geometrieachse (Y-Achse)

Kanalachse ZZ = 3. Geometrieachse (Z-Achse)

```
Programmcode                    Kommentar
N10 GEOAX()                     ; Grundkonfiguration der Geometrieachsen ist wirksam.
N20 G0 X0 Y0 Z0 U0 V0 W0        ; Alle Achsen im Eilgang auf Position 0.
N30 GEOAX(1,U,2,V,3,W)          ; Kanalachse U wird zur ersten (X), V zur zweiten (Y)
                                ; und W zur dritten Geometrieachse (Z).
N40 GEOAX(1,XX,3,ZZ)            ; Kanalachse XX wird zur ersten (X), ZZ zur dritten
                                ; Geometrieachse (Z). Kanalachse V bleibt zweite
                                ; Geometrieachse (Y).
N50 G17 G2 X20 I10 F1000        ; Vollkreis in der X/Y-Ebene. Es fahren die
                                ; Kanalachsen XX und V.
N60 GEOAX(2,W)                  ; Kanalachse W wird zweite Geometrieachse (Y).
N80 G17 G2 X20 I10 F1000        ; Vollkreis in der X/Y-Ebene. Es fahren die
                                ; Kanalachsen XX und W.
N90 GEOAX()                     ; Zurücksetzen auf Grundzustand.
N100 GEOAX(1,U,2,V,3,W)         ; Kanalachse U wird zur ersten (X), V zur zweiten
                                ; (Y) und W zur dritten Geometrieachse (Z).
N110 G1 X10 Y10 Z10 XX=25       ; Kanalachsen U, V, W fahren jeweils auf die
                                ; Position 10. XX als Zusatzachse fährt auf Position
                                25.
N120 GEOAX(0,V)                 ; V wird aus Geometrieachsverbund herausgenommen.
                                ; U und W sind weiterhin erste (X) und dritte
                                ; Geometrieachse (Z).
                                ; Die zweite Geometrieachse (Y) bleibt unbelegt.
N130 GEOAX(1,U,2,V,3,W)         ; Kanalachse U bleibt erste (X), V wird zur
                                ; zweiten (Y), W bleibt dritte Geometrieachse (Z).
```

| Programmcode | Kommentar |
|---|---|
| N140 GEOAX(3,V) | ; V wird zur dritten Geometrieachse (Z), wobei W<br>; überschrieben und damit aus dem Geometrieachsverbund<br>; herausgenommen wird. Die zweite Geometrieachse (Y)<br>; ist nach wie vor unbelegt. |

## Maschinendaten

### Achskonfiguration

Zuordnung von Geometrie-, Zusatz- und Maschinenachsen zu Kanalachsen:

- MD10000 $MN_AXCONF_MACHAX_NAME_TAB
- MD20050 $MC_AXCONF_GEOAX_ASIGN_TAB
- MD20060 $MC_AXCONF_GEOAX_NAME_TAB
- MD20070 $MC_AXCONF_MACHAX_USED
- MD20080 $MC_AXCONF_CHANAX_NAME_TAB
- MD35000 $MA_SPIND_ASSIGN_TO_MACHAX

### Reset-Verhalten

Reset-Verhalten von geänderten Geometrieachszuordnungen:

- MD20110 $MC_RESET_MODE_MASK, Bit 12
- MD20118 $MC_GEOAX_CHANGE_RESET

### NC-Start-Verhalten

- MD20112 $MC_START_MODE_MASK, Bit 12

### Mitteilung an PLC-Anwenderprogramm

Parametriermöglichkeit des M-Befehls der bei einer Geometrieachsumschaltung an die NC/PLC-Nahtstelle ausgegeben wird:

- MD22532 $MC_GEOAX_CHANGE_M_CODE

## Randbedingungen

### Keine Geometrieachsumschaltung

- Ist eine der folgenden Funktionen aktiv, ist keine Geometrieachsumschaltung möglich:
  - Transformation
  - Spline-Interpolation
  - Werkzeugradiuskorrektur
  - Werkzeugfeinkorrektur
- Die Geometrieachse und eine weitere Kanalachse haben den gleiche Namen.
- Eine der an der Geometrieachsumschaltung beteiligten Achsen ist an einer Aktion beteiligt, die über Satzgrenzen hinweg andauert. Z. B. Satzübergreifende Positionierachse oder Folgeachse einer Achskopplung.

### Rundachsen

Rundachsen können nicht zu Geometrieachsen gemacht werden.

### Achszustand nach dem Ersetzen

Eine durch die Umschaltung im Geometrieachsverbund ersetzte Achse ist nach dem Umschaltvorgang über ihren Kanalachsnamen als Zusatzachse programmierbar.

### Frames, Schutzbereiche, Arbeitsfeldbegrenzungen

Mit dem Umschalten der Geometrieachsen werden alle Frames, Schutzbereiche und Arbeitsfeldbegrenzungen gelöscht.

### Polarkoordinaten

Ein Tausch der Geometrieachsen mit `GEOAX` setzt analog einem Ebenenwechsel mit `G17-G19` die modalen Polarkoordinaten auf den Wert 0.

### DRF, NPV

Eine eventuelle Handrad-Verschiebung (DRF) oder eine externe Nullpunktverschiebung (NPV) bleibt nach der Umschaltung wirksam.

### Grundkonfiguration der Geometrieachsen

Der Befehl `GEOAX()` ruft die Grundkonfiguration des Geometrieachsverbunds auf.

Nach POWER ON und bei Umschalten in die Betriebsart "Referenzpunktfahren" wird automatisch auf die Grundkonfiguration zurückgeschaltet.

### Werkzeuglängenkorrektur

Eine aktive Werkzeuglängenkorrektur ist auch nach dem Umschaltvorgang wirksam. Sie gilt jedoch für die neu aufgenommenen bzw. positionsgetauschten Geometrieachsen als noch nicht herausgefahren. Beim ersten Bewegungsbefehl für diese Geometrieachsen besteht der resultierende Verfahrweg dementsprechend aus der Summe von Werkzeuglängenkorrektur und programmiertem Verfahrweg.

Geometrieachsen, die bei einer Umschaltung ihre Position im Achsverband beibehalten, behalten auch ihren Status bezüglich der Werkzeuglängenkorrektur.

**Geometrieachskonfiguration bei aktiver Transformation**

- Die für eine aktive Transformation über Transformations-Maschinendaten parametrierte Geometrieachskonfiguration ist über die Funktion "Umschaltbare Geometrieachsen" nicht änderbar.
- Unterschiedliche Geometrieachskonfiguration für eine Transformation, müssen in den Transformations-Maschinendaten unterschiedliche Datensätze parametriert werden.
- Eine über `GEOAX` geändert Geometrieachskonfiguration wird durch Aktivierung einer Transformation gelöscht.
- Bezüglich der Geometrieachsen haben die Transformations-spezifischen Geometrieachsparametrierungen von aktiven Transformationen Vorrang vor den die Umschaltung von Geometrieachsen betreffenden Parametrierungen.
  Beispiel: Eine Transformation ist aktiv. Laut Maschinendaten soll die Transformation bei Kanal-Reset erhalten bleiben. Gleichzeitig soll bei einem Kanal-Reset jedoch die Grundkonfiguration der Geometrieachsen hergestellt werden. Es bleibt die Geometrieachskonfiguration erhalten, die für die Transformation festgelegt ist.
- Mit dem Ausschalten einer Transformation wird wieder die parametrierte Grundeinstellung der Geometrieachskonfiguration wirksam.

**Betriebsart JOG, Maschinenfunktion REF**

Beim Umschalten in die Betriebsart JOG, Maschinenfunktion REF (Referenzpunkt fahren), wird die in den Maschinendaten parametrierte Geometrieachskonfiguration wirksam..

## 3.15.5 Achscontainer (AXCTSWE, AXCTSWED, AXCTSWEC)

Über die Befehle "AXCTSWE" bzw. "AXCTSWED" wird die Drehung des angegebenen Achscontainers freigegeben.

Über den Befehl "AXCTSWEC" wird eine bereits gegebene Freigabe zur Achscontainer-Drehung wieder zurückgenommen.

### Syntax

```
AXCTSWE(<ID>)
AXCTSWED(<ID>)
AXCTSWEC(<ID>)
```

## Bedeutung

| | | |
|---|---|---|
| AXCTSWE: | Freigabe zum Drehen des Achscontainers<br>Die Programmverarbeitung wird durch "AXCTSWE" nicht angehalten.<br>Die Drehung erfolgt, sobald alle am Achscontainer beteiligten Kanäle die Freigabe erteilt haben. | |
| AXCTSWED: | Freigabe zum Drehen des Achscontainers ohne Berücksichtiung der anderen am Achscontainer beteiligten Kanäle<br>**Hinweis**<br>• Befehlsvariante zur Vereinfachung der Inbetriebnahme des Teileprogramms bzw. Synchronaktion.<br>• Das Verhalten bezüglich der anderen am Achscontainer beteiligten Kanäle, kann vorgegeben werden über:<br>MD12760 $MN_ AXCT_FUNCTION_MASK, Bit 0 | |
| AXCTSWEC: | Rücknahme der Freigabe zum Drehen des Achscontainers<br>**Hinweis**<br>Die Freigabe zum Drehen des Achscontainers kann nur zurück genommen werden, wenn die Drehung noch nicht begonnen wurde:<br>$AN_AXCTSWA[<Achscontainer>] == 0<br>Zu Systemvariable siehe "Achscontainer (AXCTSWE, AXCTSWED, AXCTSWEC) (Seite 898)" | |
| <ID>: | Bezeichner des Achscontainers oder einer Containerachse: | |
| | <Nummer>: | Defaultbezeichner eines Achscontainers:<br>MD12750 $MN_AXCT_NAME_TAB<br>Beispiel: "CT1" |
| | <Container>: | Anwenderspezifischer Bezeichner eines Achscontainers:<br>MD12750 $MN_AXCT_NAME_TAB<br>Beispiel: "CONTAINER_1" |
| | <Achse>: | Bezeichner einer im Kanal bekannten Containerachse |

**Hinweis**

**Schrittweite**

Die Schrittweite einer Achscontainer-Drehung wird eingestellt über das Settingdatum:

SD41700 $SN_AXCT_SWWIDTH

## Weitere Informationen

### Diagnose

Der aktuelle Status eines Achscontainers kann über folgende Systemvariable gelesen werden:

| Systemvariable | Typ | Beschreibung |
|---|---|---|
| $AC_AXCTSWA[<Name>] | BOOL | Kanal-spezifischer Status des Achscontainers |
| $AN_AXCTSWA[<Achscontainer>] | BOOL | NCU-spezifischer Status des Achscontainer |

| Systemvariable | Typ | Beschreibung |
|---|---|---|
| $AN_AXCTSWE[<Achscontainer>] | INT | Slot-spezifischer Status der Achscontainer-Drehung<br>Die Systemvariable liefert **Bit-weise** den Status der Slots des Achscontainers. Jedes Bit entspricht einem Slot. |
| $AN_AXCTAS[<Achscontainer>] | INT | Anzahl der Plätze (Slots) um die der Achscontainer aktuell weitergeschaltet wurde. |

#### Achscontainer-Drehung mit implizitem GET / GETD

Über nachfolgendes Maschinendatum kann eingestellt werden, dass mit dem Befehl "AXCTSWE" alle Container-Achsen des Kanals mittels impliziten "GET / GETD" in den Kanal geholt werden. Ein Achstausch ist erst nach erfolgter Container-Drehung wieder möglich.

MD10722 $MN_AXCHANGE_MASK, Bit 1 = 1

**Hinweis**

Die Achscontainer-Drehung mit impliziten "GET / GETD" wird **nicht** für eine Achse im Zustand "Hauptlaufachse" (z. B. PLC-Achse) durchgeführt, da die Achse für die Achscontainer-Drehung den Zustand verlassen müsste.

## 3.15.6 Warten auf gültige Achsposition (WAITENC)

Mit dem Sprachbefehl "WAITENC" kann im NC-Programm gewartet werden, bis für die mit MD34800 $MA_WAIT_ENC_VALID = 1 projektierten Achsen synchronisierte bzw. restaurierte Achspositionen zur Verfügung stehen.

Im Wartezustand kann eine Unterbrechung erfolgen, z. B. durch Start eines ASUPs oder durch Betriebsartenwechsel nach JOG. Mit der Programmfortsetzung wird der Wartezustand ggf. wieder eingenommen.

**Hinweis**

Der Wartezustand wird in der Bedienoberfläche durch den Haltezustand "Warten auf Mess-System" angezeigt.

### Syntax

"WAITENC" kann im Programmteil eines beliebigen NC-Programms programmiert werden.

Die Programmierung muss in einem eigenen Satz erfolgen:

```
...
WAITENC
...
```

## Beispiel

"WAITENC" wird z. B. im ereignisgesteuerten Anwenderprogramm .../_N_CMA_DIR/_N_PROG_EVENT_SPF verwendet, wie das folgende Anwendungsbeispiel zeigt.

**Anwendungsbeispiel: Werkzeugrückzug nach POWER OFF mit Orientierungstransformation**

Eine Bearbeitung mit Werkzeugorientierung wurde durch Spannungsausfall abgebrochen. Beim anschließenden Hochlauf wird das ereignisgesteuerte Anwenderprogramm .../_N_CMA_DIR/_N_PROG_EVENT_SPF aufgerufen.

Im ereignisgesteuerten Anwenderprogramm wird mit "WAITENC" auf synchronisierte bzw. restaurierte Achspositionen gewartet, um danach einen Frame berechnen zu können, der das WKS in Werkzeugrichtung ausrichtet.

```
Programmcode                    Kommentar
...
IF $P_PROG_EVENT == 4           ; Hochlauf.
    IF $P_TRAFO <> 0            ; Transformation wurde angewählt.
        WAITENC                 ; Warten auf gültige Achspositionen der Orientie-
                                  rungsachsen.
        TOROTZ                  ; Z-Achse des WKS in Richtung der Werkzeugachse dre-
                                  hen.
    ENDIF
    M17
ENDIF
...
```

Danach kann das Werkzeug in der Betriebsart JOG durch eine Rückzugsbewegung in Richtung der Werkzeugachse freigefahren werden.

## 3.15.7 Programmierbare Parametersatzumschaltung (SCPARA)

Mit dem Befehl SCPARA kann für eine Achse die Umschaltung auf einen bestimmten Parametersatz angefordert werden.

**Hinweis**

**Keine Parametersatzumschaltung während Gewindebearbeitung**

Bei Gewindeschneiden G33 und Gewindebohren G331 / G332 wird der Parametersatz von der Steuerung ausgewählt und kann nicht verändert werden.

**Gesperrte Parametersatzumschaltung**

Eine Parametersatzumschaltung kann auch über die NC/PLC-Nahstelle angefordert werden. Um Umschaltkonflikte zu vermeiden, kann über die NC/PLC-Nahstelle die Parametersatzumschaltung der NC (SCPARA) gesperrt werden:

DB31, ... DBX9.3 (Parametersatzvorgabe durch NC gesperrt)

**Hinweis**

Wird eine Parametersatzumschaltung durch SCPARA angefordert, während die Parametersatzumschaltung über die NC/PLC-Nahstelle gesperrt ist, wird die Umschaltung ohne Fehlermeldung abgewiesen.

## Syntax

```
SCPARA[<Achse>]=<Wert>
```

## Bedeutung

| | | |
|---|---|---|
| SCPARA: | Befehl: Parametersatz umschalten | |
| <Achse>: | Achsbezeichner (Kanalachse) | |
| | Typ: | AXIS |
| <Wert>: | Parametersatznummer: 1, 2, 3, ... max. Parametersatznummer | |

## Beispiel

| Programmcode | Kommentar |
|---|---|
| ... | |
| N110 SCPARA[X] = 3 | ; Anwahl: Achse X, 3. Parametersatz |
| ... | |

## Weitere Informationen

### Freigabe der Parametersatzumschaltung

Die Parametersatzumschaltung der Achse muss explizit freigegeben werden:

MD35590 $MA_PARAMSET_CHANGE_ENABLE[<Achse>]

### Paramtersatznummer lesen

Die Nummer des angewählten Parametersatzes (Soll-Parametersatz) kann über die Systemvariable $AA_SCPAR gelesen werden.

## 3.15.8 Adaption ein-/ausschalten (CADAPTON, CADAPTOF)

Mit den vordefinierten Prozeduren CADAPTON() und CADAPTOF() können vordefinierte Adaptionen zur Anpassung von Dynamik- oder Regelungsparametern aus dem Teileprogramm heraus aktiviert, aktualisiert und deaktiviert werden. Verwendung finden sie auch im CYCLE782 (Seite 1109).

## Syntax

```
...
CADAPTON(<Result>,<Axis>,<InVar>[,<InValue>])
...
CADAPTOF(<Result>,<Axis>,<InVar>)
...
```

## Bedeutung

| | | | |
|---|---|---|---|
| `CADAPTON()`: | Adaptionsbeziehung aktivieren | | |
| `CADAPTOF()`: | Adaptionsbeziehung deaktivieren<br>**Hinweis:**<br>Permanent wirksame Adaptionsbeziehungen (MD16501 = 1) ignorieren `CADAPTOF()`, ein ggf. zuvor programmierter Eingangswert (`<InVal>`) bleibt wirksam. | | |
| `<Result>`: | Ergebnisvariable: Rückgabewert für den Status (Call by reference-Parameter) | | |
| | Datentyp: | INT | |
| | Wert: | 0 | kein Fehler |
| | | 1 | keine gültige Adaptionstabelle parametriert |
| | | 2 | Parameter `<Axis>` ungültig |
| | | 3 | Parameter `<InVar>` ungültig |
| | | 4 | reserviert |
| | | 5 | Parameter `<InVal>` ungültig |
| `<Axis>`: | Maschinenachsname der Eingangsachse der Adaptionsbeziehung | | |
| | Datentyp: | AXIS | |
| | Wertebereich: | Im Kanal definierte Maschinenachsnamen | |
| | **Hinweis:**<br>Mit diesem Parameter werden diejenigen Adaptionen adressiert, die in MD16504 $MN_CADAPT_INPUT_AX einen dem Parameter `<Axis>` entsprechenden Wert eingetragen haben. Die weiteren Parameter `<InVar>` und `<InVal>` sind dieser Achse zugeordnet. | | |
| `<InVar>`: | Eingangsgröße der Adaptionsbeziehung | | |
| | Datentyp: | INT | |
| | Wert: | 1 | Trägheit der Achse |
| | | 2 | Achsposition |
| | | 3 | Achsgeschwindigkeit |

| | | | |
|---|---|---|---|
| <InVal>: | Eingangswert der Adaptionsbeziehung<br>Optionaler Parameter für die intelligente Lastanpassung (<InVar>=1). | | |
| | Datentyp: | REAL | |
| | Wert: | > 0 | Aktuelle Trägheit |
| | | = 0 | Last/Beladung ist undefiniert bzw. nicht bekannt. Die zugeordneten Adaptionen liefern in diesem Fall als Ausgangsgröße den Ersatzwert 1.0. |
| | | < 0 | Der Aufruf wird mit <Status> = 5 (Parameter <InVal> ungültig) beendet. Der zuletzt aktivierte Eingangswert bleibt weiterhin aktiv. |
| | **Hinweis:**<br>Ohne Programmierung eines Eingangswerts wird der zuletzt programmierte Wert bzw. nach einem Steuerungshochlauf der Standardwert (= 0) wirksam. | | |

## Beispiel

```
Programmcode                                  Kommentar
DEF INT RESULT                                ; Definition der Ergebnisvariablen
...
CADAPTON(RESULT,MX1,2)                        ; Adaption aktivieren
IF RESULT <> 0                                ; Auswertung der Ergebnisvariablen (nach
                                              jeder CADAPTON/CADAPTOF-Anweisung erfor-
                                              derlich)
   MSG("CADAPT-RESULT=" << RESULT)
   STOPRE
   SETAL(61000)
ENDIF
...
CADAPTOF(RESULT,MX1,2)                        ; Adaption deaktivieren
...                                           ; Auswertung der Ergebnisvariablen
...
```

## Weitere Informationen

### Link-Achsen

Die Funktion kann nicht auf Link-Achsen (Funktion NCU-Link) angewendet werden. Die Ergebnisvariable würde in diesem Fall den Rückgabewert 2 ("Parameter <Axis> ungültig") liefern.

**Satzsuchlauf**

- **Satzsuchlauf ohne Berechnung**
  CADAPTON/CADAPTOF-Anweisungen, die zwischen Programmanfang und Zielsatz programmiert sind, werden ignoriert. Der Anwender muss das Suchziel so wählen, dass die für den Bearbeitungsabschnitt relevanten Adaptionen angewählt werden.
- **Satzsuchlauf mit Berechnung an Kontur bzw. an Satzendpunkt**
  Die Aktivierungs- bzw. Deaktivierungsbefehle sowie evtl. programmierte Eingangswerte werden aufgesammelt und der aktuelle Zustand am Ende des Suchlaufs mit den Suchlauf-Aktionssätzen ausgegeben.
- **Satzsuchlauf mit Berechnung im Modus Programmtest (SERUPRO)**
  Mit dem Ausschalten von SERUPRO wird der durch die CADAPTON/CADAPTOF-Anweisungen erreichte Zustand der Adaptionen in den realen Betrieb überführt.

## 3.15.9 FIR-Ruckfilter an Dynamikmodus anpassen (CALCFIR)

Um nach einem Wechsel des Dynamikmodus für alle Achsen im Kanal, für die ein FIR-Tiefpass-Ruckfilter aktiv ist, eine identische Dämpfungswirkung und Konturgenauigkeit zu erreichen, können mit der vordefinierten Prozedur CALCFIR die dynamikabhängigen FIR-Filtereinstellungen der begrenzenden Linearachse des Interpolationsverbundes auf alle anderen Kanalachsen mit aktivem FIR-Tiefpass-Ruckfilter übertragen und wirksam gesetzt werden.

**Anwendung**

Eingesetzt wird CALCFIR im Herstellerzyklus CUST_832.

**Standardablauf**

Am Anfang eines NC-Programms zur Freiformflächenbearbeitung wird CYCLE832 programmiert. Bei der Ausführung des Programms ruft CYCLE832 den Herstellerzyklus CUST_832 auf und die vom Maschinenhersteller im CUST_832 eingefügte vordefinierte Prozedur CALCFIR wird ausgeführt.

**Wirksamkeit**

Die durch CALCFIR überschriebenen Filtereinstellungen bleiben solange wirksam, bis sie durch den nächsten CALCFIR-Aufruf erneut überschrieben werden. Dies ist immer dann erforderlich, wenn durch einen Befehl der G-Gruppe 59 (Dynamikmodus für Bahninterpolation) eine Änderung des aktuellen Dynamikmodus programmiert ist.

**Verhalten bei Steuerungshochlauf / Kanal-Reset / Programmende-Reset**

Im Steuerungshochlauf und bei Kanal-/Programmende-Reset werden automatisch FIR-Filtereinstellungen passend zur Löschstellung der G-Gruppe 59 aktiviert.

## Voraussetzungen

- Option "Top Speed" ist gesetzt.

  **Hinweis**

  Um mit "Top Speed" optimale Ergebnisse zu erzielen, wird zusätzlich die Verwendung der Option "Top Surface" empfohlen.

- Für die Achsen im Interpolationsverbund müssen folgende Bedingungen erfüllt sein:
  - FIR-Tiefpass-Ruckfilter ist aktiv.
  - Dynamikabhängige FIR-Filtereinstellungen sind projektiert.
  - Das Überschreiben der Filtereinstellungen durch CALCFIR ist freigegeben.
- Ruckbegrenzung SOFT/SOFTA ist aktiv.

## Syntax

CALCFIR wird in einem eigenen Satz programmiert. Der Aufruf im Herstellerzyklus CUST_832 erfolgt direkt nach der Programmierung des Dynamikmodus:

```
...
DYN...
CALCFIR
...
```

## Bedeutung

| | | |
|---|---|---|
| DYN...: | Befehl aus der G-Gruppe 59 zur Auswahl des Dynamikmodus | |
| | DYNNORM: | **Normale Dynamik** aktivieren |
| | DYNPOS: | Dynamik für **Positionierbetrieb, Gewindebohren** aktivieren |
| | DYNROUGH: | Dynamik für **Schruppen** aktivieren |
| | DYNSEMIFIN: | Dynamik für **Vorschlichten** aktivieren |
| | DYNFINISH: | Dynamik für **Schlichten** aktivieren |
| | DYNPREC: | Dynamik für **Feinschlichten** aktivieren |
| CALCFIR: | Vordefinierte Prozedur zur Dynamikanpassung von FIR-Tiefpass-Ruckfiltern | |

**Hinweis**

CALCFIR führt ein implizites NEWCONFIG aus, um das Ergebnis der FIR-Filter-Berechnung sowohl in den achsspezifischen Maschinendaten zur Parametrierung der FIR-Tiefpass-Ruckfilter als auch für CPRECON (Seite 860) wirksam zu machen. Das implizite NEWCONFIG führt dazu, dass weitere NEWCONFIG-wirksame Maschinendaten wirksam werden können.

## Randbedingungen

### Verwendung in Synchronaktionen

CALCFIR ist **nicht** in Synchronaktionen programmierbar.

# 3.16 Achskopplungen

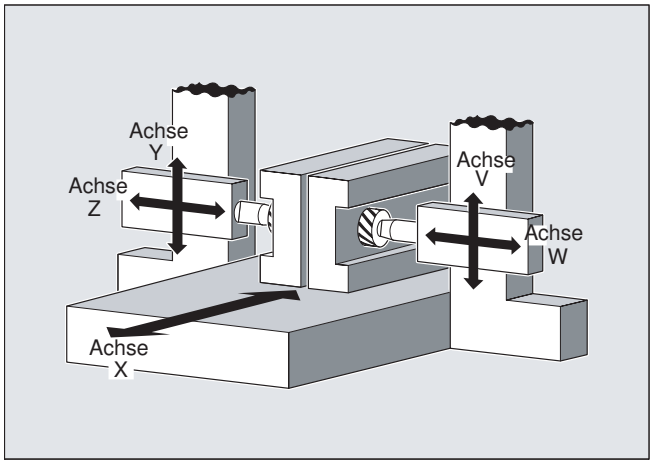
## 3.16.1 Mitschleppen (TRAILON, TRAILOF)

Beim Bewegen einer definierten Leitachse fahren ihr zugeordnete Mitschleppachsen (= Folgeachsen) unter Berücksichtigung eines Koppelfaktors die von der Leitachse abgeleiteten Verfahrwege ab.

Leitachse und Folgeachsen bilden zusammen einen Mitschleppverband.

**Anwendungsbereiche**

- Verfahren einer Achse durch eine simulierte Achse. Die Leitachse ist eine simulierte Achse und die Mitschleppachse eine reale Achse. Damit kann die reale Achse mit Berücksichtigung des Koppelfaktors verfahren werden.
- Zweiseitenbearbeitung mit 2 Mitschleppverbänden:
  1. Leitachse Y, Mitschleppachse V
  2. Leitachse Z, Mitschleppachse W




### Syntax

```
TRAILON(<Folgeachse>,<Leitachse>,<Koppelfaktor>)
TRAILOF(<Folgeachse>,<Leitachse>,<Leitachse 2>)
TRAILOF(<Folgeachse>)
```

### Bedeutung

| | | |
|---|---|---|
| TRAILON: | Befehl zum Einschalten und Definieren eines Mitschleppverbandes | |
| | Wirksamkeit: | modal |
| <Folgeachse>: | Parameter 1: Achsbezeichnung der Mitschleppachse<br>**Hinweis:**<br>Eine Mitschleppachse kann auch Leitachse für weitere Mitschleppachsen sein. Auf diese Weise können unterschiedliche Mitschleppverbände aufgebaut werden. | |
| <Leitachse>: | Parameter 2: Achsbezeichnung der Leitachse | |

| | | |
|---|---|---|
| `<Koppelfaktor>`: | Parameter 3: Koppelfaktor<br>Der Koppelfaktor gibt das gewünschte Verhältnis der Wege von Mitschleppachse und Leitachse an:<br><Koppelfaktor> = Weg der Mitschleppachse / Weg der Leitachse | |
| | Typ: | REAL |
| | Voreinstellung: | 1 |
| | Die Eingabe eines negativen Wertes bewirkt eine entgegengesetzte Verfahrbewegung der Leit- und Mitschleppachse.<br>Wird der Koppelfaktor bei der Programmierung nicht angegeben, so gilt automatisch der Koppelfaktor 1. | |
| | | |
| `TRAILOF`: | Befehl zum Ausschalten eines Mitschleppverbandes | |
| | Wirksamkeit: | modal |
| | `TRAILOF` mit 2 Parametern schaltet nur die Kopplung zur angegebenen Leitachse aus:<br>`TRAILOF(<Folgeachse>,<Leitachse>)`<br>Besitzt eine Mitschleppachse 2 Leitachsen, kann zum Ausschalten der beiden Kopplungen `TRAILOF` mit 3 Parametern aufgerufen werden:<br>`TRAILOF(<Folgeachse>,<Leitachse>,<Leitachse 2>)`<br>Das gleiche Ergebnis liefert die Programmierung von `TRAILOF` ohne Angabe einer Leitachse:<br>`TRAILOF(<Folgeachse>)` | |

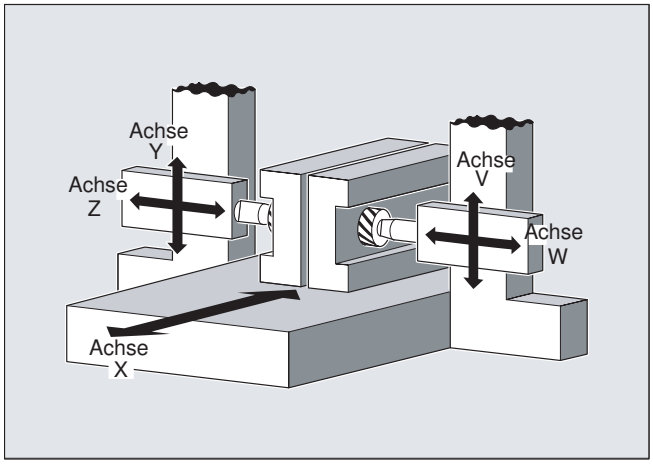
**Hinweis**

Das Mitschleppen erfolgt immer im Basiskoordinatensystem (BKS).

Die Anzahl der gleichzeitig aktivierbaren Mitschleppverbände wird nur begrenzt durch die Kombinationsmöglichkeiten der an der Maschine vorhandenen Achsen.

## Beispiel

Das Werkstück soll zweiseitig mit der dargestellten Achskonstellation bearbeitet werden. Dazu bilden Sie 2 Mitschleppverbände.



| Programmcode | Kommentar |
|---|---|
| … | |
| N100 TRAILON(V,Y) | ; Einschalten des 1. Mitschleppverbandes |
| N110 TRAILON(W,Z,-1) | ; Einschalten des 2. Mitschleppverbandes. Koppelfaktor negativ: Mitschleppachse fährt jeweils in entgegengesetzter Richtung wie Leitachse. |
| N120 G0 Z10 | ; Zustellung der Z- und W-Achse in entgegengesetzter Achsrichtung. |
| N130 G0 Y20 | ; Zustellung der Y- und V-Achse in gleicher Achsrichtung. |
| … | |
| N200 G1 Y22 V25 F200 | ; Überlagerung einer abhängigen und unabhängigen Bewegung der Mitschleppachse V. |
| … | |
| TRAILOF(V,Y) | ; Ausschalten des 1. Mitschleppverbandes. |
| TRAILOF(W,Z) | ; Ausschalten des 2. Mitschleppverbandes. |

## Weitere Informationen

### Achstypen

Ein Mitschleppverband kann aus beliebigen Kombinationen von Linear- und Rundachsen bestehen. Als Leitachse kann dabei auch eine simulierte Achse definiert werden.

### Mitschleppachsen

Einer Mitschleppachse können gleichzeitig maximal 2 Leitachsen zugeordnet werden. Die Zuordnung erfolgt in unterschiedlichen Mitschleppverbänden.

Eine Mitschleppachse kann mit allen zur Verfügung stehenden Bewegungsbefehlen programmiert werden (G0, G1, G2, G3, …). Zusätzlich zu den unabhängig definierten Wegen fährt die Mitschleppachse die mit den Koppelfaktoren aus ihren Leitachsen abgeleiteten Wege.

### Dynamikbegrenzung

Die Dynamikbegrenzung ist abhängig von der Art der Aktivierung des Mitschleppverbandes:

- Aktivierung im Teileprogramm
  Erfolgt die Aktivierung im Teileprogramm und sind alle Leitachsen als Programmachsen im aktivierenden Kanal, wird beim Verfahren der Leitachsen die Dynamik aller Mitschleppachsen so berücksichtigt, dass keine Mitschleppachse überlastet wird.
  Erfolgt die Aktivierung im Teileprogramm mit Leitachsen, die nicht als Programmachsen im aktivierenden Kanal aktiv sind ($AA_TYP ≠ 1), wird beim Verfahren der Leitachsen die Dynamik der Mitschleppachse nicht berücksichtigt. Dadurch kann es bei Mitschleppachsen mit einer geringeren als der für die Kopplung benötigten Dynamik zu einer Überlastung kommen.
- Aktivierung in Synchronaktion
  Erfolgt die Aktivierung in einer Synchronaktion, wird beim Verfahren der Leitachsen die Dynamik der Mitschleppachsen nicht berücksichtigt. Dadurch kann es bei Mitschleppachsen mit einer geringeren als der für die Kopplung benötigten Dynamik zu einer Überlastung kommen.

**⚠ VORSICHT**

**Achsüberlastung**

Wird ein Mitschleppverband

- in Synchronaktionen
- im Teileprogramm mit Leitachsen, die nicht Programmachsen im Kanal der Mitschleppachse sind,

aktiviert, dann liegt es in der besonderen Verantwortung des Anwenders/ Maschinenherstellers, geeignete Maßnahmen vorzusehen, damit es durch die Verfahrbewegungen der Leitachse nicht zu einer Überlastung der Mitschleppachsen kommt.

### Kopplungsstatus

Der Kopplungsstatus einer Achse kann im Teileprogramm abgefragt werden mit der Systemvariablen:

$AA_COUP_ACT[<Achse>]

| Wert | Bedeutung |
|---|---|
| 0 | Keine Kopplung aktiv |
| 8 | Mitschleppen aktiv |

### Restweganzeige der Mitschleppachse bei Modulo-Rundachsen

Sind Leit- und Mitschleppachse Modulo-Rundachsen, summieren sich Verfahrbewegungen in der Leitachse von n * 360° mit n = 1, 2, 3... , in der Restweganzeige der Mitschleppachse bis zum Ausschalten der Kopplung auf.

Beispiel: Programmabschnitt mit TRAILON und Leitachse B und Folgeachse C

| Programmcode | Kommentar |
|---|---|
| `TRAILON(C,B,1)` | `; Kopplung einschalten` |
| `G0 B0` | `; Ausgangsposition` |

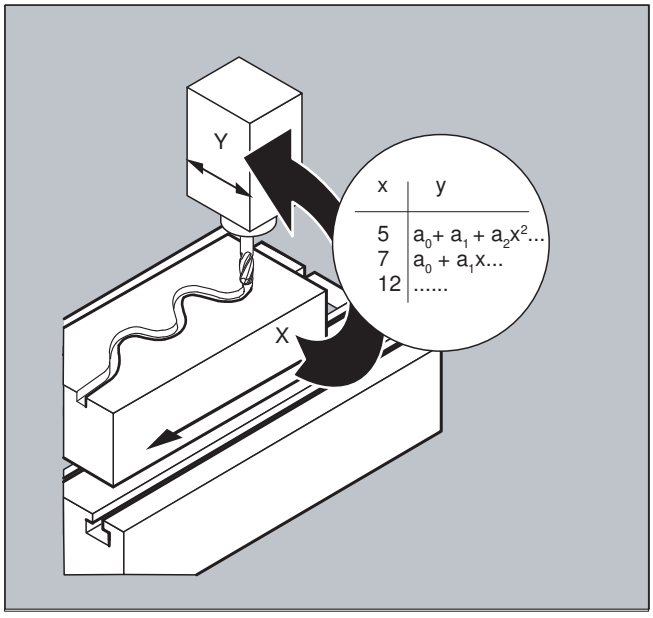
| Programmcode | Kommentar |
|---|---|
| | ; Restweganzeige am Satzanfang: |
| G91 B360 | ; B=360, C=360 |
| G91 B720 | ; B=720, C=1080 |
| G91 B360 | ; B=360, C=1440 |

## 3.16.2 Kurventabellen (CTAB)

Mit Hilfe von Kurventabellen können Positions- und Geschwindigkeitsbeziehungen zwischen zwei Achsen (Leit- und Folgeachse) programmiert werden. Die Kurventabellendefinition erfolgt im Teileprogramm.

### Anwendung

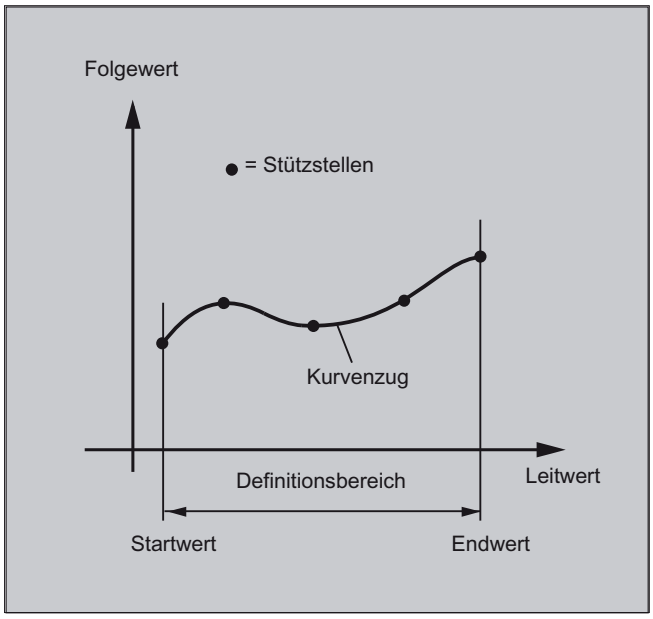
Kurventabellen ersetzen mechanische Kurvenscheiben. Die Kurventabelle bildet dabei die Grundlage für die axiale Leitwertkopplung, indem sie den funktionellen Zusammenhang zwischen Leit- und Folgewert schafft: Die Steuerung berechnet bei entsprechender Programmierung aus einander zugeordneten Positionen von Leit- und Folgeachse ein Polynom, das der Kurvenscheibe entspricht.




### 3.16.2.1 Kurventabellen definieren (CTABDEF, CATBEND)

Eine Kurventabelle stellt ein Teileprogramm oder einen Teileprogrammabschnitt dar, welcher durch Voranstellen von CTABDEF und den abschließenden Befehl CTABEND gekennzeichnet ist.

Innerhalb dieses Teileprogrammabschnitts werden durch Bewegungsanweisungen einzelnen Positionen der Leitachse eindeutige Folgeachspositionen zugeordnet, die als Stützstellen für die Berechnung eines Kurvenzugs in Form eines Polynoms bis zu maximal 5. Grades dienen.




## Voraussetzung

Für die Definition von Kurventabellen muss durch entsprechende MD-Projektierung Speicherplatz reserviert sein (→ Maschinenhersteller!).

## Syntax

```
CTABDEF(<Folgeachse>,<Leitachse>,<n>,<Periodizität>[,<Speicherort>])
...
CTABEND
```

## Bedeutung

| | |
|---|---|
| `CTABDEF( )`: | Beginn der Kurventabellendefinition |
| `CTABEND`: | Ende der Kurventabellendefinition |
| `<Folgeachse>`: | Achse, deren Bewegung über die Kurventabelle berechnet werden soll |
| `<Leitachse>`: | Achse, die die Leitwerte zur Berechnung der Folgeachsbewegung liefert |
| `<n>`: | Nummer (**ID**) der Kurventabelle<br>Die Nummer einer Kurventabelle ist eindeutig und unabhängig vom Speicherort. Es können keine Tabellen mit der gleichen Nummer im statischen und dynamischen NC-Speicher liegen. |

| | | |
|---|---|---|
| <Periodizität>: | Tabellenperiodizität | |
| | 0 | Tabelle ist nicht periodisch (wird nur einmal abgearbeitet, auch bei Rundachsen) |
| | 1 | Tabelle ist periodisch bezüglich Leitachse |
| | 2 | Tabelle ist periodisch bezüglich Leitachse und Folgeachse |
| <Speicherort>: | Angabe des Speicherorts (optional) | |
| | "SRAM" | Die Kurventabelle wird im **statischen** NC-Speicher angelegt. |
| | "DRAM" | Die Kurventabelle wird im **dynamischen** NC-Speicher angelegt. |
| | **Hinweis:**<br>Wenn für diesen Parameter kein Wert programmiert wird, dann wird der mit MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE eingestellte Standard-Speicherort verwendet. | |

**Hinweis**

**Überschreiben**

Eine Kurventabelle wird überschrieben, sobald bei einer neuen Tabellendefinition deren Nummer (<n>) benutzt wird (Ausnahme: die Kurventabelle ist in einer Achskopplung aktiv oder mit CTABLOCK gesperrt). **Beim Überschreiben wird keine entsprechende Warnung ausgegeben!**

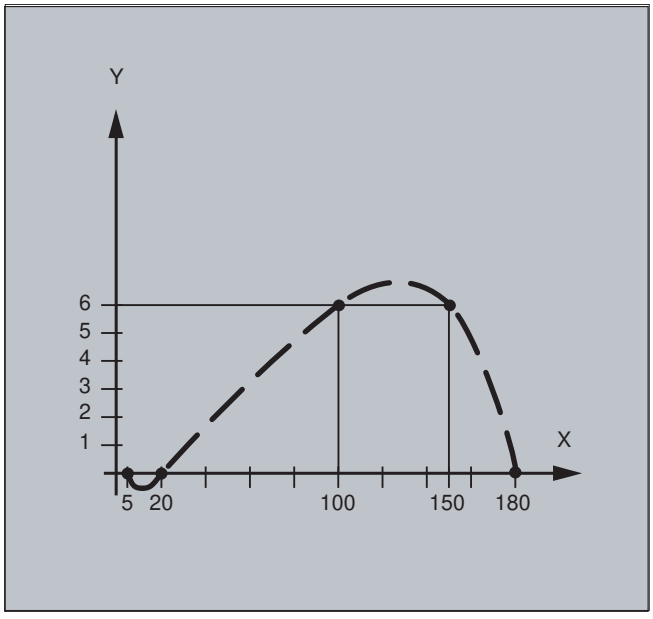
## Beispiele

### Beispiel 1: Programmabschnitt als Kurventabellendefinition

Ein Programmabschnitt soll unverändert zur Definition einer Kurventabelle benutzt werden. Der darin auftretende Befehl zum Vorlaufstopp STOPRE kann stehen bleiben und wird sofort wieder aktiv, sobald der Programmabschnitt nicht mehr zur Tabellendefinition benutzt wird und CTABDEF und CTABEND entfernt wurden.

| **Programmcode** | **Kommentar** |
|---|---|
| … | |
| CTABDEF(Y,X,1,1) | ; Definition einer Kurventabelle. |
| … | |
| IF NOT ($P_CTABDEF) | |
| STOPRE | |
| ENDIF | |
| … | |
| CTABEND | |

### Beispiel 2: Definition einer nichtperiodischen Kurventabelle




| Programmcode | Kommentar |
|---|---|
| N100 CTABDEF(Y,X,3,0) | ; Beginn der Definition einer nichtperiodischen Kurventabelle mit der Nummer 3. |
| N110 X0 Y0 | ; 1.Bewegungsanweisung, legt Startwerte und 1. Stützstelle fest:<br>Leitwert: 0, Folgewert: 0 |
| N120 X20 Y0 | ; 2.Stützstelle:<br>Leitwert: 0…20, Folgewert: Startwert…0 |
| N130 X100 Y6 | ; 3.Stützstelle:<br>Leitwert: 20…100, Folgewert: 0…6 |
| N140 X150 Y6 | ; 4.Stützstelle:<br>Leitwert: 100…150, Folgewert: 6…6 |
| N150 X180 Y0 | ; 5.Stützstelle:<br>Leitwert: 150…180, Folgewert: 6…0 |
| N200 CTABEND | ; Ende der Definition. Die Kurventabelle wird in ihrer internen Darstellung als Polynom maximal 5.Grades erzeugt. Die Berechnung des Kurvenzugs mit den angegebenen Stützstellen ist abhängig von der modal gewählten Interpolationsart (Kreis-, Linear-, Spline-Interpolation). Der Teileprogrammzustand vor Beginn der Definition wird wiederhergestellt. |

### Beispiel 3: Definition einer periodischen Kurventabelle

Definition einer periodischen Kurventabelle mit Nummer 2, Leitwertbereich von 0 bis 360, Folgeachsbewegung von 0 nach 45 und zurück nach 0:

| Programmcode | Kommentar |
|---|---|
| N10 DEF REAL DEPPOS | |

```
Programmcode                                                    Kommentar
N20 DEF REAL GRADIENT
N30 CTABDEF(Y,X,2,1)                                            ; Beginn der Definition.
N40 G1 X=0 Y=0
N50 POLY
N60 PO[X]=(45.0)
N70 PO[X]=(90.0) PO[Y]=(45.0,135.0,-90)
N80 PO[X]=(270.0)
N90 PO[X]=(315.0) PO[Y]=(0.0,-135.0,90)
N100 PO[X]=(360.0)
N110 CTABEND                                                    ; Ende der Definition.
;Test der Kurve durch Kopplung von Y an X:
N120 G1 F1000 X0
N130 LEADON(Y,X,2)
N140 X360
N150 X0
N160 LEADOF(Y,X)

N170 DEPPOS=CTAB(75.0,2,GRADIENT)                               ; Lesen der Tabellenfunktion beim
                                                                Leitwert 75.0.
N180 G0 X75 Y=DEPPOS                                            ; Positionieren von Leit- und Fol-
                                                                geachse.
;Nach Einschalten der Kopplung ist kein Synchronisieren der Folgeachse nötig.
N190 LEADON(Y,X,2)
N200 G1 X110 F1000
N210 LEADOF(Y,X)
N220 M30
```

## Weitere Informationen

### Start- und Endwert der Kurventabelle

Als Startwert für den Beginn des Definitionsbereichs der Kurventabelle gilt die erste Angabe von zusammengehörigen Achspositionen (die erste Bewegungsanweisung) innerhalb der Kurventabellendefinition. Der Endwert des Definitionsbereichs der Kurventabelle wird entsprechend durch den letzten Verfahrbefehl bestimmt.

### Verfügbarer Sprachumfang

Innerhalb der Definition der Kurventabelle steht der gesamte NC-Sprachumfang zur Verfügung.

### Hinweis

Folgende Angaben sind In Kurventabellendefinitionen nicht zulässig:

- Vorlaufstopp
- Sprünge in der Leitachsenbewegung (z. B. beim Wechsel von Transformationen)
- Bewegungsanweisung allein für die Folgeachse
- Bewegungsumkehr der Leitachse, d. h. Position der Leitachse muss immer eindeutig sein
- CTABDEF- und CTABEND-Anweisung in unterschiedlichen Programmebenen.

### Wirksamkeit von modalen Anweisungen

Sämtliche modal wirksamen Anweisungen, die innerhalb der Kurventabellendefinition getroffen werden, sind mit Abschluss der Tabellendefinition ungültig. Das Teileprogramm, in dem die Tabellendefinition erfolgt, befindet sich damit vor und nach der Tabellendefinition im gleichen Zustand.

### Zuweisungen an R-Parameter

Zuweisungen an R-Parameter innerhalb der Tabellendefinition werden nach CTABEND zurückgesetzt.

Beispiel:

| **Programmcode** | **Kommentar** |
|---|---|
| ... | |
| R10=5 R11=20 | ; R10=5 |
| ... | |
| CTABDEF | |
| G1 X=10 Y=20 F1000 | |
| R10=R11+5 | ; R10=25 |
| X=R10 | |
| CTABEND | |
| ... | ; R10=5 |

### Aktivierung von ASPLINE, BSPLINE, CSPLINE

Wird innerhalb einer Kurventabellendefinition CTABDEF ... CTABEND ein ASPLINE, BSPLINE oder CSPLINE aktiviert, so sollte vor dieser Spline-Aktivierung mindestens ein Startpunkt programmiert werden. Eine sofortige Aktivierung nach CTABDEF sollte vermieden werden, da sonst der Spline von der aktuellen Achsposition vor der Kurventabellendefinition abhängt.

Beispiel:

| **Programmcode** |
|---|
| ... |

| **Programmcode** |
| --- |
| CTABDEF(Y,X,1,0) |
| X0 Y0 |
| ASPLINE |
| X=5 Y=10 |
| X10 Y40 |
| ... |
| CTABEND |

### Wiederholte Verwendung von Kurventabellen

Der über die Kurventabelle berechnete funktionelle Zusammenhang von Leit- und Folgeachse bleibt unter der gewählten Tabellennummer über das Teileprogrammende und über POWER OFF hinaus erhalten, falls die Tabelle im statischen NC-Speicher (SRAM) abgelegt ist.

Eine Tabelle, die im dynamischen Speicher (DRAM) angelegt wurde, wird bei POWER ON gelöscht und muss eventuell noch einmal erzeugt werden.

Die einmal erstellte Kurventabelle lässt sich auf beliebige Achskombinationen von Leit- und Folgeachse anwenden und ist unabhängig davon, welche Achsen zur Erstellung der Kurventabelle benutzt wurden.

### Überschreiben von Kurventabellen

Eine Kurventabelle wird überschrieben, sobald bei einer erneuten Tabellendefinition deren Nummer benutzt wird.

Ausnahme: Eine Kurventabelle ist in einer Achskopplung aktiv oder mit CTABLOCK gesperrt.

**Hinweis**

Beim Überschreiben von Kurventabellen wird keine entsprechende Warnung ausgegeben!

### Kurventabellendefinition aktiv?

Mit der Systemvariablen `$P_CTABDEF` kann aus dem Teileprogramm heraus jederzeit abgefragt werden, ob eine Kurventabellendefinition aktiv ist.

### Aufheben der Kurventabellendefinition

Der Teileprogrammabschnitt ist nach Ausklammern der Anweisungen zur Kurventabellendefinition wieder als reales Teileprogramm verwendbar.

### Laden von Kurventabellen über "Abarbeiten von Extern"

Beim externen Abarbeiten von Kurventabellen muss die Größe des Nachladebuffers (DRAM) über MD18360 $MN_MM_EXT_PROG_BUFFER_SIZE so gewählt werden, dass die gesamte Kurventabellendefinition gleichzeitig im Nachladebuffer abgelegt werden kann. Die Teileprogrammbearbeitung wird anderenfalls mit einem Alarm abgebrochen.

### Sprünge der Folgeachse

Abhängig von der Einstellung im Maschinendatum:
MD20900 $MC_CTAB_ENABLE_NO_LEADMOTION
können Sprünge der Folgeachse bei fehlender Bewegung der Leitachse toleriert werden.

### 3.16.2.2 Vorhandensein einer Kurventabelle prüfen (CTABEXISTS)

Mit dem Befehl CTABEXISTS kann geprüft werden, ob eine bestimmte Kurventabellennummer im NC-Speicher vorhanden ist.

#### Syntax

```
CTABEXISTS(<n>)
```

#### Bedeutung

| | | |
|---|---|---|
| CTABEXISTS: | Prüft, ob die Kurventabelle mit Nummer <n> im statischen oder dynamischen NC-Speicher vorhanden ist | |
| | 0 | Tabelle existiert nicht |
| | 1 | Tabelle existiert |
| <n>: | Nummer (**ID**) der Kurventabelle | |

### 3.16.2.3 Kurventabellen löschen (CTABDEL)

Mit CTABDEL können Kurventabellen gelöscht werden.

**Hinweis**

Kurventabellen, die in einer Achskopplung aktiv sind, können nicht gelöscht werden.

#### Syntax

```
CTABDEL(<n>)
CTABDEL(<n>,<m>)
CTABDEL(<n>,<m>,<Speicherort>)
CTABDEL()
CTABDEL(,,<Speicherort>)
```

#### Bedeutung

| | |
|---|---|
| CTABDEL: | Befehl zum Löschen von Kurventabellen |
| <n>: | Nummer (**ID**) der zu löschenden Kurventabelle<br>Beim Löschen eines Kurventabellenbereichs CTABDEL(<n>,<m>) wird mit <n> die Nummer der ersten Kurventabelle des Bereichs angegeben. |
| <m>: | Beim Löschen eines Kurventabellenbereichs CTABDEL(<n>,<m>) wird mit <m> die Nummer der letzten Kurventabelle des Bereichs angegeben.<br><m> muss größer <n> sein! |

| | | |
|---|---|---|
| <Speicherort>: | Angabe des Speicherorts (optional)<br>Beim Löschen **ohne** Speicherort-Angabe werden die angegebenen Kurventabellen im statischen und dynamischen NC-Speicher gelöscht.<br>Beim Löschen **mit** Speicherort-Angabe werden von den angegebenen Kurventabellen nur diejenigen gelöscht, die im angegebenen Speicher liegen. Die übrigen bleiben bestehen. | |
| | "SRAM" | Löschen im **statischen** NC-Speicher |
| | "DRAM" | Löschen im **dynamischen** NC-Speicher |

Wird CTABDEL ohne Angabe der zu löschenden Kurventabelle programmiert, dann werden **alle** Kurventabellen bzw. alle Kurventabellen im angegebenen Speicher gelöscht:

| | |
|---|---|
| CTABDEL(): | Löscht alle Kurventabellen im statischen und dynamischen NC-Speicher |
| CTABDEL(,,"SRAM"): | Löscht alle Kurventabellen im statischen NC-Speicher |
| CTABDEL(,,"DRAM"): | Löscht alle Kurventabellen im dynamischen NC-Speicher |

**Hinweis**

Wenn beim Mehrfachlöschen CTABDEL(<n>,<m>) oder CTABDEL() wenigstens eine der zu löschenden Kurventabellen in einer Kopplung aktiv ist, dann wird der Löschbefehl nicht ausgeführt, d. h. **keine** der adressierten Kurventabellen wird gelöscht.

### 3.16.2.4 Kurventabellen gegen Löschen und Überschreiben sperren (CTABLOCK, CTABUNLOCK)

Kurventabellen können durch Setzen von Sperren vor unbeabsichtigtem Löschen und Überschreiben geschützt werden. Eine gesetzte Sperre kann jederzeit auch wieder aufgehoben werden.

## Syntax

**Sperre setzen:**

```
CTABLOCK(<n>)
CTABLOCK(<n>,<m>)
CTABLOCK(<n>,<m>,<Speicherort>)
CTABLOCK()
CTABLOCK(,,<Speicherort>)
```

**Sperre aufheben:**

```
CTABUNLOCK(<n>)
CTABUNLOCK(<n>,<m>)
CTABUNLOCK(<n>,<m>,<Speicherort>)
CTABUNLOCK()
CTABUNLOCK(,,<Speicherort>)
```

## Bedeutung

| | | |
|---|---|---|
| CTABLOCK: | Befehl zum **Setzen** einer Sperre gegen Löschen/Überschreiben | |
| CTABUNLOCK: | Befehl zum **Aufheben** einer Sperre gegen Löschen/Überschreiben<br>CTABUNLOCK gibt die mit CTABLOCK gesperrten Kurventabellen wieder frei. Kurventabellen, die in einer aktiven Kopplung wirken, bleiben weiterhin gesperrt und können nicht gelöscht werden. Die Sperre mit CTABLOCK ist aufgehoben, sobald die Sperrung durch die aktive Kopplung mit Deaktivierung der Kopplung aufgehoben wird. Damit kann diese Tabelle gelöscht werden. Ein nochmaliger CTABUNLOCK-Aufruf ist nicht notwendig. | |
| <n>: | Nummer (**ID**) der zu sperrenden/entsperrenden Kurventabelle<br>Beim Sperren/Entsperren eines Kurventabellenbereichs CTABLOCK(<n>,<m>)/CTABUNLOCK(<n>,<m>) wird mit <n> die Nummer der ersten Kurventabelle des Bereichs angegeben. | |
| <m>: | Beim Sperren/Entsperren eines Kurventabellenbereichs CTABLOCK(<n>,<m>)/CTABUNLOCK(<n>,<m>) wird mit <m> die Nummer der letzten Kurventabelle des Bereichs angegeben.<br><m> muss größer <n> sein! | |
| <Speicherort>: | Angabe des Speicherorts (optional)<br>Beim Setzen/Aufheben einer Sperre **ohne** Speicherort-Angabe werden die angegebenen Kurventabellen im statischen und dynamischen NC-Speicher gesperrt/entsperrt.<br>Beim Setzen/Aufheben einer Sperre **mit** Speicherort-Angabe werden von den angegebenen Kurventabellen nur diejenigen gesperrt/entsperrt, die im angegebenen Speicher liegen. Die übrigen werden nicht gesperrt/entsperrt. | |
| | "SRAM" | Sperre setzen/aufheben im **statischen** NC-Speicher |
| | "DRAM" | Sperre setzen/aufheben im **dynamischen** NC-Speicher |

Wird CTABLOCK/CTABUNLOCK ohne Angabe der zu sperrenden/entsperrenden Kurventabelle programmiert, dann werden **alle** Kurventabellen bzw. alle Kurventabellen im angegebenen Speicher gesperrt/entsperrt:

| | |
|---|---|
| CTABLOCK(): | Sperrt alle Kurventabellen im statischen und dynamischen NC-Speicher |
| CTABLOCK(,,"SRAM"): | Sperrt alle Kurventabellen im statischen NC-Speicher |
| CTABLOCK(,,"DRAM"): | Sperrt alle Kurventabellen im dynamischen NC-Speicher |
| | |
| CTABUNLOCK(): | Entsperrt alle Kurventabellen im statischen und dynamischen NC-Speicher |
| CTABUNLOCK(,,"SRAM"): | Entsperrt alle Kurventabellen im statischen NC-Speicher |
| CTABUNLOCK(,,"DRAM"): | Entsperrt alle Kurventabellen im dynamischen NC-Speicher |

### 3.16.2.5 Kurventabellen: Tabelleneigenschaften ermitteln (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD)

Mit diesen Befehlen können wichtige Eigenschaften einer Kurventabelle (Tabellennummer, Sperrzustand, Speicherort, Periodizität) abgefragt werden.

## Syntax

```
CTABID(<p>)
CTABID(<p>,<Speicherort>)
CTABISLOCK(<n>)
CTABMEMTYP(<n>)
TABPERIOD(<n>)
```

## Bedeutung

| | | |
|---|---|---|
| CTABID: | Liefert die **Tabellennummer**, die im angegebenen Speicher als die <p>-te Kurventabelle eingetragen ist.<br>Beispiel:<br>CTABID(1,"SRAM") liefert die Nummer der ersten Kurventabelle im statischen NC-Speicher. Die erste Kurventabelle entspricht dabei der Kurventabelle mit der höchsten Tabellennummer.<br>**Hinweis:**<br>Wird zwischen aufeinander folgenden Aufrufen von CTABID die Reihenfolge der Kurventabellen im Speicher geändert, z. B. durch Löschen von Kurventabellen mit CTABDEL, kann CTABID(<p>,...) mit derselben Nummer <p> eine andere Kurventabelle liefern als vorher. | |
| CTABISLOCK: | Gibt den **Sperrzustand** der Kurventabelle mit Nummer <n> zurück: | |
| | 0 | Tabelle ist nicht gesperrt |
| | 1 | Tabelle ist gesperrt durch CTABLOCK |
| | 2 | Tabelle ist gesperrt durch aktive Kopplung |
| | 3 | Tabelle ist gesperrt durch CTABLOCK und aktive Kopplung |
| | -1 | Tabelle existiert nicht |
| CTABMEMTYP: | Liefert den **Speicherort** der Kurventabelle mit Nummer <n>: | |
| | 0 | Tabelle im statischen NC-Speicher |
| | 1 | Tabelle im dynamischen NC-Speicher |
| | -1 | Tabelle existiert nicht |
| CTABPERIOD: | Liefert die **Periodizität** der Kurventabelle mit Nummer <n>: | |
| | 0 | Tabelle ist nicht periodisch |
| | 1 | Tabelle ist periodisch in der Leitachse |
| | 2 | Tabelle ist periodisch in der Leit- und Folgeachse |
| | -1 | Tabelle existiert nicht |
| <p>: | Eintragsnummer im Speicher | |
| <n>: | Nummer (**ID**) der Kurventabelle | |
| <Speicherort>: | Angabe des Speicherorts (optional) | |
| | "SRAM" | **Statischer** NC-Speicher |
| | "DRAM" | **Dynamischer** NC-Speicher |
| | **Hinweis:**<br>Wenn für diesen Parameter kein Wert programmiert wird, dann wird der mit MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE eingestellte Standard-Speicherort verwendet. | |

### 3.16.2.6 Kurventabellenwerte lesen (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX)

Folgende Kurventabellenwerte können im Teileprogramm gelesen werden:

- Folgeachs- und Leitachswerte am Anfang und Ende einer Kurventabelle
- Folgeachswerte am Anfang und Ende eines Kurvensegments
- Folgeachswert zu einem Leitachswert
- Leitachswert zu einem Folgeachswert
- Minimal- und Maximalwert der Folgeachse
  - im gesamten Definitionsbereich der Kurventabelle
    oder
  - in einem definierten Kurventabellenintervall

## Syntax

```
CTABTSV(<n>,<Gradient>[,<Folgeachse>])
CTABTEV(<n>,<Gradient>[,<Folgeachse>])
CTABTSP(<n>,<Gradient>[,<Leitachse>])
CTABTEP(<n>,<Gradient>[,<Leitachse>])
CTABSSV(<Leitwert>,<n>,<Gradient>[,<Folgeachse>])
CTABSEV(<Leitwert>,<n>,<Gradient>[,<Folgeachse>])
CTAB(<Leitwert>,<n>,<Gradient>[,<Folgeachse>,<Leitachse>]
CTABINV(<Folgewert>,<Näherungswert>,<n>,<Gradient>[,<Folgeachse>,<Le
itachse>]
CTABTMIN(<n>[,<Folgeachse>])
CTABTMAX(<n>[,<Folgeachse>])
CTABTMIN(<n>,<a>,<b>[,<Folgeachse>,<Leitachse>])
CTABTMAX(<n>,<a>,<b>[,<Folgeachse>,<Leitachse>])
```

## Bedeutung

| | |
|---|---|
| CTABTSV: | Folgeachswert am **Anfang** der Kurventabelle Nr. <n> lesen |
| CTABTEV: | Folgeachswert am **Ende** der Kurventabelle Nr. <n> lesen |
| CTABTSP: | Leitachswert am **Anfang** der Kurventabelle Nr. <n> lesen |
| CTABTEP: | Leitachswert am **Ende** der Kurventabelle Nr. <n> lesen |
| CTABSSV: | Folgeachswert am **Anfang** des zum angegebenen Leitachswert (<Leitwert>) gehörenden Kurvensegments lesen |
| CTABSEV: | Folgeachswert am **Ende** des zum angegebenen Leitachswert (<Leitwert>) gehörenden Kurvensegments lesen |
| CTAB: | Folgeachswert zum angegebenen Leitachswert (<Leitwert>) lesen |
| CTABINV: | Leitachswert zum angegebenen Folgeachswert (<Folgewert>) lesen |
| CTABTMIN: | **Minimalwert** der Folgeachse bestimmen:<br>• im gesamten Definitionsbereich der Kurventabelle oder<br>• in einem definierten Intervall <a> ... <b> |

| | |
|---|---|
| CTABTMAX: | **Maximalwert** der Folgeachse bestimmen:<br>• im gesamten Definitionsbereich der Kurventabelle<br>oder<br>• in einem definierten Intervall <a> ... <b> |
| <n>: | Nummer (**ID**) der Kurventabelle |
| <Gradient>: | Im Parameter <Gradient> wird die **Steigung** der Kurventabellenfunktion an der ermittelten Position zurückgegeben |
| <Folgeachse>: | Achse, deren Bewegung über die Kurventabelle berechnet werden soll (optional) |
| <Leitachse>: | Achse, die die Leitwerte zur Berechnung der Folgeachsbewegung liefert (optional) |
| <Folgewert>: | Folgeachswert zum Lesen des zugehörigen Leitachswerts bei CTABINV |
| <Leitwert>: | Leitachswert:<br>• zum Lesen des zugehörigen Folgeachswerts bei CTAB<br>oder<br>• für die Auswahl des Kurvensegments bei CTABSSV/CTABSEV |
| <Näherungswert>: | Die Zuordnung eines Leitachswerts zu einem Folgeachswert bei CTABINV muss nicht immer eindeutig sein. CTABINV benötigt daher als Parameter einen Näherungswert für den erwarteten Leitachswert. |
| <a>: | **Untere** Grenze des Leitwertintervalls bei CTABTMIN/CTABTMAX |
| <b>: | **Obere** Grenze des Leitwertintervalls bei CTABTMIN/CTABTMAX |
| | **Hinweis:**<br>Das Leitwertintervall <a> ... <b> muss innerhalb des Definitionsbereichs der Kurventabelle liegen. |

## Beispiele

### Beispiel 1:

Folgeachs- und Leitachswerte am Anfang und Ende der Kurventabelle sowie Minimal- und Maximalwert der Folgeachse im gesamten Definitionsbereich der Kurventabelle bestimmen.

```
Programmcode                          Kommentar
N10 DEF REAL STARTPOS
N20 DEF REAL ENDPOS
N30 DEF REAL STARTPARA
N40 DEF REAL ENDPARA
N50 DEF REAL MINVAL
N60 DEF REAL MAXVAL
N70 DEF REAL GRADIENT
...
N100 CTABDEF(Y,X,1,0)                 ; Beginn der Tabellendefinition
N110 X0 Y10                           ; Startposition 1.Tabellensegment
N120 X30 Y40                          ; Endposition 1.Tabellensegment =
                                      Startposition 2.Tabellensegment
N130 X60 Y5                           ; Endposition 2.Tabellensegment = ...
N140 X70 Y30
```

```
Programmcode                                Kommentar
N150 X80 Y20
N160 CTABEND                                ; Ende der Tabellendefinition.
...
N200 STARTPOS=CTABTSV(1,GRADIENT)           ; Folgeachswert am Kurventabellenanfang = 10
N210 ENDPOS=CTABTEV(1,GRADIENT)             ; Folgeachswert am Kurventabellenende = 20
N220 STARTPARA=CTABTSP(1,GRADIENT)          ; Leitachswert am Kurventabellenanfang = 0
N230 ENDPARA=CTABTEP(1,GRADIENT)            ; Leitachswert am Kurventabellenende = 80
N240 MINVAL=CTABTMIN(1)                     ; Minimalwert der Folgeachse bei Y=5
N250 MAXVAL=CTABTMAX(1)                     ; Maximalwert der Folgeachse bei Y=40
```

### Beispiel 2:

Bestimmung der Folgeachswerte am Anfang und Ende des zum Leitachswert X=30 gehörenden Kurvensegments.

```
Programmcode                                Kommentar
N10 DEF REAL STARTPOS
N20 DEF REAL ENDPOS
N30 DEF REAL GRADIENT
...
N100 CTABDEF(Y,X,1,0)                       ; Beginn der Tabellendefinition.
N110 X0 Y0                                  ; Startposition 1.Tabellensegment
N120 X20 Y10                                ; Endposition 1.Tabellensegment =
                                            Startposition 2.Tabellensegment
N130 X40 Y40                                ; Endposition 2.Tabellensegment = ...
N140 X60 Y10
N150 X80 Y0
N160 CTABEND                                ; Ende der Tabellendefinition.
...
N200 STARTPOS=CTABSSV(30.0,1,GRADIENT)      ; Startposition Y im 2.Segment = 10
N210 ENDPOS=CTABSEV(30.0,1,GRADIENT)        ; Endposition Y im 2.Segment = 40
```

## Weitere Informationen

### Verwendung in Synchronaktionen

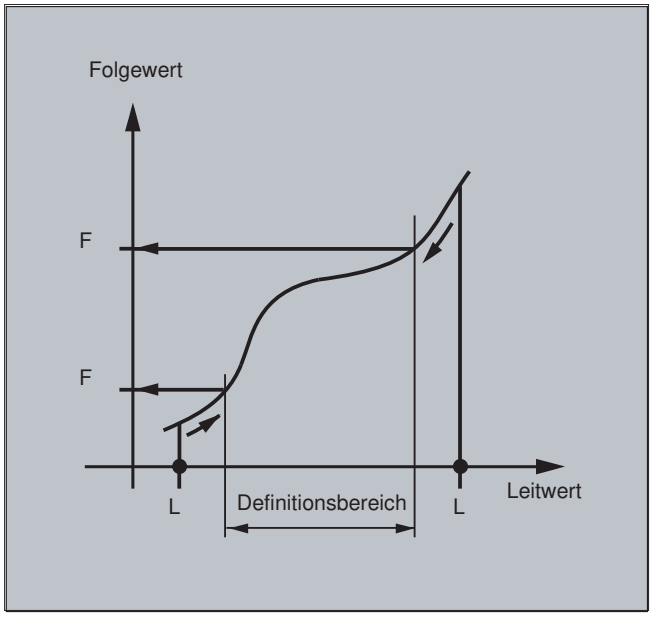
Alle Befehle zum Lesen von Kurventabellenwerten können auch in Synchronaktionen verwendet werden (siehe auch Kapitel "Bewegungssynchronaktionen").

Bei Verwendung der Befehle CTABINV, CTABTMIN und CTABTMAX ist darauf zu achten, dass:

- zum Ausführungszeitpunkt ausreichend NC-Leistung verfügbar ist
  oder
- vor dem Aufruf die Anzahl der Segmente der Kurventabelle abgefragt wird, um gegebenenfalls die betreffende Tabelle unterteilen zu können

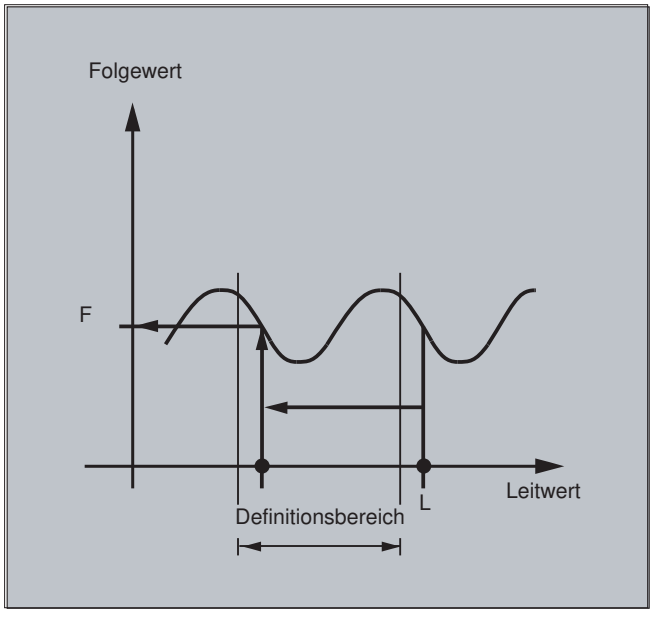
### CTAB bei nichtperiodischen Kurventabellen

Liegt der angegebene <Leitwert> außerhalb des Definitionsbereichs, wird als Folgewert die obere bzw. untere Grenze ausgegeben:




### CTAB bei periodischen Kurventabellen

Liegt der angegebene <Leitwert> außerhalb des Definitionsbereichs, wird der Leitwert Modulo des Definitionsbereichs bewertet und der entsprechende Folgewert ausgegeben:



#### Näherungswert für CTABINV

Der Befehl CTABINV benötigt einen Näherungswert für den erwarteten Leitwert. CTABINV gibt den Leitwert zurück, der dem Näherungswert am nächsten liegt. Der Näherungswert kann z. B. der Leitwert aus dem vorherigen Interpolatortakt sein.

#### Steigung der Kurventabellenfunktion

Die Ausgabe der Steigung (<Gradient>) ermöglicht es, die Geschwindigkeit der Leit- oder Folgeachse an der entsprechenden Position zu berechnen.

#### Angabe der Leit- oder Folgeachse

Die optionale Angabe der Leit- und/oder Folgeachse ist wichtig, falls Leit- und Folgeachse in verschiedenen Längeneinheiten projektiert sind.

#### CTABSSV, CTABSEV

Die Befehle CTABSSV und CTABSEV sind in folgenden Fällen **nicht** dazu geeignet, programmierte Segmente abzufragen:

- Kreise oder Evolventen sind programmiert.
- Fasen bzw. Runden mit CHF/RND ist aktiv.
- Überschleifen mit G643 ist aktiv.
- NC-Satz-Kompression (COMP...) ist aktiv.

### 3.16.2.7 Kurventabellen: Ressourcennutzung prüfen (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL)

Mit diesen Befehlen hat der Programmierer die Möglichkeit, sich aktuell über die Belegung der Ressourcen für Kurventabellen, Tabellensegmente und Polynome zu informieren.

## Syntax

```
CTABNO
CTABNOMEM(<Speicherort>)
CTABFNO(<Speicherort>)
CTABSEGID(<n>,<Speicherort>)
CTABSEG(<Speicherort>,<Segmentart>)
CTABFSEG(<Speicherort>,<Segmentart>)
CTABMSEG(<Speicherort>,<Segmentart>)
CTABPOLID(<n>)
CTABPOL(<Speicherort>)
CTABFPOL(<Speicherort>)
CTABMPOL(<Speicherort>)
```

## Bedeutung

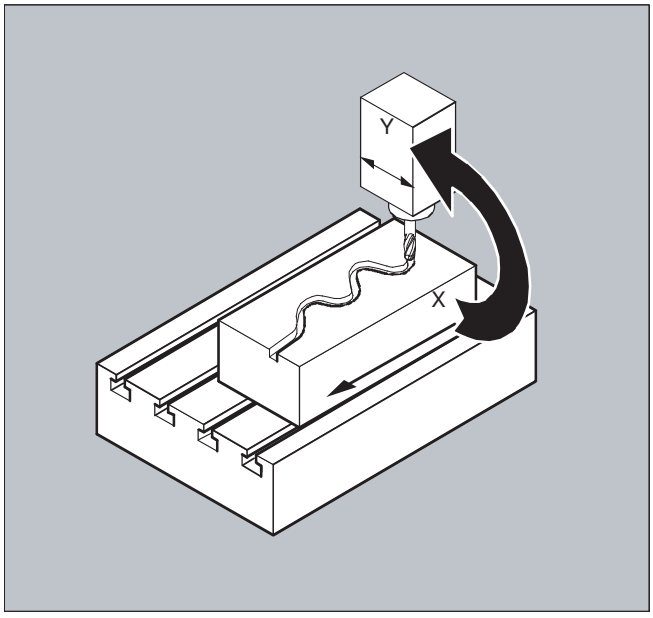
| | | |
|---|---|---|
| CTABNO: | Gesamtanzahl der **definierten** Kurventabellen bestimmen (im statischen und dynamischen NC-Speicher) | |
| CTABNOMEM: | Anzahl der **definierten** Kurventabellen im angegebenen <Speicherort> bestimmen | |
| CTABFNO: | Anzahl der **noch möglichen** Kurventabellen im angegebenen <Speicherort> bestimmen | |
| CTABSEGID: | Anzahl der Kurvensegmente der angegebenen <Segmentart> bestimmen, die von der Kurventabelle mit Nummer <n> verwendet werden | |
| CTABSEG: | Anzahl der **verwendeten** Kurvensegmente der angegebenen <Segmentart> im angegebenen <Speicherort> bestimmen | |
| CTABFSEG: | Anzahl der **noch möglichen** Kurvensegmente der angegebenen <Segmentart> im angegebenen <Speicherort> bestimmen | |
| CTABMSEG: | Anzahl der **maximal möglichen** Kurvensegmente der angegebenen <Segmentart> im angegebenen <Speicherort> bestimmen | |
| CTABPOLID: | Anzahl der Kurvenpolynome bestimmen, die von der Kurventabelle mit Nummer <n> verwendet werden | |
| CTABPOL: | Anzahl der **verwendeten** Kurvenpolynome im angegebenen <Speicherort> bestimmen | |
| CTABFPOL: | Anzahl der **noch möglichen** Kurvenpolynome im angegebenen <Speicherort> bestimmen | |
| CTABMPOL: | Anzahl der **maximal möglichen** Kurvenpolynome im angegebenen <Speicherort> bestimmen | |
| <n>: | Nummer (**ID**) der Kurventabelle | |
| <Speicherort>: | Angabe des Speicherorts (optional) | |
| | "SRAM" | **Statischer** NC-Speicher |
| | "DRAM" | **Dynamischer** NC-Speicher |
| | **Hinweis:**<br>Wenn für diesen Parameter kein Wert programmiert wird, dann wird der mit MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE eingestellte Standard-Speicherort verwendet. | |
| <Segmentart>: | Angabe der Segmentart (optional) | |
| | "L" | Lineare Segmente |
| | "P" | Polynomsegmente |
| | **Hinweis:**<br>Wenn für diesen Parameter kein Wert programmiert wird, dann wird die Summe aus Linear- und Polynom-Segmenten ausgegeben. | |

## 3.16.3 Axiale Leitwertkopplung (LEADON, LEADOF)

**Hinweis**

Diese Funktion steht für SINUMERIK 828D nicht zur Verfügung!

Bei der axialen Leitwertkopplung werden eine Leit- und eine Folgeachse synchron verfahren. Dabei ist die jeweilige Position der Folgeachse über eine Kurventabelle bzw. ein daraus berechnetes Polynom eindeutig einer - ggf. simulierten - Position der Leitachse zugeordnet.




**Leitachse** heißt diejenige Achse, die die Eingangswerte für die Kurventabelle liefert.
**Folgeachse** heißt die Achse, die die über die Kurventabelle errechneten Positionen einnimmt.

### Ist- und Sollwertkopplung

Als Leitwerte, also Ausgangswerte zur Positionsermittlung der Folgeachse können verwendet werden:

- Istwerte der Leitachsposition: Istwertkopplung
- Sollwerte der Leitachsposition: Sollwertkopplung

Die Leitwertkopplung gilt immer im Basiskoordinatensystem.

Zur Erstellung von Kurventabellen siehe Kapitel "Kurventabellen".

## Syntax

```
LEADON(<Folgeachse>,<Leitachse>,<n>)
LEADOF(<Folgeachse>,<Leitachse>)
```

oder Ausschalten ohne Angabe der Leitachse:

```
LEADOF(<Folgeachse>)
```

Die Leitwertkopplung kann sowohl vom Teileprogramm als auch während der Bewegung aus Synchronaktionen heraus ein- und ausgeschaltet werden.

## Bedeutung

| | |
|---|---|
| LEADON: | Leitwertkopplung einschalten |
| LEADOF: | Leitwertkopplung ausschalten |
| <Folgeachse>: | Folgeachse |
| <Leitachse>: | Leitachse |
| <n>: | Kurventabellen-Nummer |
| $SA_LEAD_TYPE: | Umschaltung zwischen Soll- und Istwertkopplung |

### Leitwertkopplung ausschalten, LEADOF

Mit dem Ausschalten der Leitwertkopplung wird die Folgeachse wieder zur normalen Kommandoachse!

### Axiale Leitwertkopplung und verschiedene Betriebszustände, RESET

Abhängig von der Einstellung im Maschinendatum werden Leitwertkopplungen mit RESET ausgeschaltet.

## Beispiel Leitwertkopplung aus Synchronaktion

Bei einer Pressenanlage soll eine herkömmliche mechanische Kopplung zwischen einer Leitachse (Stempelwelle) und Achsen eines Transfersystems aus Transferachsen und Hilfsachsen durch ein elektronisches Koppelsystem ersetzt werden.

Es demonstriert, wie bei einer Pressenanlage ein mechanisches Transfersystem durch ein elektronisches Transfersystem ersetzt wird. Die Kopplungs- und Entkopplungsvorgänge sind als **statische Synchronaktionen** realisiert.

Von der Leitachse LW (Stempelwelle) werden Transferachsen und Hilfsachsen als Folgeachsen über Kurventabellen definiert gesteuert.

### Folgeachsen

X Vorschub- bzw. Längsachse
YL Schließ- bzw. Querachse
ZL Hubachse
U Walzenvorschub, Hilfsachse
V Richtkopf, Hilfsachse
W Befettung, Hilfsachse

### Aktionen

Als Aktionen treten in den Synchronaktionen z. B. auf:

- Einkoppeln, `LEADON(<Folgeachse>,<Leitachse>,<Kurventabellen-Nummer>)`
- Auskoppeln, `LEADOF(<Folgeachse>,<Leitachse>)`
- Istwertsetzen, `PRESETON(<Achse>,<Wert>)`
- Merker setzen, `$AC_MARKER[i]=<Wert>`

- Kopplungsart: reeller/virtueller Leitwert
- Anfahren von Achspositionen, `POS[<Achse>]=<Wert>`

### Bedingungen

Als Bedingungen werden digitale schnelle Eingänge, Echtzeitvariablen `$AC_MARKER` und Positionsvergleiche, mit dem logischen Operator AND verknüpft, ausgewertet.

### Hinweis

Im folgenden Beispiel wurden Zeilenwechsel, Einrückungen und **Fett**satz ausschließlich dafür verwendet, die Lesbarkeit der Programmierung zu erhöhen. Für die Steuerung ist alles unter einer Zeilennummer stehende einzeilig.

### Kommentar

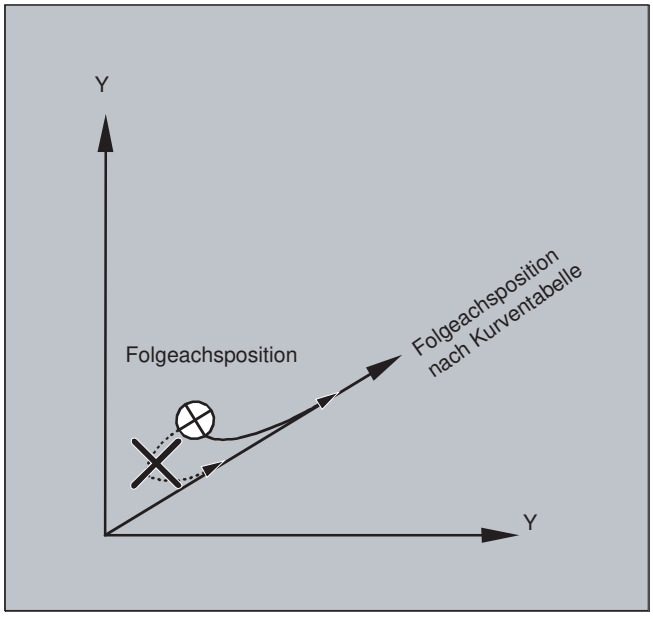
```
Programmcode                                                    Kommentar
                                                                ; Definiert sämtliche statische Synchronaktionen.
                                                                ; ****Marker rücksetzen
N2 $AC_MARKER[0]=0 $AC_MARKER[1]=0 $AC_MARKER[2]=0 $AC_MARKER[3]=0 $AC_MARKER[4]=0 $AC_MARKER[5]=0
$AC_MARKER[6]=0 $AC_MARKER[7]=0
                                                                ; **** E1 0=>1 Kopplung Transfer EIN
N10 IDS=1 EVERY ($A_IN[1]==1) AND ($A_IN[16]==1) AND ($AC_MARKER[0]==0)
DO LEADON(X,LW,1) LEADON(YL,LW,2) LEADON(ZL,LW,3) $AC_MARKER[0]=1
                                                                ; **** E1 0=>1 Kopplung Walzenvorschub EIN
N20 IDS=11 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[5]==0)
DO LEADON(U,LW,4) PRESETON(U,0) $AC_MARKER[5]=1
                                                                ; **** E1 0->1 Kopplung Richtkopf EIN
N21 IDS=12 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[6]==0)
DO LEADON(V,LW,4) PRESETON(V,0) $AC_MARKER[6]=1
                                                                ; **** E1 0->1 Kopplung Befettung EIN
N22 IDS=13 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[7]==0)
DO LEADON(W,LW,4) PRESETON(W,0) $AC_MARKER[7]=1
                                                                ; **** E2 0=>1 Kopplung AUS
N30 IDS=3 EVERY ($A_IN[2]==1)
DO LEADOF(X,LW) LEADOF(YL,LW) LEADOF(ZL,LW) LEADOF(U,LW) LEADOF(V,LW) LEADOF(W,LW) $AC_MARKER[0]=0
$AC_MARKER[1]=0 $AC_MARKER[3]=0 $AC_MARKER[4]=0 $AC_MARKER[5]=0 $AC_MARKER[6]=0 $AC_MARKER[7]=0
....
N110 G04 F01
N120 M30
```

## Beschreibung

Die Leitwertkopplung erfordert die Synchronisation von Leit- und Folgeachse. Diese Synchronisation kann nur erreicht werden, wenn die Folgeachse bei Einschalten der Leitwertkopplung innerhalb des Toleranzbereiches des aus der Kurventabelle berechneten Kurvenzugs steht.

Der Toleranzbereich für die Stellung der Folgeachse ist über Maschinendatum `MD 37200: COUPLE_POS_POL_COARSE A_LEAD_TYPE` definiert.

Befindet sich die Folgeachse mit dem Einschalten der Leitwertkopplung noch nicht an der entsprechenden Position, wird der Synchronlauf automatisch hergestellt, sobald sich der berechnete Positionssollwert für die Folgeachse der tatsächlichen Folgeachsposition nähert. Die Folgeachse wird dabei während des Synchronisationsvorganges in die Richtung verfahren, die durch die Sollgeschwindigkeit der Folgeachse (berechnet aus Leitachsgeschwindigkeit und nach Kurventabelle CTAB) definiert ist.

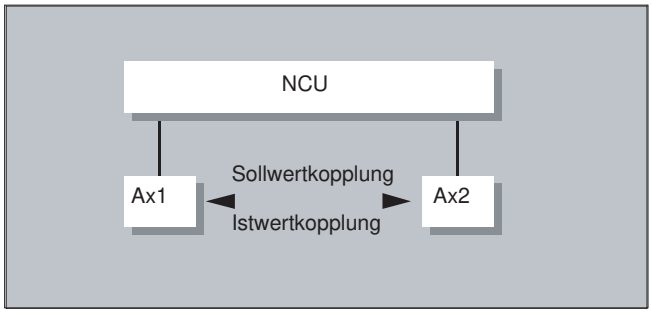



### Kein Synchronlauf

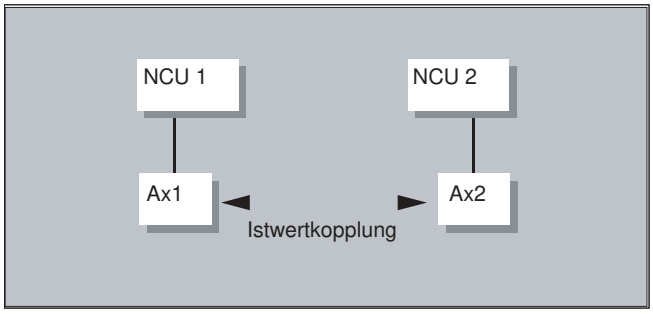
Entfernt sich die berechnete Folgeachssollposition mit Einschalten der Leitwertkopplung von der aktuellen Folgeachsposition, wird kein Synchronlauf hergestellt.

### Ist- und Sollwertkupplung

Die Sollwertkopplung liefert im Vergleich zur Istwertkopplung einen besseren Synchronlauf zwischen Leit- und Folgeachse und ist deshalb standardmäßig voreingestellt.




Sollwertkopplung ist nur möglich, wenn Leit- und Folgeachse von derselben NCU interpoliert werden. Bei einer externen Leitachse kann die Folgeachse nur über Istwerte an die Leitachse gekoppelt werden.




Eine **Umschaltung** ist über das Settingdatum $SA_LEAD_TYPE möglich.

Das Umschalten zwischen Ist- und Sollwertkopplung sollte immer bei Stillstand der Folgeachse erfolgen. Denn nur im Stillstand wird nach dem Umschalten neu synchronisiert.

### Anwendungsbeispiel

Das Lesen der Istwerte kann bei großen Maschinenerschütterungen nicht fehlerfrei erfolgen. Beim Einsatz der Leitwertkopplung im Pressentransfer kann es daher in den Arbeitsschritten mit größten Erschütterungen notwendig werden, von Istwertkopplung auf Sollwertkopplung umzuschalten.

**Leitwertsimulation** bei Sollwertkopplung

Über Maschinendatum lässt sich der Interpolator für die Leitachse vom Servo trennen. Damit können bei Sollwertkopplung Sollwerte ohne tatsächliche Bewegung der Leitachse erzeugt werden.

Die über Sollwertkopplung erzeugten Leitwerte sind zur Benutzung z. B. in Synchronaktionen aus folgenden Variablen lesbar:

| | |
|---|---|
| - $AA_LEAD_P | Leitwert Position |
| - $AA_LEAD_V | Leitwert Geschwindigkeit |

### Leitwerte erzeugen

Leitwerte können wahlweise mit anderen selbst programmierten Verfahren erzeugt werden. Die so erzeugten Leitwerte werden in die Variable

| | |
|---|---|
| - $AA_LEAD_SP | Leitwert Position |
| - $AA_LEAD_SV | Leitwert Geschwindigkeit |

geschrieben und aus ihnen gelesen. Zur Benutzung dieser Variablen muss das Settingdatum $SA_LEAD_TYPE = 2 gesetzt werden.

### Status der Kopplung

Im NC-Teileprogramm können Sie den Kopplungsstatus mit folgender Systemvariablen abfragen:

`$AA_COUP_ACT[Achse]`
0: Keine Kopplung aktiv
16: Leitwertkopplung aktiv

### Status-Verwaltung bei Synchronaktionen

Schalt- und Koppelvorgänge werden über Echtzeitvariablen:

```
$AC_MARKER[i] = n
```
verwaltet mit:
i Merker-Nummer
n Statuswert

## 3.16.4 Elektronisches Getriebe (EG)

Mit Hilfe der Funktion "Elektronisches Getriebe" ist es möglich, die Bewegung einer **Folgeachse** nach linearem Bewegungssatz abhängig von bis zu fünf **Leitachsen** zu steuern. Die Zusammenhänge zwischen den Leitachsen und der Folgeachse sind je Leitachse durch den Koppelfaktor definiert.

Der berechnete Folgeachs-Bewegungsanteil wird aus den einzelnen Leitachsen-Bewegungsanteilen multipliziert mit den jeweiligen Koppelfaktoren durch Addition gebildet. Bei der Aktivierung eines EG-Achsverbundes kann die Synchronisation der Folgeachse auf eine definierte Position veranlasst werden. Ein Getriebeverband kann aus dem Teileprogramm:

- definiert,
- eingeschaltet,
- ausgeschaltet,
- gelöscht

werden.

Die Folgeachsbewegung kann wahlweise abgeleitet werden aus den

- Sollwerten der Leitachsen sowie den
- Istwerten der Leitachsen.

Als Erweiterung können auch nichtlineare Zusammenhänge zwischen den Leitachsen und der Folgeachse über **Kurventabellen** (siehe Kapitel Bahnverhalten) realisiert werden.
Elektronische Getriebe können kaskadiert werden, d. h. die Folgeachse eines Elektronischen Getriebes kann Leitachse für ein weiteres Elektronisches Getriebe sein.

### 3.16.4.1 Elektronisches Getriebe definieren (EGDEF)

Ein EG-Achsverband wird durch die Angabe der Folgeachse und mindestens einer, jedoch höchstens fünf Leitachsen mit dem jeweiligen Kopplungstyp festgelegt.

#### Voraussetzung

Voraussetzung für eine EG-Achsverband-Definition:

Für die Folgeachse darf noch keine Achskopplung definiert sein (ggf. muss eine bestehende vorher mit EGDEL gelöscht werden).

#### Syntax

```
EGDEF(Folgeachse,Leitachse1,Kopplungstyp1,Leitachse2,Kopplungstyp2,...)
```

## Bedeutung

| | | |
|---|---|---|
| `EGDEF`: | Definition eines elektronischen Getriebes | |
| `Folgeachse`: | Achse, die von Leitachsen beeinflusst wird | |
| `Leitachse1 ,..., Leitachse5` | Achsen, die die Folgeachse beeinflussen | |
| `Kopplungstyp1 ,..., Kopplungstyp5` | Kopplungstyp<br>Der Kopplungstyp muss nicht für alle Leitachsen gleich sein und ist daher für jede Leitachse einzeln anzugeben. | |
| | **Wert:** | **Bedeutung:** |
| | 0 | Die Folgeachse wird beeinflusst vom **Istwert** der entsprechenden Leitachse. |
| | 1 | Die Folgeachse wird beeinflusst vom **Sollwert** der entsprechenden Leitachse. |

### Hinweis

Die Koppelfaktoren werden bei der Definition des EG-Kopplungsverbandes mit Null vorbesetzt.

### Hinweis

EGDEF löst Vorlaufstopp aus. Die Getriebedefinition mit EGDEF ist auch dann unverändert zu verwenden, wenn bei Systemen eine oder mehrere Leitachsen über **Kurventabelle** auf die Folgeachse einwirken.

## Beispiel

| **Programmcode** | **Kommentar** |
|---|---|
| `EGDEF(C,B,1,Z,1,Y,1)` | `; Definition eines EG-Achsverbandes. Die Leitachsen B, Z, Y beeinflussen die Folgeachse C über den Sollwert.` |

## 3.16.4.2 Elektronisches Getriebe einschalten (EGON, EGONSYN, EGONSYNE)

Für das Einschalten eines EG-Achsverbandes existieren 3 Varianten.

## Syntax

### Variante 1:

Der EG-Achsverband wird ohneSynchronisation selektiv eingeschaltet mit:
`EGON(FA,"Satzwechselmodus",LA1,Z1,N1,LA2,Z2,N2,...,LA5,Z5,N5)`

### Variante 2:

Der EG-Achsverband wird mitSynchronisation selektiv eingeschaltet mit:
`EGONSYN(FA,"Satzwechselmodus",SynPosFA,[,LAi,SynPosLAi,Zi,Ni])`

**Variante 3:**

Der EG-Achsverband wird mitSynchronisation selektiv eingeschaltet und der Anfahrmodus vorgegeben mit:
`EGONSYNE(FA,"Satzwechselmodus",SynPosFA,Anfahrmodus[,LAi,SynPosLAi,Zi,Ni])`

## Bedeutung

**Variante 1:**

| | | |
|---|---|---|
| FA | Folgeachse | |
| `Satzwechselmodus`: | Folgende Modi können benutzt werden: | |
| | `"NOC"` | Satzwechsel erfolgt sofort |
| | `"FINE"` | Satzwechsel erfolgt bei "Synchronlauf fein" |
| | `"COARSE"` | Satzwechsel erfolgt bei "Synchronlauf grob" |
| | `"IPOSTOP"` | Satzwechsel erfolgt bei sollwertseitigem Synchronlauf |
| LA1, ... LA5 | Leitachsen | |
| Z1, ... Z5 | Zähler für den Koppelfaktor i | |
| N1, ... N5 | Nenner für den Koppelfaktor i<br>Koppelfaktor i = Zähler i/Nenner i | |

Es dürfen nur die Leitachsen programmiert werden, die zuvor mit EGDEF spezifiziert worden sind. Es muss mindestens eine Leitachse programmiert werden.

**Variante 2:**

| | | |
|---|---|---|
| `FA` | Folgeachse | |
| `Satzwechselmodus`: | Folgende Modi können benutzt werden: | |
| | `"NOC"` | Satzwechsel erfolgt sofort |
| | `"FINE"` | Satzwechsel erfolgt bei "Synchronlauf fein" |
| | `"COARSE"` | Satzwechsel erfolgt bei "Synchronlauf grob" |
| | `"IPOSTOP"` | Satzwechsel erfolgt bei sollwertseitigem Synchronlauf |
| `[,LAi,SynPosLAi,Zi,Ni]` | (Eckige Klammern nicht schreiben)<br>Mind. 1, max. 5 Folgen von: | |
| `LA1, ... LA5` | Leitachsen | |
| `SynPosLAi` | Synchronposition für die i. Leitachse | |
| `Z1, ... Z5` | Zähler für den Koppelfaktor i | |
| `N1, ... N5` | Nenner für den Koppelfaktor i<br>Koppelfaktor i = Zähler i/Nenner i | |

Es dürfen nur Leitachsen programmiert werden, die zuvor mit EGDEF spezifiziert worden sind. Durch die programmierten "Synchronpositionen" für die Folgeachse (SynPosFA) und für die Leitachsen (SynPosLA) werden Positionen definiert, in denen der Koppelverband als *synchron* gilt. Sofern sich das elektronische Getriebe beim Einschalten nicht in synchronem Zustand befindet, fährt die Folgeachse auf ihre definierte Synchronposition.

**Variante 3:**

Die Parameter entsprechen denen der Variante 2 zuzüglich:

| | | |
|---|---|---|
| `Anfahrmodus`: | Folgende Modi können benutzt werden: | |
| | `"NTGT"` | Nächste Zahnlücke zeitoptimiert anfahren |
| | `"NTGP"` | Nächste Zahnlücke wegoptimiert anfahren |
| | `"ACN"` | Rundachse in negativer Drehrichtung verfahren absolut |
| | `"ACP"` | Rundachse in positiver Drehrichtung verfahren absolut |
| | `"DCT"` | Zeitoptimiert zur programmierten Synchronposition |
| | `"DCP"` | Wegoptimiert zur programmierten Synchronposition |

Die Variante 3 hat nur Auswirkungen auf Modulo-Folgeachsen, die an Modulo-Leitachsen gekoppelt sind. Zeitoptimierung berücksichtigt die Geschwindigkeitsgrenzen der Folgeachse.

## Weitere Informationen

**Beschreibung der Einschaltvarianten**

Variante 1:

Die Positionen der Leitachsen sowie der Folgeachse zum Zeitpunkt des Einschaltens werden gespeichert als "Synchronpositionen". Die "Synchronpositionen" können mit den Systemvariablen `$AA_EG_SYN` gelesen werden.

Variante 2:

Wenn Modulo-Achsen im Koppelverband sind, werden ihre Positionswerte modulo reduziert. Damit ist gewährleistet, dass die nächstmögliche Synchronposition angefahren wird (sog. *relative Synchronisation*: z. B. die nächste Zahnlücke). Wenn für die Folgeachse nicht "Freigabe Folgeachsüberlagerung" Nahtstellensignal DB(30 +Achsnummer), DBX 26 Bit 4 gegeben ist, wird nicht auf die Synchronposition gefahren. Stattdessen wird das Programm beim EGONSYN-Satz angehalten und es wird der selbstlöschende Alarm 16771 gemeldet, solange bis das o.g. Signal gesetzt wird.

Variante 3:

Der Zahnabstand (Grad) ergibt sich aus: 360 * Zi/Ni. Für den Fall, dass die Folgeachse zum Aufrufzeitpunkt steht, liefert wegoptimiert das gleiche Verhalten wie zeitoptimiert.

Bei bereits fahrender Folgeachse wird mit NTGP unabhängig von der aktuellen Geschwindigkeit der Folgeachse auf die nächste Zahnlücke synchronisiert. Bei bereits fahrender Folgeachse wird mit NTGT abhängig von der aktuellen Geschwindigkeit der Folgeachse auf die nächste Zahnlücke synchronisiert. Die Achse wird dazu ggf. auch abgebremst.

**Kurventabellen**

Wird für eine der Leitachsen eine **Kurventabelle** verwendet, so muss:

Ni der Nenner des Koppelfaktors linearer Kopplungen auf 0 gesetzt werden. (Nenner 0 wäre für lineare Kopplungen unzulässig). Nenner Null ist für die Steuerung das Kennzeichen, dass

Zi als Nummer der zu verwendenden Kurventabelle interpretiert werden soll. Die Kurventabelle mit der angegebenen Nummer muss zum Einschaltzeitpunkt bereits definiert sein.

LAi Die Angabe der Leitachse entspricht der Leitachsangabe bei Kopplung über Koppelfaktor (lineare Kopplung).

Weitere Hinweise über die Nutzung von Kurventabellen und das Kaskadieren von Elektronischen Getrieben und deren Synchronisierung finden Sie in:
**Weitere Funktionen:**
Funktionshandbuch Achsen und Spindeln

**Verhalten des Elektronischen Getriebes bei Power On, RESET, Betriebsartenwechsel, Satzsuchlauf**

- Nach Power On ist **keine** Kopplung aktiv.
- Aktive Kopplungen bleiben über RESET und Betriebsartenwechsel erhalten.
- Bei Satzsuchlauf werden Befehle zum Schalten, Löschen, Definieren des Elektronischen Getriebes nicht ausgeführt und nicht aufgesammelt, sondern übergangen.

**Systemvariablen des Elektronischen Getriebes**

Mit Hilfe der Systemvariablen des Elektronischen Getriebes kann das Teileprogramm aktuelle Zustände eines EG-Achsverbandes ermitteln und ggf. darauf reagieren.

Die Systemvariablen des Elektronischen Getriebes sind wie folgt gekennzeichnet:

$AA_EG_ ...

oder

$VA_EG_ ...

**Weitere Informationen**
Funktionshandbuch Systemvariablen

### 3.16.4.3 Elektronisches Getriebe ausschalten (EGOFS, EGOFC)

Für das Ausschalten eines aktiven EG-Achsverbandes existieren 3 Varianten.

## Programmierung

**Variante 1:**

| Syntax | Bedeutung |
|---|---|
| `EGOFS(Folgeachse)` | Das elektronische Getriebe wird ausgeschaltet. Die Folgeachse wird zum Stillstand abgebremst. Der Aufruf löst Vorlaufstopp aus. |

**Variante 2:**

| Syntax | Bedeutung |
|---|---|
| `EGOFS(Folgeachse,Leitachse1, …,Leitachse5)` | Diese Parametrierung des Befehls erlaubt **selektiv** den Einfluss einzelner Leitachsen auf die Bewegung der Folgeachse zu beseitigen. |

Es muss wenigstens eine Leitachse angegeben werden. Der Einfluss der angegebenen Leitachsen auf die Folgeachse wird gezielt ausgeschaltet. Der Aufruf löst Vorlaufstopp aus. Verbleiben noch aktive Leitachsen, so läuft die Folgeachse unter deren Einfluss weiter. Sind alle Leitachseneinflüsse auf diese Weise ausgeschaltet, so wird die Folgeachse zum Stillstand abgebremst.

**Variante 3:**

| Syntax | Bedeutung |
|---|---|
| `EGOFC(Folgespinde1)` | Das elektronische Getriebe wird ausgeschaltet. Die Folgespindel läuft mit der zum Ausschaltzeitpunkt aktuellen Drehzahl/Geschwindigkeit weiter. Der Aufruf löst Vorlaufstopp aus. |

**Hinweis**

Diese Variante ist nur für Spindeln erlaubt.

### 3.16.4.4 Definition eines Elektronischen Getriebes löschen (EGDEL)

Ein EG-Achsverband muss ausgeschaltet sein, bevor seine Definition gelöscht werden kann.

## Programmierung

| Syntax | Bedeutung |
|---|---|
| `EGDEL(Folgeachse)` | Die Kopplungsdefinition des Achsverbandes wird gelöscht. Es wird bis zum Erreichen der maximalen Anzahl gleichzeitig aktivierter Achsverbände wieder möglich, weitere Achsverbände mit `EGDEF` neu zu definieren. Der Aufruf löst Vorlaufstopp aus. |

### 3.16.4.5 Umdrehungsvorschub (G95) / Elektronisches Getriebe (FPR)

Mit dem FPR-Befehl kann auch die Folgeachse eines Elektronischen Getriebes als vorschubbestimmende Achse des Umdrehungsvorschubes angegeben werden. Für diesen Fall gilt folgendes Verhalten:

- Der Vorschub ist abhängig von der Sollgeschwindigkeit der Folgeachse des Elektronischen Getriebes.
- Die Sollgeschwindigkeit wird berechnet aus den Geschwindigkeiten der Leitspindeln und Modulo-Leitachsen (die nicht Bahnachsen sind) und deren zugeordneten Koppelfaktoren.
- Geschwindigkeitsanteile von linearen bzw. nicht Modulo-Leitachsen und überlagerte Bewegungen der Folgeachse werden nicht berücksichtigt.

## 3.16.5 Synchronspindel

Im Synchronbetrieb gibt es eine Leitspindel (LS) und eine Folgespindel (FS), das sog. **Synchronspindelpaar**. Die Folgespindel folgt bei aktiver Kopplung (Synchronbetrieb) den Bewegungen der Leitspindel entsprechend dem festgelegten Funktionszusammenhang.

Die Synchronspindelpaare lassen sich für jede Maschine sowohl mit Hilfe von kanalspezifischen Maschinendaten fest projektieren oder über das CNC-Teileprogramm anwendungsspezifisch definieren. Je NC-Kanal sind bis zu 2 Synchronspindelpaare gleichzeitig betreibbar.

Die Kopplung kann aus dem Teileprogramm

- definiert bzw. geändert
- eingeschaltet
- ausgeschaltet
- gelöscht

werden.

Darüber hinaus kann abhängig vom Softwarestand

- auf die Synchronlaufbedingung gewartet
- das Satzwechselverhalten verändert
- die Kopplungsart entweder Sollwertkopplung oder Istwertkopplung ausgewählt oder der Winkelversatz zwischen Leit- und Folgespindel vorgegeben
- beim Einschalten der Kopplung eine vorhergehende Programmierung der Folgespindel übernommen
- entweder eine gemessene oder eine bereits bekannte Synchronlaufabweichung korrigiert

werden.

### 3.16.5.1 Synchronspindel: Programmierung (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC)

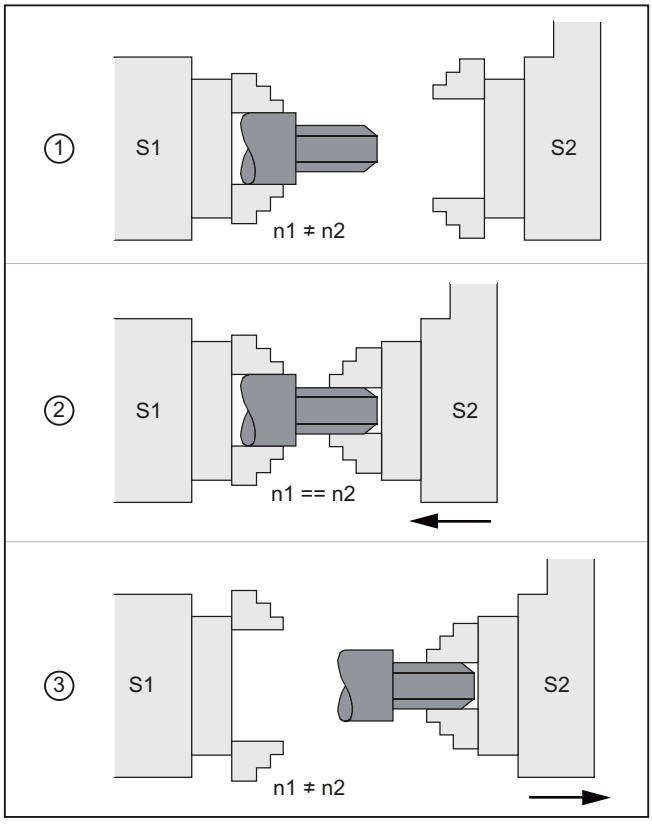
Die Funktion "Synchronspindel" ermöglicht ein drehzahlsynchrones Verfahren von Folge- (FS) und Leitspindel (LS) mit programmierbarem Übersetzungsverhältniss.

Die Funktion bietet folgende Modi:

- Drehzahlsynchronität ($n_{FS} = n_{LS}$)
- Lagesynchronität ($\phi_{FS} = \phi_{LS}$)
- Lagesynchronität mit Winkelversatz ($\phi_{FS} = \phi_{LS} + \Delta\phi$)

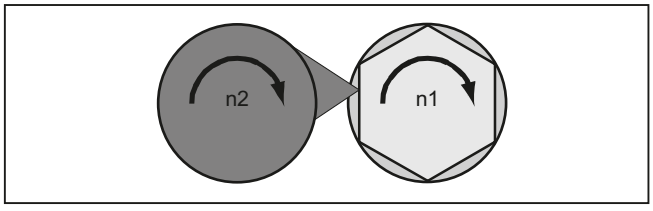
Anwendungsbeispiele:

- Fliegende Werkstückübergabe z.B. zur Rückseitenbearbeitung, Übersetzungsverhältnis: 1:1




① Drehzahl synchronisieren
② Werkstück übergeben
③ Rückseite bearbeiten

- Mehrkantbearbeitung (Polygondrehen), Drehzahlsynchronität, Übersetzungsverhältnis: $n_1$:$n_2$




## Syntax

```
COUPDEF(<FS>,<LS>,<ZFS>,<NLS>,<Satzwechsel>,<Koppelart>)
COUPON(<FS>,<LS>,<POSFS>)
COUPONC(<FS>,<LS>)
COUPOF(<FS>,<LS>,<POSFS>,<POSLS>)
COUPOFS(<FS>,<LS>)
COUPOFS(<FS>,<LS>,<POSFS>)
COUPRES(<FS>,<LS>)
```

```
COUPDEL(<FS>,<LS>)
WAITC(<FS>,<Satzwechsel>,<LS>,<Satzwechsel>)
```

**Hinweis**

**Verkürzte Schreibweise**

Bei den Anweisungen COUPOF, COUPOFS, COUPRES und COUPDEL ist eine verkürzte Schreibweise ohne Angabe der Leitspindel möglich.

## Bedeutung

| | |
|---|---|
| COUPDEF: | Kopplung anwenderspezifisch definieren/ändern |
| COUPON: | Kopplung einschalten. Ausgehend von der aktuellen Drehzahl synchronisiert sich die Folgespindel auf die Leitspindel |
| COUPONC: | Kopplung beim Einschalten mit vorhergehender Programmierung von M3 S... oder M4 S... übernehmen.<br>Eine Differenzdrehzahl der Folgespindel wird sofort übernommen. |
| COUPOF: | Kopplung ausschalten.<br>• mit sofortigem Satzwechsel:<br>COUPOF(<S2>,<S1>)<br>• Satzwechsel erst nach Überfahren der Ausschaltposition(en) <POSFS> bzw. <POSLS>:<br>COUPOF(<S2>,<S1>,<POSFS>)<br>COUPOF(<S2>,<S1>,<POSFS>,<POSLS>) |
| COUPOFS: | Ausschalten einer Kopplung mit Stopp der Folgespindel.<br>Satzwechsel schnellstmöglich mit sofortigen Satzwechsel:<br>COUPOFS(<S2>,<S1>)<br>Satzwechsel erst nach Überfahren der Ausschaltposition:<br>COUPOFS(<S2>,<S1>,<POSFS>) |
| COUPRES: | Kopplungsparameter zurücksetzen auf projektierte MD und SD |
| COUPDEL: | Anwenderdefinierte Kopplung löschen |
| WAITC: | Warten Synchronlaufbedingung<br>(NOC werden auf IPO bei Satzwechsel aufgehoben) |
| <FS>: | Bezeichnung der Folgespindel |
| **Optionale Parameter:** | |
| <LS>: | Bezeichnung der Leitspindel<br>Angabe mit Spindelnummer: z. B. S2, S1 |
| <ZFS>, <NLS>: | Übersetzungsverhältnis zwischen FS und LS.<br><ZFS> / <NLS> = Zähler / Nenner<br>Voreinstellung: <ZFS> / <NLS> = 1.0 ; Angabe des Nenners optional |

| | | |
|---|---|---|
| <Satzwechsel>: | Satzwechselverhalten<br>Der Satzwechsel erfolgt: | |
| | "NOC" | sofort |
| | "FINE" | mit Erreichen von "Synchronlauf fein" |
| | "COARSE" | mit Erreichen von "Synchronlauf grob" |
| | "IPOSTOP" | mit Erreichen von IPOSTOP, d. h. nach sollwertseitigem Synchronlauf (Voreinstellung) |
| | Das Satzwechselverhalten ist modal wirksam. | |
| <Koppelart>: | Kopplungsart: Kopplung zwischen FS und LS | |
| | "DV" | Sollwertkopplung (Voreinstellung) |
| | "AV" | Istwertkopplung |
| | "VV" | Geschwindigkeitskopplung |
| | Die Kopplungsart ist modal wirksam. | |
| <POSFS>: | Winkelversatz zwischen Leit- und Folgespindel | |
| | Wertebereich: | 0°… 359,999° |
| <POSFS>,<POSLS>: | Ausschaltpositionen von Folge- und Leitspindel<br>"Der Satzwechsel wird nach überfahren der $POS_{FS}$, $POS_{LS}$ freigegeben" | |
| | Wertebereich: | 0°… 359,999° |

## Beispiele

### Arbeiten mit Leit- und Folgespindel

| **Programmcode** | **Kommentar** |
|---|---|
| | Leitspindel = Masterspindel = Spindel 1 |
| | Folgespindel = Spindel 2 |
| N05 M3 S3000 M2=4 S2=500 | Leitspindel dreht mit 3000 U/min,<br>Folgespindel dreht mit 500 U/min. |
| N10 COUPDEF(S2,S1,1,1,"NOC","Dv") | Definition der Kopplung (kann auch projektiert werden). |
| ... | |
| N70 SPCON | Leitspindel in Lageregelung nehmen (Sollwertk.). |
| N75 SPCON(2) | Folgespindel in Lageregelung nehmen. |
| N80 COUPON(S2,S1,45) | Fliegend auf Offsetposition = 45 Grad einkoppeln. |
| ... | |
| N200 FA[S2]=100 | Positioniergeschwindigkeit = 100 grd/min |
| N205 SPOS[2]=IC(-90) | 90 Grad überlagert in negative Richtung fahren. |
| N210 WAITC(S2,"Fine") | Warten auf Synchronlauf "fein". |
| N212 G1 X... Y... F... | Bearbeitung |
| ... | |
| N215 SPOS[2]=IC(180) | 180 Grad überlagert in positive Richtung fahren. |
| N220 G4 S50 | Verweilzeit = 50 Umdrehungen der Masterspindel |
| N225 FA[S2]=0 | Projektierte Geschw. (MD) aktivieren. |
| N230 SPOS[2]=IC(-7200) | 20 Umdrehungen. Mit projektierter Geschwindigkeit in negative Richtung fahren. |

| Programmcode | Kommentar |
|---|---|
| ... | |
| N350 COUPOF(S2,S1) | Fliegend auskoppeln, S=S2=3000 |
| N355 SPOSA[2]=0 | FS bei Null Grad stoppen. |
| N360 G0 X0 Y0 | |
| N365 WAITS(2) | Warten auf Spindel 2. |
| N370 M5 | FS stoppen. |
| N375 M30 | |

### Programmierung einer Differenzdrehzahl

| Programmcode | Kommentar |
|---|---|
| | Leitspindel = Masterspindel = Spindel 1 |
| | Folgespindel = Spindel 2 |
| N01 M3 S500 | Leitspindel dreht mit 500 U/min. |
| N02 M2=3 S2=300 | Folgespindel dreht mit 300 U/min. |
| ... | |
| N10 G4 F1 | Verweilzeit der Masterspindel. |
| N15 COUPDEF (S2,S1,-1) | Koppelfaktor mit Übersetzungsverhältnis -1:1 |
| N20 COUPON (S2,S1) | Kopplung aktivieren. Die Drehzahl der Folgespindel ergibt sich aus der Drehzahl der Leitspindel und dem Koppelfaktor. |
| ... | |
| N26 M2=3 S2=100 | Programmierung einer Differenzdrehzahl. |

### Beispiele der Übernahme einer Bewegung zur Differenzdrehzahl

1. Kopplung bei vorhergehender Programmierung der Folgespindel mit COUPON einschalten

| Programmcode | Kommentar |
|---|---|
| | Leitspindel = Masterspindel = Spindel 1 |
| | Folgespindel = Spindel 2 |
| N05 M3 S100 M2=3 S2=200 | Leitspindel dreht mit 100 U/min, Folgespindel mit 200 U/min. |
| N10 G4 F5 | Verweilzeit = 5 Sekunden der Masterspindel |
| N15 COUPDEF(S2,S1,1) | Übersetzungsverhältnis FS zu LS ist 1,0 (Voreinstellung). |
| N20 COUPON(S2,S1) | Fliegend auf Leitspindel einkoppeln. |
| N10 G4 F5 | Folgespindel dreht mit 100 U/min. |

2. Kopplung bei vorhergehender Programmierung der Folgespindel mit COUPONC einschalten

| Programmcode | Kommentar |
|---|---|
| | Leitspindel = Masterspindel = Spindel 1 |
| | Folgespindel = Spindel 2 |
| N05 M3 S100 M2=3 S2=200 | Leitspindel dreht mit 100 U/min, Folgespindel mit 200 U/min. |
| N10 G4 F5 | Verweilzeit = 5 Sekunden der Masterspindel |

| Programmcode | Kommentar |
|---|---|
| N15 COUPDEF(S2,S1,1) | Übersetzungsverhältnis FS zu LS ist 1,0 (Voreinstellung). |
| N20 COUPONC(S2,S1) | Fliegend auf Leitspindel einkoppeln und vorhergehende Drehzahl zu S2 übernehmen. |
| N10 G4 F5 | S2 dreht mit 100U/min + 200U/min = 300U/min |

3. Kopplung bei stehender Folgespindel mit COUPON einschalten

| Programmcode | Kommentar |
|---|---|
| | Leitspindel = Masterspindel = Spindel 1 |
| | Folgespindel = Spindel 2 |
| N05 SPOS=10 SPOS[2]=20 | Folgespindel S2 im Positionierbetrieb. |
| N15 COUPDEF(S2,S1,1) | Übersetzungsverhältnis FS zu LS ist 1,0 (Voreinstellung). |
| N20 COUPON(S2,S1) | Fliegend auf Leitspindel einkoppeln. |
| N10 G4 F1 | Kopplung wird geschlossen, S2 bleibt auf 20 Grad stehen. |

4. Kopplung bei stehender Folgespindel mit COUPONC einschalten

**Hinweis**

**Positionier- oder Achsbetrieb**

Befindet sich die Folgespindel vor dem Einkoppeln im Positionier- oder Achsbetrieb, dann verhält sich die Folgespindel bei COUPON(<FS>,<LS>) und COUPONC(<FS>,<LS>) gleich.

**Hinweis**

**Leitspindel und Achsbetrieb**

Befindet sich die Leitspindel vor der Definition der Kopplung im Achsbetrieb, wirkt auch nach dem Einschalten der Kopplung der Geschwindigkeitsgrenzwert aus Maschinendatum:

MD32000 $MA_MAX_AX_VELO (maximale Achsgeschwindigkeit)

Zur Vermeidung dieses Verhaltens muss die Achse vor der Definition der Kopplung in den Spindelbetrieb (M3 S... oder M4 S...) geschaltet werden.

## Weitere Informationen

### Projektierte Kopplung

Bei der projektierten Kopplung werden LS und FS über Maschinendaten festgelegt. Die projektierten Spindeln können im Teileprogramm nicht verändert werden. Die Parametrierung der Kopplung kann mit COUPDEF im Teileprogramm erfolgen (Voraussetzung: kein Schreibschutz festgelegt).

### Anwenderdefinierte Kopplung

Mit COUPDEF kann eine Kopplung im Teileprogramm neu definiert oder verändert werden. Ist bereits eine Kopplung aktiv, muss diese vor der Definition einer neuen Kopplung zuerst mit COUPDEL gelöscht werden.

Eine Kopplung wird vollständig definiert durch:

```
COUPDEF(<FS>,<LS>,<ÜFS>,<ÜLS>, Satzwechselverhalten, Koppelart)
```

### Folgespindel (FS) und Leitspindel (LS)

Mit den Achsnamen für die FS und LS wird die Kopplung eindeutig bestimmt. Die Achsnamen müssen mit jeder Anweisung `COUPDEF` programmiert werden. Die anderen Kopplungsparameter sind modal wirksam und müssen nur programmiert werden, wenn sie geändert werden.

Beispiel:

```
COUPDEF(S2,S1)
```

### Übersetzungsverhältnis

Das Übersetzungsverhältnis wird als Drehzahlverhältnis zwischen FS und LS angegeben:

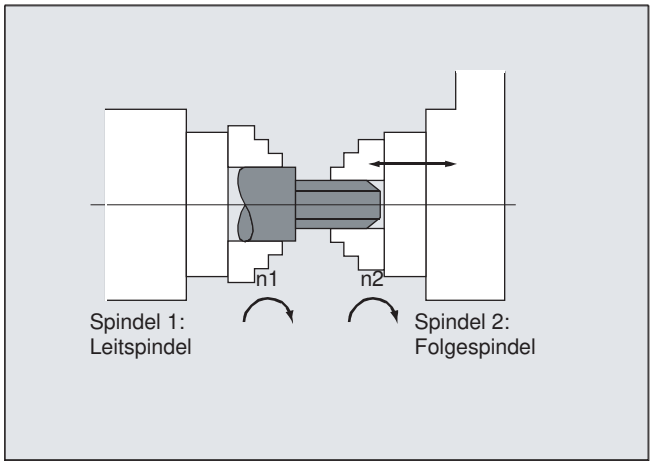
Folgespindel / Leitspindel = Zähler / Nenner

Der Zähler muss programmiert werden. Der Nenner muss nicht programmiert werden. Für den Nenner wird dann der Defaultwert 1.0 gesetzt.

Beispiel:

Folgespindel S2 und Leitspindel S1, Übersetzungsverhältnis = 1 / 1

```
COUPDEF(S2, S1, 1.0)
```




**Hinweis**

Das Übersetzungsverhältnis kann auch bei eingeschalteter Kopplung und drehenden Spindeln verändert werden.

### Satzwechselverhalten NOC, FINE, COARSE, IPOSTOP

Bei der Programmierung des Satzwechselverhaltens ist folgende verkürzte Schreibweise möglich:

- `"NO"`: sofort (Voreinstellung)
- `"FI"`: mit Erreichen von "Synchronlauf fein"

- "CO": mit Erreichen von "Synchronlauf grob"
- "IP": mit Erreichen von IPOSTOP, d. h. nach sollwertseitigem Synchronlauf

### Kopplungsart

**Hinweis**

Die Kopplungsart darf nur bei ausgeschalteter Kopplung verändert werden.

### Synchronbetrieb einschalten COUPON, <POSFS>

- Einschalten der Kopplung mit beliebigem Winkelversatz zwischen LS und FS:
  - COUPON(S2,S1)
  - COUPON(S2)
- Einschalten der Kopplung mit Winkelversatz <POSFS>
  <POSFS> bezieht sich auf die 0°-Position der Leitspindel in positiver Drehrichtung
  Wertebereich <POSFS>: 0°… 359,999°
  - COUPON(S2,S1,30)

**Hinweis**

Der Winkelversatz kann auch bei aktiver Kopplung geändert werden.

### Positionieren der Folgespindel

Auch bei eingeschalteter Synchronspindelkopplung lassen sich FS unabhängig von der LS im Bereich ±180° positionieren.

- Spindelpositionierung der FS mit SPOS
  Beispiel: SPOS[2]=IC(-90)
  Weitere Informationen zu SPOS finden sich in:
  **Weitere Informationen**
  Spindeln positionieren (SPOS, SPOSA, M19, M70, WAITS) (Seite 121)

### Differenzdrehzahl

Eine Differenzdrehzahl entsteht im Drehzahlsteuerbetrieb und aktiver Synchronspindelkopplung durch vorzeichenbehaftete Überlagerung einer FS-Drehzahl aufgrund LS-Bewegung und einer FS-Drehzahl aufgrund Spindelprogrammierung:

- Synchronspindelkopplung mit `COUPONC`
- `S<FS>=<Drehzahl>` [ `M<FS>=<Drehrichtung>]`

**Hinweis**

Randbedingungen

- Mit der Drehrichtung `M3` / `M4` muss auch die Drehzahl `S...` neu programmiert werden.
- Die Überlagern einer Spindeldrehzahl ( `M<Drehrichtung> S<FS>`) durch die LS-Bewegung bei Synchronspindelkopplung `COUPONC` wird nur wirksam, wenn die Überlagerung freigegeben ist.
- Die Dynamik der Leitspindel muss so weit eingeschränkt werden, dass bei einer Überlagerung der Folgespindel deren Dynamikgrenzwerte nicht überschritten werden.

Weitere Informationen zur Differenzdrehzahl siehe:
**Weitere Informationen**
Funktionshandbuch Achsen und Spindeln

### Geschwindigkeit, Beschleunigung: FA, ACC, OVRA, VELOLIMA

Axiale Geschwindigkeit und Beschleunigung einer Folgespindeln sind programmierbar mit:

- `FA[SPI(S<n>)]` bzw. `FA[S<n>]` (axiale Geschwindigkeit)
- `ACC[SPI(S<n>)]` bzw. `ACC[S<n>]` (axiale Beschleunigung)
- `OVRA[SPI(S<n>)]` bzw. `OVRA[S<n>]` (axialer Override)
- `VELOLIMA[SPI(S<n>)]` bzw. `VELOLIMA[S<n>]` (axiale Geschwindigkeitsüberhöhung bzw. -reduktion)

Mit `<n>` = 1, 2, 3, ... (Spindelnummer der Folgespindeln)

**Weitere Informationen**
Vorschubregelung (Seite 108)

**Hinweis**

Eine Reduktion oder Überhöhung des maximalen axialen Rucks ist bei Spindeln nicht wirksam.

Weitere Informationen zur axialen Dynamik finden sich in:
**Weitere Informationen**
Funktionshandbuch Achsen und Spindeln

### Programmierbares Satzwechselverhalten WAITC

Mit `WAITC` kann das Satzwechselverhalten, z. B. nach Änderung von Kopplungsparametern oder Positioniervorgängen, mit unterschiedlichen Synchronlaufbedingungen (grob, fein, `IPOSTOP`) vorgegeben werden. Sind keine Synchronlaufbedingungen angegeben, gilt das bei der Definition `COUPDEF` angegebene Satzwechselverhalten.

Beispiele

- Warten auf das Erreichen der Synchronlaufbedingung `FINE` bei Folgespindel S2 und `COARSE` bei Folgespindel S4: `WAITC(S2,"FINE",S4,"COARSE")`
- Warten auf das Erreichen der Synchronlaufbedingung entsprechend `COUPDEF`: `WAITC( )`

### Kopplung ausschalten COUPOF

Mit `COUPOF` kann das Ausschaltverhalten der Kopplung vorgegeben werden:

- Ausschalten der Kopplung mit sofortigem Satzwechsel:
  - `COUPOF(S2,S1)` (mit Angabe der Leitspindel)
  - `COUPOF(S2)` (ohne Angabe der Leitspindel)
- Ausschalten der Kopplung nach Überfahren von Ausschaltpositionen. Der Satzwechsel erfolgt nach dem Überfahren der Ausschaltpositionen.
  - `COUPOF(S2,S1,150)` (Ausschaltposition FS: 150°)
  - `COUPOF(S2,S1,150,30)` (Ausschaltposition FS: 150°, LS: 30°)

### Kopplung ausschalten mit Stopp der Folgespindel COUPOFS

Mit `COUPOFS` kann das Ausschaltverhalten der Kopplung mit Stopp der Folgespindel vorgegeben werden:

- Ausschalten der Kopplung mit Stopp der Folgespindel und sofortigem Satzwechsel:
  - `COUPOFS(S2,S1)` (mit Angabe der Leitspindel)
  - `COUPOFS(S2)` (ohne Angabe der Leitspindel)
- Ausschalten der Kopplung nach Überfahren von Ausschaltpositionen mit Stopp der Folgespindel. Der Satzwechsel erfolgt nach dem Überfahren der Ausschaltpositionen.
  - `COUPOFS(S2,S1,150)` (Ausschaltposition FS: 150°)

### Kopplungen löschen COUPDEL

Mit `COUPDEL` wird die Kopplung gelöscht:

- `COUPDEL(S2,S1)` (mit Angabe der Leitspindel)
- `COUPDEL(S2)` (ohne Angabe der Leitspindel)

### Kopplungsparameter zurücksetzen COUPRES

Mit `COUPRES` werden die in den Maschinen- und Settingdaten parametrierten Werte der Kopplung aktiviert:

- `COUPRES(S2,S1)` (mit Angabe der Leitspindel)
- `COUPRES(S2)` (ohne Angabe der Leitspindel)

### Systemvariablen

- Aktueller Kopplungszustand der Folgespindel
  Der aktuelle Kopplungszustand einer Folgespindel kann bitcodiert gelesen werden über:
  `<Wert> = $AA_COUP_ACT[<FS>]`

| Bit | <Wert> | Bedeutung |
|---|---|---|
| - | 0 | keine Kopplung aktiv |
| 2 | 4 | Synchronspindelkopplung aktiv |

**Hinweis**

- Alle andere Werte beziehen sich auf Achsbetrieb
- Ist die Spindel eine Folgespindel von mehreren Kopplungen, wird als Wert der Kopplungszustand aller Kopplungen als Summenzustand zurückgegeben.

- Aktueller Winkelversatz
  Der aktuelle Winkelversatz der Folgespindel zur Leitspindel kann gelesen werden über:
  - `$AA_COUP_OFFS[<FS>]` (Sollwertseitiger Winkelversatz)
  - `$VA_COUP_OFFS[<FS>]` (Istwertseitiger Winkelversatz)

  Anwendungsbeispiel
  Korrektur der Winkelversatzdifferenz im NC-Programm nach Aufheben des Nachführbetriebs:
  Winkelversatzdifferenz = Programmierter Winkelversatz - Systemvariable

### Weitere Informationen

Ausführliche Informationen zu Systemvariablen finden sich in:

Listenhandbuch Systemvariablen

## 3.16.6 Generische Kopplung (CP...)

Die "Generische Kopplung" ist eine allgemeine Kopplungsfunktion, in der alle Kopplungseigenschaften der bestehenden Kopplungsarten (Mitschleppen, Leitwertkopplung, Elektronisches Getriebe und Synchronspindel) zusammengefasst sind.

Die Funktion ermöglicht eine flexible Art der Programmierung:

- Der Anwender kann die für seine Applikation notwendigen Kopplungseigenschaften frei auswählen (Baukastenprinzip).
- Jede Kopplungseigenschaft ist einzeln programmierbar.
- Die Kopplungseigenschaften einer definierten Kopplung (z. B. Koppelfaktor) sind änderbar.
- Eine spätere Nutzung weiterer Kopplungseigenschaften ist möglich.
- Das Koordinatenbezugssystem der Folgeachse (Basiskoordinatensystem oder Maschinenkoordinatensystem) ist programmierbar.
- Bestimmte Kopplungseigenschaften können auch in Synchronaktionen programmiert werden.
  **Weitere Informationen:** Funktionshandbuch Synchronaktionen

**Hinweis**

Die bisherigen Kopplungsaufrufe für das Mitschleppen (TRAIL*), Leitwertkopplung (LEAD*), Elektronisches Getriebe (EG*) und Synchronspindel (COUP*) werden über Anpasszyklen weiterhin unterstützt.

## Übersicht aller Schlüsselwörter und Kopplungseigenschaften

Die folgende Tabelle stellt eine Übersicht aller Schlüsselwörter der Generischen Kopplung und der damit programmierbaren Kopplungseigenschaften dar:

| Schlüsselwort | Kopplungseigenschaft / Bedeutung | Syntax | | |
|---|---|---|---|---|
| CPDEF | Anlegen eines Koppelmoduls | CPDEF=(<FAx>) | | |
| CPDEL | Löschen eines Koppelmoduls | CPDEL=(<FAx>) | | |
| CPLA | Definition einer Leitachse | CPLA[<FAx>]=(<LAx>) | | |
| CPLDEF | Definition einer Leitachse und Anlegen eines Koppelmoduls<br>(auch möglich mit CPDEF + CPLA) | CPLDEF[<FAx>]=(<LAx>)<br>oder<br>CPDEF=(<FAx>) CPLA[<FAx>]=(<LAx>) | | |
| CPLDEL | Löschen einer Leitachse eines Koppelmoduls<br>(auch möglich mit CPDEL + CPLA) | CPLDEL[<FAx>]=(<LAx>)<br>oder<br>CPDEL=(<FAx>) CPLA[<FAx>]=(<LAx>) | | |
| | | | | |
| CPON | Einschalten eines Koppelmoduls | CPON=(<FAx>) | | |
| CPOF | Ausschalten eines Koppelmoduls | CPOF=(<FAx>) | | |
| CPLON | Einschalten einer Leitachse eines Koppelmoduls | CPLON[<FAx>]=<LAx> | | |
| CPLOF | Ausschalten einer Leitachse eines Koppelmoduls | CPLOF[<FAx>]=<LAx> | | |
| | | | | |
| CPLNUM | Zähler des Koppelfaktors | CPLNUM[FAx,LAx]=<Wert> | | |
| CPLDEN | Nenner des Koppelfaktors | CPLDEN[FAx,LAx]=<Wert> | | |
| CPLCTID | Nummer der Kurventabelle | CPLCTID[FAx,LAx]=<Wert> | | |
| | | | | |
| CPLSETVAL | Kopplungsbezug | CPLSETVAL[FAx,LAx]="<Kopplungsbezug>" | | |
| | | "<Kopplungsbezug>": | "CMDPOS" | Sollwertkopplung |
| | | | "CMDVEL" | Geschwindigkeitskopplung |
| | | | "ACTPOS" | Istwertkopplung |
| CPFRS | Koordinatenbezugssystem | CPFRS[FAx]="<Koordinatenbezug>" | | |
| | | "<Koordinatenbezug>": | "BCS" | Basiskoordinatensystem |
| | | | "MCS" | Maschinenkoordinatensystem |

| Schlüsselwort | Kopplungseigenschaft / Bedeutung | Syntax | | |
|---|---|---|---|---|
| CPBC | Satzwechselkriterium | CPBC[FAx]="<Satzwechselkriterium>" | | |
| | | "<Satzwechsel-kriterium>": | "NOC" | Satzwechsel erfolgt unabhängig vom Kopplungszustand. |
| | | | "IPOSTOP" | Satzwechsel erfolgt beim sollwertseitigen Synchronlauf. |
| | | | "COARSE" | Satzwechsel erfolgt beim istwertseitigen Synchronlauf "Grob". |
| | | | "FINE" | Satzwechsel erfolgt beim istwertseitigen Synchronlauf "Fein". |
| | | | | |
| CPFPOS + CPON | Synchronposition der Folgeachse beim Einschalten | CPON=FAx CPFPOS[FAx]=<Wert> | | |
| CPLPOS + CPON | Synchronposition der Leitachse beim Einschalten | CPLPOS[FAx,LAx]=<Wert> | | |

| Schlüsselwort | Kopplungseigenschaft / Bedeutung | Syntax | | |
|---|---|---|---|---|
| CPFMSON | Synchronisationsmodus | CPFMSON[FAx]="<Synchronisationsmodus>" | | |
| | | "<Synchronisationsmodus>": | "CFAST" | Die Kopplung wird zeitoptimiert geschlossen. |
| | | | "CCOARSE" | Die Kopplung wird erst eingeschaltet, wenn sich die gemäß Koppelgesetz geforderte Folgeachsposition im Bereich der aktuellen Folgeachsposition befindet. |
| | | | "NTGT" | Die nächste Zahnlücke wird zeitoptimiert angefahren. |
| | | | "NTGP" | Die nächste Zahnlücke wird wegoptimiert angefahren. |
| | | | "NRGT" | Das nächste Segment gemäß Verhältnis aus Gangzahl zu Zähnezahl wird zeitoptimiert angefahren. |
| | | | "NRGP" | Das nächste Segment gemäß Verhältnis aus Gangzahl zu Zähnezahl wird wegoptimiert angefahren. |
| | | | "ACN" | Nur bei Rundachsen!<br>Die Rundachse verfährt in negativer Achsrichtung die Synchronposition an. Die Synchronisation erfolgt sofort. |
| | | | "ACP" | Nur bei Rundachsen!<br>Die Rundachse verfährt in positiver Achsrichtung die Synchronposition an. Die Synchronisation erfolgt sofort. |
| | | | "DCT" | Nur bei Rundachsen!<br>Die Rundachse verfährt zeitoptimiert zur pen Synchronposition an. Die Synchronisation erfolgt sofort. |
| | | | "DCP" | Nur bei Rundachsen!<br>Die Rundachse verfährt wegoptimiert zur programmierten Synchronposition an. Die Synchronisation erfolgt sofort. |
| | | | | |

| Schlüsselwort | Kopplungseigenschaft / Bedeutung | Syntax | | |
|---|---|---|---|---|
| CPFMON | Verhalten der Folgeachse beim Einschalten | CPFMON[FAx]="<Einschaltverhalten>" | | |
| | | "<Einschalt-verhalten>": | "STOP" | Nur bei Spindeln!<br>Eine aktive Bewegung der Folgespindel wird vor dem Einschalten gestoppt. |
| | | | "CONT" | Nur bei Spindeln und Hauptlaufachsen!<br>Die aktuelle Bewegung der Folgeachse/-spindel wird in die Kopplung als Startbewegung übernommen. |
| | | | "ADD" | Nur bei Spindeln!<br>Die Bewegungsanteile der Kopplung wirken zusätzlich zur aktuellen überlagerten Bewegung, d. h. die aktuelle Bewegung der Folgeachse/-spindel wird als überlagerte Bewegung beibehalten. |
| CPFMOF | Verhalten der Folgeachse beim vollständigen Ausschalten | CPFMOF[FAx]="<Ausschaltverhalten>" | | |
| | | "<Ausschalt-verhalten>": | "STOP" | Stopp der Folgeachse/-spindel.<br>Eine aktive überlagerte Bewegung wird zum Stillstand abgebremst. Danach wird die Kopplung geöffnet |
| | | | "CONT" | Nur bei Spindeln und Hauptlaufachsen!<br>Die Folgespindel läuft mit der zum Ausschaltzeitpunkt aktuellen Drehzahl/ Geschwindigkeit weiter. |
| CPFPOS + CPOF | Ausschaltposition der Folgeachse beim Ausschalten | CPOF=(FAx) CPFPOS[FAx]=<Wert> | | |
| | | | | |

| Schlüsselwort | Kopplungseigenschaft / Bedeutung | Syntax | | |
|---|---|---|---|---|
| CPMRESET | Kopplungsverhalten bei RESET | CPMRESET[FAx]="<Reset-Verhalten>" | | |
| | | "<Reset-Verhalten>": | "NONE" | Der aktuelle Zustand der Kopplung bleibt erhalten. |
| | | | "ON" | Ist das entsprechende Koppelmodul angelegt, so wird die Kopplung eingeschaltet. Es werden alle definierten Leitachsbeziehungen aktiviert. Dies erfolgt auch, wenn bereits alle oder Teile dieser Leitachsbeziehungen aktiv sind, d. h. auch bei einer vollständig aktivierten Kopplung erfolgt eine Neusynchronisation. |
| | | | "OF" | Eine aktive überlagerte Bewegung wird zum Stillstand abgebremst. Danach wird die Kopplung ausgeschaltet. Wurde das entsprechende Koppelmodul ohne explizite Definition (CPDEF) angelegt, so wird das Koppelmodul gelöscht. Im anderen Fall bleibt es weiterhin angelegt, d. h. es kann weiterhin verwendet werden. |
| | | | "OFC" | Nur bei Spindeln möglich!<br>Die Folgespindel läuft mit der zum Ausschaltzeitpunkt aktuellen Drehzahl/ Geschwindigkeit weiter. Die Kopplung wird ausgeschaltet. Wurde das entsprechende Koppelmodul ohne explizite Definition (CPDEF) angelegt, so wird das Koppelmodul gelöscht. Im anderen Fall bleibt es weiterhin angelegt, d. h. es kann weiterhin verwendet werden. |
| | | | "DEL" | Eine aktive überlagerte Bewegung wird zum Stillstand abgebremst. Danach wird die Kopplung deaktiviert und anschließend gelöscht. |
| | | | "DELC" | Nur bei Spindeln möglich!<br>Die Folgespindel läuft mit der zum Ausschaltzeitpunkt aktuellen Drehzahl/ |

| Schlüsselwort | Kopplungseigenschaft / Bedeutung | Syntax | | |
|---|---|---|---|---|
| | | | | Geschwindigkeit weiter. Die Kopplung wird deaktiviert und anschließend gelöscht. |
| CPMSTART | Kopplungsverhalten beim Teileprogrammstart | CPMSTART[FAx]="<Start-Verhalten>" | | |
| | | "<Start-Verhalten>": | "NONE" | Der aktuelle Zustand der Kopplung bleibt erhalten. |
| | | | "ON" | Kopplung eingeschaltet. Es werden alle definierten Leitachsbeziehungen aktiviert. Dies erfolgt auch, wenn bereits alle oder Teile dieser Leitachsbeziehungen aktiv sind, d. h. auch bei einer vollständig aktivierten Kopplung erfolgt eine Neusynchronisation. |
| | | | "OF" | Die Kopplung wird ausgeschaltet. Wurde das entsprechende Koppelmodul ohne explizite Definition (CPDEF) angelegt, so wird das Koppelmodul gelöscht. Im anderen Fall bleibt es weiterhin angelegt, d. h. es kann weiterhin verwendet werden. |
| | | | "DEL" | Die Kopplung wird deaktiviert und anschließend gelöscht. |
| CPMPRT | Kopplungsverhalten beim Teileprogrammstart unter Satzsuchlauf via Programmtest | CPMPRT[FAx]="<Start-Verhalten>" | | |
| | | "<Start-Verhalten>": | siehe CPMSTART | |
| | | | | |
| CPLINTR | Verschiebewert für den Eingangswert einer Leitachse | CPLINTR[FAx,LAx]=<Wert> | | |
| CPLINSC | Skalierfaktor für den Eingangswert einer Leitachse | CPLINSC[FAx,LAx]=<Wert> | | |
| CPLOUTTR | Verschiebewert für den Ausgangswert einer Kopplung | CPLOUTTR[FAx,LAx]=<Wert> | | |
| CPLOUTSC | Skalierfaktor für den Ausgangswert einer Kopplung | CPLOUTSC[FAx,LAx]=<Wert> | | |
| | | | | |
| CPSYNCOP | Schwellwert für den Positionssynchronlauf "Grob" | CPSYNCOP[FAx]=<Wert> | | |
| CPSYNFIP | Schwellwert für den Positionssynchronlauf "Fein" | CPSYNFIP[FAx]=<Wert> | | |
| CPSYNCOP2 | Zweiter Schwellwert für den Positionssynchronlauf "Grob" | CPSYNCOP2[FAx]=<Wert> | | |

| Schlüsselwort | Kopplungseigenschaft / Bedeutung | Syntax | | |
|---|---|---|---|---|
| CPSYNFIP2 | Zweiter Schwellwert für den Positionssynchronlauf "Fein" | CPSYNFIP2[FAx]=<Wert> | | |
| CPSYNCOV | Schwellwert für den Geschwindigkeitssynchronlauf "Grob" | CPSYNCOV[FAx]=<Wert> | | |
| CPSYNFIV | Schwellwert für den Geschwindigkeitssynchronlauf "Fein" | CPSYNFIV[FAx]=<Wert> | | |
| | | | | |
| CPMBRAKE | Verhalten der Folgeachse bei bestimmten Stoppsignalen und -kommandos | CPMBRAKE[FAx]=<Bitcodierter Wert> | | |
| CPMVDI | Verhalten der Folgeachse auf bestimmte NC/PLC-Nahtstellensignale | CPMVDI[FAx]=<Bitcodierter Wert> | | |
| CPMALARM | Unterdrückung spezieller kopplungsbezogener Alarmausgaben | CPMALARM[FAx]=<Bitcodierter Wert> | | |
| | | | | |
| CPSETTYPE | Kopplungstyp | CPSETTYPE[FAx]="<Kopplungstyp>" | | |
| | | "<Kopplungstyp>": | "CP" | Freie Programmierbarkeit |
| | | | "TRAIL" | Kopplungstyp "Mitschleppen" |
| | | | "LEAD" | Kopplungstyp "Leitwertkopplung" |
| | | | "EG" | Kopplungstyp "Elektronisches Getriebe" |
| | | | "COUP" | Kopplungstyp "Synchronspindel" |

FAx: Folgeachse/-spindel

LAx: Leitachse/-spindel

**Hinweis**

Kopplungseigenschaften, die nicht explizit programmiert werden (im Teileprogramm oder in Synchronaktionen), werden mit ihren Standardeinstellungen wirksam.

Abhängig von der Einstellung des Schlüsselworts CPSETTYPE können statt der Standardeinstellungen (CPSETTYPE="CP") auch voreingestellte Kopplungseigenschaften wirksam werden.

## Weitere Informationen

Für ausführliche Informationen zur Generischen Kopplung siehe:

- Funktionshandbuch Achsen und Spindeln

## 3.16.7 Tangentialsteuerung

### 3.16.7.1 Kopplung definieren (TANG)

Über die vordefinierte Prozedur TANG(...) wird eine Tangentialkopplung zwischen einer Rundachse als Folgeachse und zwei Geometrieachsen als Leitachsen definiert. Die Folgeachse wird dabei kontinuierlich zur Bahntangente der Leitachsen ausgerichtet.

**Hinweis**

**Koppelfaktor**

Ein Koppelfaktor von 1 muss nicht explizit programmiert werden.
Durch den Koppelfaktor -1 wird die Richtung der Tangentialachse gedreht.

### Syntax

```
TANG(<Folgeachse>, <Leitachse_1>, <Leitachse_2>, <Koppelfaktor>,
<Koordinatensystem>, <Optimierung>)
```

### Bedeutung

| | | | |
|---|---|---|---|
| `TANG(...)`: | Definieren einer Tangentialkopplung | | |
| `<Folgeachse>`: | Achsename der Folgeachse (Rundachse) | | |
| | Datentyp: | AXIS | |
| | Wertebereich: | Kanalachsnamen | |
| `<Leitachse_1>` `<Leitachse_2>`: | Achsenamen der Leitachsen (Geometrieachsen) [1] | | |
| | Datentyp: | AXIS | |
| | Wertebereich: | Geometrieachsnamen des Kanals | |
| `<Koppelfaktor>`: | Faktor n der Winkeländerung der Folgeachse zur Änderung der Bahntangente der Leitachsen:<br>$\text{Winkeländerung}_{\text{Folgeachse}} = \text{Winkeländerung}_{\text{Bahntangente}} * n$ | | |
| | Datentyp: | REAL | |
| | Defaultwert: | 1,0 | |
| `<Koordinatensystem>`: | Wirksames Koordinatensystem [2] | | |
| | Datentyp: | CHAR | |
| | Wert: | `"B"`: | Basiskoordinatensystem (Defaultwert) |
| | | `"W"`: | Werkstückkoordinatensystem (nicht verfügbar) |

| | | | |
|---|---|---|---|
| <Optimierung>: | Optimierungsart | | |
| | Datentyp: | CHAR | |
| | Wert: | "S": | Standard (Defaultwert)<br>Die Dynamik der Rundachse hat keine Rückwirkung auf die Leitachsen. Ist die Dynamik der Rundachse höher als für die Nachführung erforderlich, ist dieses Verfahren ausreichend genau.Ist die Dynamik der Rundachse nicht hoch genug, um der Änderung der Bahntangenten zu folgen, weicht die Ausrichtung der Rundachse entlang eines nicht definierten Überschleifwegs von der Sollausrichtung ab. |
| | | "P": | Die Dynamik der Rundachse wird bei der Bahnplanung der Leitachsen berücksichtigt.<br>Dazu müssen beim Einschalten der Tangentialkopplung mit TANGON() zwei zusätzliche Parameter angegeben werden:<br>• Überschleifweg<br>• Winkeltoleranz<br>Siehe Kapitel "Kopplung einschalten (TANGON) (Seite 960)"<br>**Hinweis**<br>Im Zusammenhang mit kinematischen Transformationen wird empfohlen, die Optimierungsart "P" zu verwenden. |
| **Hinweis**<br>Defaultwerte müssen nicht explizit programmiert werden. | | | |
| [1] **Hinweis**<br>Als Leitachsen der Tangentialkopplung müssen die Geometrieachsen verwendet werden, die, bezogen auf die Grundstellung der Maschine, im Maschinenkoordinatensystem (MKS) die programmierte Bahn abfahren. Wird z. B. auf einer Fräsmaschine mit Schwenkkopf der Schwenkzyklus CYCLE800 verwendet, erfolgt, abhängig von der Konfiguration des Zyklus, die Interpolation im WKS z. B. mit den Geometrieachsen X und Y. Die Tangentialkopplung muss aber mit den Geometrieachsen als Leitachsen definiert werden, die im MKS die programmierte Bahn abfahren. Dazu sind als Leitachsen die Geometrieachsen im **ungeschwenkten** Zustand der Maschine zu verwenden. | | | |
| [2] **Hinweis**<br>Das Basiskoordinatensystem (BKS) darf gegenüber dem MKS nicht gedreht sein. Wird z. B. über den Befehl ROT oder über den Schwenkzyklus CYCLE800 das BKS gedreht, arbeitet die Tangentialsteuerung nicht mehr korrekt. | | | |

### 3.16.7.2 Zwischensatzerzeugung einschalten (TLIFT)

Überschreitet die Tangentenänderung der Folgeachse an einer Stelle der programmierten Bahn der Leitachsen den im Maschinendatum MD37400 $MA_EPS_TLIFT_TANG_STEP parametrierten Grenzwert, erfolgt die weitere Bahnplanung abhängig vom eingestellten Eckenverhalten. Ohne Verwendung der vordefinierten Prozedur TLIFT(...) wird entsprechend des im Zusammenhang mit TANG(...) (Seite 958) und TANGON(...) (Seite 960) programmierten Überschleifenverhaltens verfahren.

#### Zwischensatzerzeugung einschalten

Mit Programmierung von TLIFT(...) im Anschluss an TANG(...) wird vom Vorlauf beim Erkennen einer Ecke an dieser Stelle der Bahn ein von der Steuerung automatisch generierter Zwischensatz eingefügt.

Bei der Abarbeitung des Programms werden dann bei Erreichen des Zwischensatzes die Leitachsen angehalten. Im Zwischensatz wird die Folgeachse mit maximaler Achsdynamik in Richtung der Bahntangente des nachfolgenden Satzes gedreht. Danach werden die Leitachsen weiter auf der programmierten Bahn verfahren.

#### Zwischensatzerzeugung ausschalten

Zum Ausschalten der Zwischensatzerzeugung muss die Tangentialkopplung mittels TANG(...), aber ohne nachfolgendes Einschalten der Zwischensatzerzeugung mittels TLIFT(...), erneut definiert werden.

### Syntax

```
TLIFT(<Folgeachse>)
```

### Bedeutung

| | | |
|---|---|---|
| `TLIFT(...)`: | Einschalten der Eckenerkennung mit Zwischensatzberechnung | |
| `<Folgeachse>`: | Achsename der Folgeachse (Rundachse) | |
| | Datentyp: | AXIS |
| | Wertebereich: | Kanalachsnamen |

### Drehgeschwindigkeit der Folgeachse

#### Bahnachse

Wurde die Folgeachse vor der Aktivierung der Tangentialkopplung schon als Bahnachse verfahren, erfolgt die Drehbewegung im Zwischensatz als Bahnachse.

Durch die Vorgabe des Bezugsradius mit `FGREF[<Achse>]=0.001`, erfolgt die Drehbewegung mit der parametrierten maximalen Achsgeschwindigkeit:

MD32000 $MA_MAX_AX_VELO[<Folgeachse>]

#### Positionierachse

Wurde die Folgeachse vor der Aktivierung der Tangentialkopplung noch nicht als Bahnachse verfahren, erfolgt die Drehung im Zwischensatz als Positionierachse.

Die Drehbewegung erfolgt dabei mit der parametrierten Positionierachsgeschwindigkeit:

MD32060 $MA_POS_AX_VELO[<Folgeachse>]

## 3.16.7.3 Kopplung einschalten (TANGON)

Über die vordefinierte Prozedur TANGON(...) wird eine zuvor mit TANG(...) (Seite 958) definierte Tangentialkopplung eingeschaltet. Die Folgeachse wird dann beim nachfolgenden Verfahren der Leitachsen kontinuierlich zur Bahntangente ausgerichtet.

### Winkel der Folgeachse

Der Winkel, den die Folgeachse in Bezug zur Bahntangente einnimmt, ist abhängig vom in TANG(...) vorgegebenen Übersetzungsverhältnis, dem im Maschinendatum MD37402 $MA_TANG_OFFSET parametrierten Offsetwinkel und dem dazu additv wirkenden bei TANGON(...) vorgegebenen Offsetwinkel.

### Optimierung "P"

Wurde bei der Definition der Tangentialkopplung (TANG(...)) als Optimierungsparameter der Wert "P" angegeben, muss beim Einschalten der Kopplung der Parameter "Überschleifweg" und optional der Parameter "Winkeltoleranz" angegeben werden.

Wird als Winkeltoleranz der Wert 0 vorgegeben, wirkt nur der Parameter "Überschleifweg".

Wird als Winkeltoleranz ein Wert größer 0 vorgegeben, ergibt sich der wirksame Überschleifweg aus dem Minimum des parametrierten Überschleifwegs und des Überschleifwegs aufgrund der parametrierten Winkeltoleranz.

Ist die Dynamik der Folgeachse nicht ausreichend, um den parametrierten Bedingungen zu folgen, wird die Bahngeschwindigkeit der Leitachsen entsprechend reduziert.

## Syntax

```
TANGON(<Folgeachse>, <Offetwinkel>, <Überschleifweg>,
<Winkeltoleranz>)
```

## Bedeutung

| | | |
|---|---|---|
| `TANGON(...)`: | Einschalten einer Tangentialkopplung | |
| `<Folgeachse>`: | Achsename der Folgeachse (Rundachse) | |
| | Datentyp: | AXIS |
| | Wertebereich: | Kanalachsnamen |
| `<Offsetwinkel>`: | Offsetwinkel der Folgeachse zur Bahntangente<br>Bezugspunkt ist der Nullpunkt der Rundachse. | |
| | Datentyp: | REAL |
| `<Überschleifweg>`: | Maximal zulässiger Überschleifweg<br>Würde der Überschleifweg aufgrund der Dynamikbedingungen größer, wird die Bahngeschwindigkeit der Leitachsen vermindert. | |
| | Datentyp: | REAL |
| `<Winkeltoleranz>`: | Maximal zulässige Toleranz bezüglich des vorgegebenen Winkels zwischen der Folgeachs-Nullstellung und der Bahntangente | |
| | Datentyp: | REAL |

### 3.16.7.4 Kopplung ausschalten (TANGOF)

Über die vordefinierte Prozedur TANGOF(...) wird eine mit TANG(...) (Seite 958) definierte und mit TANGON(...) (Seite 960) eingeschaltete Tangentialkopplung ausgeschaltet. Die Folgeachse wird dann nicht mehr auf die Bahntangente der Leitachse ausgerichtet. Die Kopplung der Folgeachse an die Leitachsen bleibt aber auch nach dem Ausschalten weiterhin bestehen, was z.B. folgende Funktionen verhindert:

- Ebenenwechsel
- Geometrieachsumschaltung
- Definition einer neuen Tangentialkopplung für die Folgeachse

Das vollständige Aufheben der Bindung der Folgeache an die Leitachsen erfolgt erst nach dem Löschen der Kopplung durch TANGDEL(...) (Seite 962).

#### Programmierung

```
TANGOF(<Folgeachse>)
```

#### Bedeutung

| | | |
|---|---|---|
| `TANGOF(...)`: | Ausschalten einer Tangentialkopplung | |
| `<Folgeachse>`: | Achsename der Folgeachse (Rundachse) | |
| | Datentyp: | AXIS |
| | Wertebereich: | Kanalachsnamen |

### 3.16.7.5 Kopplung löschen (TANGDEL)

Eine mit TANG(...) (Seite 958) definierte Tangentialkopplung bleibt auch nach dem Ausschalten der Tangentialkopplung mit TANGOF(...) (Seite 962) hinaus bestehen. Die bestehende Tangentialkopplung verhindert dann z.B. weiterhin folgende Funktionen:

- Ebenenwechsel
- Geometrieachsumschaltung
- Definition einer neuen Tangentialkopplung für die Folgeachse

Mit der vordefinierten Prozedur TANGDEL(...) wird nach dem Ausschalten der Tangentialkopplung mit TANGOF(...) die bestehende Tangentialkopplung gelöscht.

#### Sytax

```
TANGDEL(<Folgeachse>)
```

#### Bedeutung

| | | |
|---|---|---|
| `TANGDEL(...)`: | Löschen einer mit TANG() definierten Tangentialkopplung | |
| | Wirksamkeit: | Satzweise |

| | | |
|---|---|---|
| <Folgeachse>: | Achsename der Folgeachse, deren Tangentialkopplung gelöscht werden soll | |
| | Datentyp: | AXIS |
| | Wertebereich: | Kanalachsnamen |

## Beispiele

### Leitachwechsel

Bevor für die Folgeachse eine neue Tangentialkopplung mit einer anderen Leitachse definiert werden kann, muss die bestehende Tangentialkopplung zuerst gelöscht werden.

```
Programmcode                    Kommentar
N10 TANG(A, X, Y, 1)            ; Tangentialkopplung für Folgeachse A definieren: A zu
                                X und Y
N20 TANGON(A)                   ; Tangentialkopplung für Folgeachse A einschalten
N30 X10 Y20
...
N80 TANGOF(A)                   ; Tangentialkopplung für Folgeachse A ausschalten
N90 TANGDEL(A)                  ; Tangentialkopplung für Folgeachse A löschen
...
N120 TANG(A, X, Z)              ; Neue Tangentialkopplung für Folgeachse A definieren
N130 TANGON(A)                  ; Neue Tangentialkopplung für Folgeachse A einschalten
...
```

### Geometrieachsumschaltung

Bevor eine Geometrieachsumschaltung für eine bestehende Kopplung vorgenommen werden kann, muss die Kopplung zuerst gelöscht werden.

```
Programmcode                    Kommentar
N10 GEOAX(2, Y1)                ; 2. Geometrieachse = Maschinenachse Y1
N20 TANG(A, X, Y)               ; Tangentialkopplung für Folgeachse A definieren
N30 TANGON(A, 90)               ; Tangentialkopplung für Folgeachse A einschalten
N40 G2 F8000 X0 Y0 I0 J50       ; Bewegungssatz
N50 TANGOF(A)                   ; Tangentialkopplung für Folgeachse A ausschalten
N60 TANGDEL(A)                  ; Tangentialkopplung für Folgeachse A löschen
N70 GEOAX(2, Y2)                ; 2. Geometrieachse = Maschinenachse Y2
N80 TANG(A, X, Y)               ; Neue Tangentialkopplung für Folgeachse A definieren
N90 TANGON(A, 90)               ; Neue Tangentialkopplung für Folgeachse A einschal-
                                ten
...
```

## 3.16.8 Master/Slave-Kopplung (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)

Die "Master/Slave-Kopplung" ermöglicht:

- das Einkoppeln der Slave-Achsen auf die Master-Achse im Stillstand der beteiligten Achsen.
- das Koppeln und Trennen von **drehenden**, drehzahlgesteuerten Spindeln.
- die dynamische Projektierung.

**Hinweis**

**Positionierbetrieb**

Bei Achsen und Spindeln im Positionierbetrieb wird die Kopplung nur im Stillstand geschlossen und getrennt.

### Syntax

```
MASLON(<Slave_1>,<Slave_2>,...)
MASLOF(<Slave_1>,<Slave_2>,...)
MASLOFS(<Slave_1>,<Slave_2>,...)
```

Dynamische Projektierung:

```
MASLDEF(<Slave_1>,<Slave_2>,...,<Master>)
MASLDEL(<Slave_1>,<Slave_2>,...)
```

### Bedeutung

| | | |
|---|---|---|
| MASLON: | Eine temporäre Master/Slave-Kopplung einschalten | |
| | <Slave_x>,...: | Slave-Achse 1 ... n |
| MASLOF: | Eine aktive Master/Slave-Kopplung trennen | |
| | <Slave_1>,...: | Slave-Achse 1 ... n |
| MASLOFS: | Master/Slave-Kopplung trennen und Slave-Spindeln automatisch abbremsen (siehe Hinweis "Koppelverhalten bei Spindeln im Drehzahlsteuerbetrieb"!) | |
| | <Slave_1>,...: | Slave-Achse 1 ... n |
| MASLDEF: | Master/Slave-Verband aus dem Teileprogramm heraus anlegen/ändern | |
| | <Slave_1>,...: | Slave-Achse 1 ... n |
| | <Master>: | Master-Achse |
| MASLDEL: | Master/Slave-Kopplung trennen und Definition des Verbandes löschen | |
| | <Slave_1>,...: | Slave-Achse 1 ... n |
| | **Hinweis:**<br>Die in den Maschinendaten projektierten Master-/Slave-Definitionen bleiben erhalten. | |

**Hinweis**

**Koppelverhalten bei Spindeln im Drehzahlsteuerbetrieb**

Bei Spindeln im Drehzahlsteuerbetrieb wird das Koppelverhalten von MASLON, MASLOF, MASLOFS und MASLDEL explizit über das folgende Maschinendatum festgelegt:

MD37263 $MA_MS_SPIND_COUPLING_MODE

In der Standardeinstellung mit MD37263 = 0 erfolgt das Einkoppeln und Trennen der Slave-Achsen ausschließlich im Stillstand der beteiligten Achsen. MASLOFS entspricht dem MASLOF.

Bei MD37263 = 1 wird die Koppelanweisung unmittelbar, und damit auch in der Bewegung ausgeführt. Die Kopplung wird bei MASLON sofort geschlossen und bei MASLOFS oder MASLOF sofort getrennt. Die zu diesem Zeitpunkt drehenden Slave-Spindeln behalten bei MASLOF ihre Drehzahlen bis zur erneuten Drehzahlprogrammierung bei. Bei MASLOFS werden sie dagegen automatisch abgebremst.

**Hinweis**

Bei MASLOF/MASLOFS entfällt der implizite Vorlaufstopp. Bedingt durch den fehlenden Vorlaufstopp liefern die $P-Systemvariablen für die Slave-Achsen bis zum Zeitpunkt erneuter Programmierung keine aktualisierten Werte.

**Hinweis**

Für die Slave-Achse kann der Istwert durch PRESETON auf den gleichen Wert der Master-Achse synchronisiert werden. Dazu muss die dauerhafte Master/Slave-Kopplung kurzfristig ausgeschaltet werden, um den Istwert der nicht referenzierten Slave-Achse mit POWER ON auf den Wert der Master-Achse zu setzen. Danach wird die dauerhafte Kopplung wieder hergestellt.

Die dauerhafte Master/Slave-Kopplung wird mit folgender MD-Einstellung aktiviert:
MD37262 $MA_MS_COUPLING_ALWAYS_ACTIVE = 1
Sie hat keine Auswirkung auf die Sprachbefehle der temporären Kopplung.

## Beispiele

### Beispiel 1: Istwertsetzen bei den Slave-Achse einer Master/Slave-Kopplung

Bei einer permanenten Master/Slave-Kopplung wird mit PRESETON der Istwert der Slave-Achse auf den Wert der Master-Achse gesetzt.

| Programmcode | Kommentar |
|---|---|
| $MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=0 | ; Permanente Kopplung der Slave-Achse ausschalten |
| NEWCONF | ; Maschinendatenänderung aktivieren |
| STOPRE | |
| MASLOF(Y1) | ; Temporäre Kopplung ausschalten |
| PRESETON(AX2,$VA_IM(M_AX)) | ; Istwert der Slave-Achse = Istwert der Master-Achse |

| Programmcode | Kommentar |
|---|---|
| `$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=1` | `; Permanente Kopplung der Slave-Achse einschalten` |
| `NEWCONF` | `; Maschinendatenänderung aktivieren` |

### Beispiel 2: Dynamische Projektierung einer Master/Slave-Kopplung

Damit die Kopplung nach der Achscontainer-Drehung mit einer anderen Spindel geschlossen werden kann, muss vorher die alte Kopplung getrennt, die Projektierung gelöscht und die neue Kopplung projektiert werden.

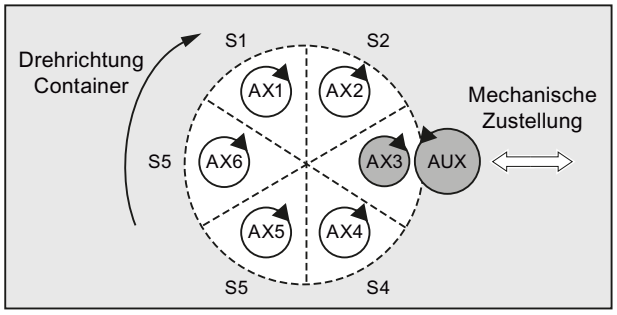



Bild 3-14 Vor der Achscontainer-Drehung

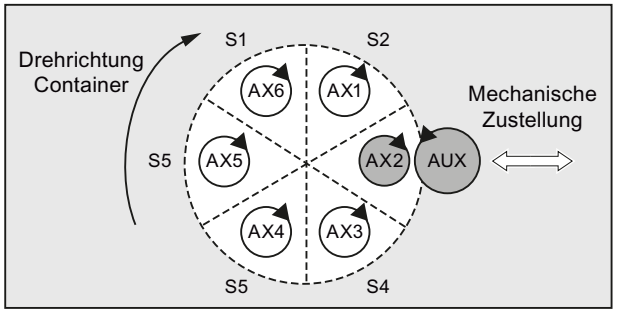



Bild 3-15 Nach der Achscontainer-Drehung um einen Slot

| Programmcode | Kommentar |
|---|---|
| `MASLDEF(AUX,S3)` | `; AUX: Slave, S3: Master = AX3` |
| `MASLON(AUX)` | `; Kopplung ein` |
| `M3=3 S3=4000` | `; Master drehen` |
| `MASLDEL(AUX)` | `; Kopplung trennen und löschen` |
| `AXCTSWE(CT1)` | `; Achscontainer-Drehung freigeben` |
| `MASLDEF(AUX,S3)` | `; AUX: Slave, S3: Master = AX2` |

## Weitere Informationen

- Funktionshandbuch Technologien
- Funktionshandbuch Basisfunktionen

# 3.17 Synchronaktionen

## 3.17.1 Definition einer Synchronaktion

Eine Synchronaktion wird in einem Satz eines Teileprogramms definiert. Innerhalb dieses Satzes dürfen keine weiteren Befehle programmiert werden, die nicht Bestandteil der Synchronaktion sind.

Eine Synchronaktion besteht aus folgenden Komponenten:

| Gültigkeit, Ident-Nr. (optional) | Bedingungsteil (optional) | | | Aktionsteil bei erfüllter Bedingung | | | Aktionsteil bei nicht erfüllter Bedingung (optional) | | |
|---|---|---|---|---|---|---|---|---|---|
| | Häufigkeit | G-Befehl (optional) | Bedingung | Schlüsselwort | G-Befehl (optional) | Aktionen | Schlüsselwort | G-Befehl (optional) | Aktionen |
| --- [1]<br>ID=<Nr><br>IDS=<Nr> | --- [1]<br>WHENEVER<br>FROM<br>WHEN<br>EVERY | G... | Logischer Ausdruck | DO | G... | Aktion 1<br>...<br>Aktion n | ELSE | G... | Aktion 1<br>...<br>Aktion n |

[1] nicht programmiert

### Syntax

```
DO <Aktion 1> ... <Aktion n>
<Häufigkeit> [<G-Funktion>] <Bedingung> DO <Aktion 1> ... <Aktion n>
ID=<Nr> <Häufigkeit> [<G-Funktion>] <Bedingung> DO <Aktion 1> ...
<Aktion n>
IDS=<Nr> <Häufigkeit> [<G-Funktion>] <Bedingung> DO <Aktion 1> ...
<Aktion n>
IDS=<Nr> <Häufigkeit> [<G-Funktion>] <Bedingung> DO <Aktion 1...n>
ELSE <Aktion 1...n>
```

### Weitere Informationen

Funktionshandbuch Synchronaktionen

# 3.18 Pendeln

## 3.18.1 Asynchrones Pendeln (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB)

Eine Pendelachse fährt zwischen den zwei Umkehrpunkten 1 und 2 mit gegebenem Vorschub hin und her, bis die Pendelbewegung abgeschaltet wird.

Andere Achsen können während der Pendelbewegung beliebig interpoliert werden. Über eine Bahnbewegung oder mit einer Positionierachse kann eine kontinuierliche Zustellung erreicht werden. Dabei besteht jedoch **kein Zusammenhang** zwischen der Pendel- und der Zustellbewegung.

### Eigenschaften des asynchronen Pendelns

- Das asynchrone Pendeln ist achsspezifisch über Satzgrenzen hinweg wirksam.
- Über das Teileprogramm ist ein satzsynchrones Einschalten der Pendelbewegung gewährleistet.
- Eine gemeinsame Interpolation von mehreren Achsen und eine Überlagerung von Pendelstrecken sind nicht möglich.

### Programmierung

Über die folgenden Befehle ist ein der Abarbeitung des NC-Programms entsprechendes Einschalten und Beeinflussen des asynchronen Pendelns vom Teileprogramm her möglich.

Die programmierten Werte werden satzsynchron im Hauptlauf in die entsprechenden Settingdaten eingetragen und bleiben bis zur nächsten Änderung wirksam.

### Syntax

```
OSP1[<Achse>]=<Wert> OSP2[<Achse>]=<Wert>
OST1[<Achse>]=<Wert> OST2[<Achse>]=<Wert>
FA[<Achse>]=<Wert>
OSCTRL[<Achse>]=(<Setzoption>,<Rücksetzoption>)
OSNSC[<Achse>]=<Wert>
OSE[<Achse>]=<Wert>
OSB[<Achse>]=<Wert>
OS[<Achse>]=1
OS[<Achse>]=0
```

### Bedeutung

| | | | |
|---|---|---|---|
| `<Achse>`: | Name der Pendelachse | | |
| `OS`: | Pendeln ein-/ausschalten | | |
| | Wert: | 1 | Pendeln **ein**schalten |
| | | 0 | Pendeln **aus**schalten |
| `OSP1`: | Position von Umkehrpunkt 1 festlegen | | |

| | | | |
|---|---|---|---|
| OSP2: | Position von Umkehrpunkt 2 festlegen<br>**Hinweis:**<br>Falls ein inkrementelles Verfahren aktiv ist, so wird die Position inkrementell zur letzten im NC-Programm programmierten entsprechenden Umkehrposition berechnet. | | |
| OST1: | Haltezeit im Umkehrpunkt 1 in [s] festlegen | | |
| OST2: | Haltezeit im Umkehrpunkt 2 in [s] festlegen | | |
| | <Wert>: | -2 | Interpolation wird ohne Warten auf Genauhalt fortgesetzt |
| | | -1 | Warten auf Genauhalt grob |
| | | 0 | Warten auf Genauhalt fein |
| | | >0 | Warten auf Genauhalt fein und anschließend Abwarten der angegebenen Haltezeit<br>**Hinweis:**<br>Die Einheit für die Haltezeit ist identisch mit der über G4 programmierten Haltezeit. |
| FA: | Vorschubgeschwindigkeit festlegen<br>Als Vorschubgeschwindigkeit gilt die definierte Vorschubgeschwindigkeit der Positionierachse. Ist keine Vorschubgeschwindigkeit definiert, gilt der im Maschinendatum hinterlegte Wert. | | |
| OSCTRL: | Setz- und Rücksetzoptionen angeben<br>Die Optionswerte 0 - 3 verschlüsseln das Verhalten an den Umkehrpunkten beim Ausschalten. Es kann eine der Varianten 0 - 3 ausgewählt werden. Die übrigen Einstellungen sind nach Bedarf kombinierbar mit der gewählten Variante. Mehrere Optionen werden durch Pluszeichen (+) aneinandergefügt. | | |
| | <Wert>: | 0 | Beim Abschalten der Pendelbewegung im nächsten Umkehrpunkt stoppen (Voreinstellung)<br>**Hinweis:**<br>Nur durch Rücksetzen der Werte 1 und 2 möglich. |
| | | 1 | Beim Abschalten der Pendelbewegung in Umkehrpunkt 1 stoppen |
| | | 2 | Beim Abschalten der Pendelbewegung in Umkehrpunkt 2 stoppen |
| | | 3 | Beim Abschalten der Pendelbewegung keinen Umkehrpunkt anfahren, falls keine Ausfeuerungshübe programmiert sind |
| | | 4 | Nach dem Ausfeuern Endposition anfahren |
| | | 8 | Wird die Pendelbewegung durch Restweglöschen abgebrochen, sollen anschließend Ausfeuerungshübe abgearbeitet und ggf. die Endposition angefahren werden. |
| | | 16 | Wird die Pendelbewegung durch Restweglöschen abgebrochen, soll wie beim Abschalten die entsprechende Umkehrposition angefahren werden. |
| | | 32 | Geänderter Vorschub ist erst ab dem nächsten Umkehrpunkt aktiv |
| | | 64 | FA gleich 0, FA = 0: Wegüberlagerung ist aktiv<br>FA ungleich 0, FA <> 0: Geschwindigkeitsüberlagerung ist aktiv |
| | | 128 | Bei Rundachse DC (kürzester Weg) |
| | | 256 | Ausfeuerhub wird als Doppelhub ausgeführt.(Standard) 1=Ausfeuerhub wird als Einzelhub ausgeführt. |
| | | 512 | Zuerst Startposition anfahren |
| OSNSC: | Anzahl der Ausfeuerungshübe festlegen | | |

| | |
|---|---|
| OSE: | Endposition (im WKS) festlegen, die nach Ausschalten des Pendelns angefahren werden soll<br>**Hinweis:**<br>Bei Programmierung von "OSE" wird für "OSCTRL" implizit Option 4 wirksam. |
| OSB: | Startposition (im WKS) festlegen, die vor Einschalten des Pendelns angefahren werden soll<br>Die Startposition wird vor Umkehrpunkt 1 angefahren. Stimmt die Startposition mit der Umkehrposition 1 überein, so wird als nächstes die Umkehrposition 2 angefahren. Beim Erreichen der Startposition wirkt keine Haltezeit, auch wenn die Startposition mit der Umkehrposition 1 übereinstimmt, stattdessen wird auf Genauhalt fein gewartet. Eine eingestellte Genauhaltbedingung wird eingehalten.<br>**Hinweis:**<br>Damit die Startposition angefahren wird, muss im Settingdatum SD43770 $SA_OSCILL_CTRL_MASK Bit 9 gesetzt sein. |

## Beispiele

### Beispiel 1: Pendelachse soll zwischen zwei Umkehrpunkten pendeln

Die Pendelachse Z soll zwischen Position 10 und 100 pendeln. Umkehrpunkt 1 soll mit Genauhalt fein, Umkehrpunkt 2 mit Genauhalt grob angefahren werden. Der Vorschub für die Pendelachse soll 250 betragen. Am Ende der Bearbeitung sollen 3 Ausfeuerungshübe erfolgen und die Pendelachse soll die Endposition 200 ansteuern. Der Vorschub für die Zustellachse soll 1 betragen, das Ende der Zustellung in X-Richtung soll bei Position 15 erreicht sein.

| **Programmcode** | **Kommentar** |
|---|---|
| `WAITP(X,Y,Z)` | `; Ausgangsstellung.` |
| `G0 X100 Y100 Z100` | `; Umschalten in Positionierachsbetrieb.` |
| `WAITP(X,Z)` | |
| `OSP1[Z]=10 OSP2[Z]=100` | `; Umkehrpunkt 1, Umkehrpunkt 2.` |
| `OSE[Z]=200` | `; Endposition.` |
| `OST1[Z]=0 OST2[Z]=-1` | `; Haltezeit an U1: Genauhalt fein` |
| | `; Haltezeit an U2: Genauhalt grob` |
| `FA[Z]=250 FA[X]=1` | `; Vorschub Pendelachse, Vorschub Zustellachse.` |
| `OSCTRL[Z]=(4,0)` | `; Setzoptionen.` |
| `OSNSC[Z]=3` | `; 3 Ausfeuerhübe.` |
| `OS[Z]=1` | `; Pendeln starten.` |
| `WHEN $A_IN[3]==TRUE DO DELDTG(X)` | `; Restweglöschen.` |
| `POS[X]=15` | `; Ausgangsstellung X-Achse` |
| `POS[X]=50` | `; Endstellung X-Achse.` |
| `OS[Z]=0` | `; Pendeln stoppen.` |
| `M30` | |

**Hinweis**

Der Befehlsfolge "OSP1[Z]=..." bis "OSNCS[Z]=..." kann auch in einem Satz programmiert werden.

### Beispiel 2: Pendeln mit Online-Änderung der Umkehrposition

Die für das asynchrone Pendeln erforderlichen Settingdaten können im Teileprogramm eingestellt werden.

Werden im Teileprogramm die Settingdaten direkt beschrieben, so wird die Änderung schon zum Vorlaufzeitpunkt wirksam. Synchrones Verhalten kann über einen Vorlaufstopp (STOPRE) erreicht werden.

| Programmcode | Kommentar |
|---|---|
| `$SA_OSCILL_REVERSE_POS1[Z]=-10` | |
| `$SA_OSCILL_REVERSE_POS2[Z]=10` | |
| `G0 X0 Z0` | |
| `WAITP(Z)` | |
| `ID=1 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS1[Z] DO $AA_OVR[X]=0` | `; Wenn der Istwert der Pendelachse den Umkehrpunkt überschritten hat, wird die Zustellachse angehalten.` |
| `ID=2 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[X]=0` | |
| `OS[Z]=1 FA[X]=1000 POS[X]=40` | `; Pendeln einschalten.` |
| `OS[Z]=0` | `; Pendeln ausschalten.` |
| `M30` | |

## Weitere Informationen

### Pendelachse

Für die Pendelachse gilt:

- Jede Achse kann als Pendelachse benutzt werden.
- Gleichzeitig können mehrere Pendelachsen aktiv sein (maximal: Anzahl der Positionierachsen).
- Für die Pendelachse ist immer - unabhängig vom im Programm aktuell gültigen G-Befehl - Linearinterpolation G1 aktiv.

Die Pendelachse kann:

- Eingangsachse für die dynamische Transformation sein
- Führungsachse bei Gantry- und Mitschleppachsen sein
- verfahren werden:
  - ohne Ruckbegrenzung "BRISK"
    oder
  - mit Ruckbegrenzung "SOFT"
    oder
  - mit geknickter Beschleunigungskennlinie (wie Positionierachsen)

### Pendelumkehrpunkte

Bei der Festlegung der Pendelpositionen sind die aktuellen Verschiebungen zu beachten:

- Absolute Angabe
  "OSP1[Z]=<Wert>"
  Position Umkehrpunkt = Summe der Verschiebungen + programmierter Wert
- Relative Angabe
  "OSP1[Z]=IC(<Wert>)"
  Position Umkehrpunkt = Umkehrpunkt 1 + programmierter Wert

Beispiel:

**Programmcode**

```
N10 OSP1[Z]=100 OSP2[Z]=110
...
N40 OSP1[Z]=IC(3)
```

### WAITP

Soll mit einer Geometrieachse gependelt werden, so muss diese mit "WAITP" zum Pendeln freigegeben werden.

Nach beendetem Pendeln wird mit "WAITP" die Pendelachse wieder als Positionierachse eingetragen und kann wieder normal verwendet werden.

### Pendeln mit Bewegungssynchronaktionen und Haltezeiten

Nach Ablauf der eingestellten Haltezeiten findet beim Pendeln der interne Satzwechsel statt (sichtbar an den neuen Restwegen der Achsen). Beim Satzwechsel wird die Ausschaltfunktion überprüft. Dabei wird nach der eingestellten Steuereinstellung für den Bewegungsablauf (`OSCTRL`) die Ausschaltfunktion festgelegt. *Dieses Zeitverhalten ist durch den Vorschuboverride beeinflussbar.*

Unter Umständen wird danach noch ein Pendelhub ausgeführt, bevor die Ausfeuerungshübe gestartet oder die Endposition angefahren wird. *Es entsteht dabei der Eindruck, es verändert sich das Ausschaltverhalten. Dies ist aber nicht der Fall.*

## 3.18.2 Über Synchronaktionen gesteuertes Pendeln (OSCILL)

Bei dieser Art des Pendelns ist nur an den Umkehrpunkten bzw. innerhalb definierter Umkehrbereiche eine Zustellbewegung zugelassen.

Je nach Anforderung kann die Pendelbewegung während der Zustellung

- fortgeführt oder
- angehalten werden, bis die Zustellung vollständig ausgeführt ist.

### Syntax

1. **Parameter für das Pendeln festlegen**
2. **Bewegungssynchronaktionen definieren**
3. **Achsen zuordnen, Zustellung festlegen**

### Bedeutung

| | |
|---|---|
| `OSP1[<Pendelachse>]=` | Position des Umkehrpunkts 1 |
| `OSP2[<Pendelachse>]=` | Position des Umkehrpunkts 2 |
| `OST1[<Pendelachse>]=` | Haltezeit in Umkehrpunkt 1 in Sekunden |
| `OST2[<Pendelachse>]=` | Haltezeit in Umkehrpunkt 2 in Sekunden |
| `FA[<Pendelachse>]=` | Vorschub der Pendelachse |
| `OSCTRL[<Pendelachse>]=` | Setz- bzw. Rücksetzoptionen |
| `OSNSC[<Pendelachse>]=` | Anzahl der Ausfeuerungshübe |
| `OSE[<Pendelachse>]=` | Endposition |
| `WAITP(<Pendelachse>)` | Achse für das Pendeln freigeben |

#### Achszuordnung, Zustellung

```
OSCILL[<Pendelachse>]=(<Zustellachse 1>,<Zustellachse 2>,<Zustellachse 3>)
POSP[<Zustellachse>]=(<Endpos>,<Teillänge>,<Modus>)
```

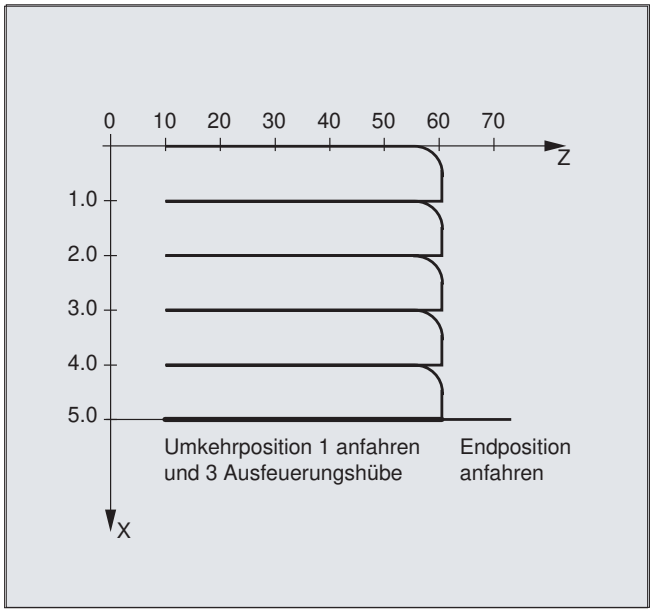
| | |
|---|---|
| `OSCILL`: | Zustellachse(n) der Pendelachse zuordnen |
| `POSP`: | Gesamt- und Teilzustellungen festlegen (siehe Kapitel Datei- und Programmverwaltung) |
| `Endpos`: | Endposition für die Zustellachse, nachdem alle Teilzustellungen abgefahren sind. |
| `Teillänge`: | Größe der Teilzustellung am Umkehrpunkt/Umkehrbereich |
| `Modus`: | Aufteilung der Gesamtzustellung in Teilzustellungen<br>= zwei gleich große Restschritte (Voreinstellung);<br>= alle Teilzustellungen gleich groß |

#### Bewegungssynchronaktionen

| | |
|---|---|
| `WHEN… … DO` | wenn…, dann… |
| `WHENEVER … DO` | immer wenn…, dann… |

## Beispiel

Im Umkehrpunkt 1 soll keine Zustellung erfolgen. Beim Umkehrpunkt 2 soll die Zustellung bereits im Abstand ii2 vor dem Umkehrpunkt 2 erfolgen und die Pendelachse im Umkehrpunkt nicht auf das Beenden der Teilzustellung warten. Die Achse Z ist Pendelachse und die Achse X Zustellachse.




### 1. Parameter für das Pendeln

| Programmcode | Kommentar |
|---|---|
| `DEF INT ii2` | `; Variable für Umkehrbereich 2 definieren` |
| `OSP1[Z]=10 OSP2[Z]=60` | `; Umkehrpunkt 1 und 2 definieren` |
| `OST1[Z]=0 OST2[Z]=0` | `; Umkehrpunkt 1: Genauhalt fein`<br>`Umkehrpunkt 2: Genauhalt fein` |
| `FA[Z]=150 FA[X]=0.5` | `; Vorschub Pendelachse Z, Vorschub Zustellachse X` |
| `OSCTRL[Z]=(2+8+16,1)` | `; Pendelbewegung abschalten im Umkehrpunkt 2; nach RWL Ausfeuern und Endposition anfahren; nach RWL entsprechende Umkehrposition anfahren` |
| `OSNC[Z]=3` | `; Ausfeuerungshübe` |
| `OSE[Z]=70` | `; Endposition = 70` |
| `ii2=2` | `; Umkehrbereich einstellen` |
| `WAITP(Z)` | `; Erlaube Pendeln für Z-Achse` |

### 2. Bewegungssynchronaktion

| Programmcode | Kommentar |
|---|---|
| `WHENEVER $AA_IM[Z]<$SA_OSCILL_REVERSE_POS2[Z]`<br>`DO ->`<br>`  $AA_OVR[X]=0 $AC_MARKER[0]=0` | `; Immer wenn die aktuelle Position der Pendelachse Z im MKS kleiner als der Beginn des Umkehrbereichs 2 ist, dann setze den axialen Override der Zustellachse X auf 0% und den Merker mit dem Index 0 auf den Wert 0.` |

| Programmcode | Kommentar |
|---|---|
| WHENEVER $AA_IM[Z]>=$SA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[Z]=0 | ; Immer wenn die aktuelle Position der Pendelachse Z im MKS größer gleich der Umkehrposition 2 ist, dann setze den axialen Override der Pendelachse Z auf 0%. |
| WHENEVER $AA_DTEPW[X]==0 DO $AC_MARKER[0]=1 | ; Immer wenn der Restweg der Teilzustellung gleich 0 ist, dann setze den Merker mit dem Index 0 auf den Wert 1. |
| WHENEVER $AC_MARKER[0]==1 DO $AA_OVR[X]=0 $AA_OVR[Z]=100 | ; Immer wenn der Merker mit dem Index 0 gleich 1 ist, dann setze den axialen Override der Zustellachse X auf 0%. Damit wird eine zu frühe Zustellung verhindert (Pendelachse Z hat den Umkehrbereich 2 noch nicht wieder verlassen, die Zustellachse X ist aber bereit für eine erneute Zustellung). Setze den axialen Override der Pendelachse Z von 0% (Aktion der 2. Synchronaktion) zum Verfahren wieder auf 100%. |

-> muss in einem Satz programmiert werden

### 3. Pendeln starten

| Programmcode | Kommentar |
|---|---|
| OSCILL[Z]=(X) POSP[X]=(5,1,1) | ; Starten der Achsen<br>Der Pendelachse Z wird die Achse X als Zustellachse ;zugewiesen.<br>Die Achse X soll bis Endposition 5 in Schritten von 1 fahren. |
| M30 | ; Programmende |

## Weitere Informationen

1. **Pendelparameter festlegen**
   Vor dem Bewegungssatz, der die Zuordnung von Zustell- und Pendelachse sowie die Festlegung der Zustellung enthält, sind die Parameter für das Pendeln festzulegen (siehe "Asynchrones Pendeln").
2. **Bewegungssynchronaktionen festlegen**
   Über Synchronbedingungen erfolgt:
   **Zustellung unterdrücken**, bis sich die Pendelachse innerhalb eines Umkehrbereichs (ii1, ii2) oder an einem Umkehrpunkt (U1, U2) befindet.
   **Pendelbewegung** während der Zustellung im Umkehrpunkt **anhalten**.
   **Pendelbewegung** nach beendeter Teilzustellung wieder **starten**.
   **Start der nächsten Teilzustellung** festlegen.
3. **Pendeln- und Zustellachse zuordnen** sowie **Gesamt- und Teilzustellung** festlegen.

## Pendelparameter festlegen

### Zuordnung von Pendel- und Zustellachse: OSCILL

```
OSCILL[<Pendelachse>]=(<Zustellachse1>,<Zustellachse2>,<Zustellachse
3>)
```

Mit dem Befehl "OSCILL" erfolgen die Achszuordnungen und der Start der Pendelbewegung.

Maximal können einer Pendelachse 3 Zustellachsen zugewiesen werden.

**Hinweis**

Vor dem Start des Pendelns müssen die Synchronbedingungen für das Verhalten der Achsen festgelegt sein.

### Zustellungen festlegen: POSP

```
POSP[<Zustellachse>]=(<Endpos>,<Teillänge>,<Modus>)
```

Mit dem Befehl "POSP" werden der Steuerung mitgeteilt:

- Gesamtzustellung (über die Endposition)
- Die Größe der jeweiligen Teilzustellung am Umkehrpunkt bzw. im Umkehrbereich
- Das Teilzustellverhalten bei Erreichen der Endposition (über den Modus)

| | |
|---|---|
| Modus = 0 | Für die beiden letzten Teilzustellungen erfolgt eine Aufteilung des verbleibenden Weges bis zum Zielpunkt auf 2 gleich große Restschritte (Voreinstellung). |
| Modus = 1 | Alle Teilzustellungen sind gleich groß. Sie werden aus der Gesamtzustellung berechnet. |

## Bewegungssynchronaktionen festlegen

Die im folgenden ausgeführten Bewegungssynchronaktionen werden ganz allgemein zum Pendeln verwendet.

Sie finden Beispiellösungen für die Lösung von einzelnen Anforderungen, die Ihnen als Bausteine für die Erstellung von anwenderspezifischen Pendelbewegungen dienen.

**Hinweis**

Im Einzelfall können die Synchronbedingungen auch anders programmiert werden.

### Schlüsselwörter

| | |
|---|---|
| `WHEN … DO …` | wenn…, dann… |
| `WHENEVER … DO` | immer wenn…, dann… |

### Funktionen

Mit den im folgenden detailliert beschriebenen Sprachmitteln können Sie folgende Funktionen realisieren:

1. Zustellung im Umkehrpunkt.
2. Zustellung im Umkehrbereich.
3. Zustellung in beiden Umkehrpunkten.
4. Anhalten der Pendelbewegung im Umkehrpunkt.
5. Pendelbewegung wieder starten.
6. Teilzustellung nicht zu früh starten.

Für alle hier beispielhaft dargestellten Synchronaktionen gelten die Annahmen:

- Umkehrpunkt 1 < Umkehrpunkt 2
- Z = Pendelachse
- X = Zustellachse

**Hinweis**

Für nähere Erläuterungen, siehe Kapitel Bewegungssynchronaktionen.

## Pendeln- und Zustellachse zuordnen sowie Gesamt- und Teilzustellung festlegen

### Zustellung im Umkehrbereich

Die Zustellbewegung soll innerhalb eines Umkehrbereichs beginnen, bevor der Umkehrpunkt erreicht ist.

Diese Synchronaktionen verhindern die Zustellbewegung, bis sich die Pendelachse in einem Umkehrbereich befindet.

Unter den gegebenen Annahmen (siehe oben) ergeben sich folgende Anweisungen:

**Umkehrbereich 1:**

```
WHENEVER $AA_IM[Z]>$SA_OSCILL_RESERVE_POS1[Z]+ii1 DO $AA_OVR[X]=0
```

Immer wenn die aktuelle Position der Pendelachse im MKS größer als der Beginn des Umkehrbereichs 1 ist, dann setze den axialen Override der Zustellachse auf 0%.

**Umkehrbereich 2:**

```
WHENEVER $AA_IM[Z]<$SA_OSCILL_RESERVE_POS2[Z]+ii2 DO $AA_OVR[X]=0
```

Immer wenn die aktuelle Position der Pendelachse im KS kleiner als der Beginn des Umkehrbereichs 2 ist, dann setze den axialen Override der Zustellachse auf 0%.

### Zustellung im Umkehrpunkt

Solange die Pendelachse den Umkehrpunkt nicht erreicht hat, findet keine Bewegung der Zustellachse statt.

Unter den gegebenen Annahmen (siehe oben) ergeben sich folgende Anweisungen:

**Umkehrbereich 1:**

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Immer wenn die aktuelle Position der Pendelachse Z im MKS größer oder kleiner als die Position des Umkehrpunkts 1 ist, dann setze den axialen Override der Zustellachse X auf 0% und den axialen Override der Pendelachse Z auf 100%.

**Umkehrbereich 2:**

Für Umkehrpunkt 2:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Immer wenn die aktuelle Position der Pendelachse Zu im MKS größer oder kleiner als die Position des Umkehrpunkts 2 ist, dann setze den axialen Override der Zustellachse X auf 0% und den axialen Override der Pendelachse Z auf 100%.

### Anhalten der Pendelbewegung im Umkehrpunkt

Die Pendelachse wird am Umkehrpunkt angehalten, gleichzeitig beginnt die Zustellbewegung. Die Pendelbewegung wird fortgesetzt, wenn die Zustellbewegung vollständig ausgeführt ist.

Gleichzeitig kann diese Synchronaktion dazu benutzt werden, die Zustellbewegung zu starten, falls diese durch eine vorhergehende Synchronaktion, die noch wirksam ist, gestoppt wurde.

Unter den gegebenen Annahmen (siehe oben) ergeben sich folgende Anweisungen:

**Umkehrbereich 1:**

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Immer wenn die aktuelle Position der Pendelachse im MKS gleich der Umkehrposition 1 ist, dann setze den axialen Ovberrie der Pendelachse auf 0% und den axialen Override der Zustellachse auf 100%.

**Umkehrbereich 2:**

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Immer wenn die aktuelle Position der Pendelachse Zu im MKS gleich der Umkehrposition 2 ist, dann setze den axialen Override der PendelachseX auf 0% und den axialen Override der Zustellachse auf 100%.

### Online-Auswertung des Umkehrpunkts

Steht auf der rechten Seite des Vergleichs eine mit `$$` gekennzeichnete Hauptlaufvariable, so werden die beiden Variablen im IPO-Takt laufend ausgewertet und miteinander verglichen.

**Hinweis**

Mehr Informationen hierzu siehe Kapitel "Bewegungssynchronaktionen".

### Pendelbewegung wieder starten

Diese Synchronaktion wird dazu benutzt, die Bewegung der Pendelachse fortzusetzen, wenn die Teilzustellbewegung abgeschlossen ist.

Unter den gegebenen Annahmen (siehe oben) ergeben sich folgende Anweisungen:

```
WHENEVER $AA_DTEPW[X]==0 DO $AA_OVR[Z]=100
```

Immer wenn der Restweg für die Teilzustellung der Zustelalchse X im WKS gleich Null ist, dann setze den axialen Override der Pendelachse auf 100%.

### Nächste Teilzustellung

Nach erfolgter Zustellung muss ein zu frühes Starten der nächsten Teilzustellung verhindert werden.

Dazu wird ein kanalspezifischer Merker (`$AC_MARKER[Index]`) verwendet, der am Ende der Teilzustellung (Teilrestweg ≡ 0) gesetzt wird und beim Verlassen des Umkehrbereichs gelöscht wird. Dann wird mit einer Synchronaktion die nächste Zustellbewegung verhindert.

Unter den gegebenen Annahmen (siehe oben) ergeben sich z. B. für Umkehrpunkt 1 folgende Anweisungen:

#### 1. Marker setzen:

```
WHENEVER $AA_DTEPW[X]==0 DO $AC_MARKER[1]=1
```

Immer wenn der Restweg für die Teilzustellung der Zustellachse X im WKS gleich Null ist, dann setze den Merker mit Index 1 auf 1.

#### 2. Marker löschen

```
WHENEVER $AA_IM[Z]<> $SA_OSCILL_RESERVE_POS1[Z] DO $AC_MARKER[1]=0
```

Immer wenn die aktuelle Position der Pendelachse Z ium MKS größer oder kleiner als die Position des Umkehrpunkts 1 ist, dann setze den Merker 1 auf 0.

#### 3. Zustellung verhindern

```
WHENEVER $AC_MARKER[1]==1 DO $AA_OVR[X]=0
```

Immer wenn der Merker 1 gleich ist, dann setze den axialen Override der Zustellachse X auf 0%.

# 3.19 Stanzen und Nibbeln

## 3.19.1 Aktivierung/Deaktivierung

### 3.19.1.1 Stanzen und Nibbeln ein-/ausschalten (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC)

#### Stanzen bzw. Nibbeln aktivieren/deaktivieren

Mit `PON` und `SON` wird die Stanz- bzw. Nibbelfunktion aktiviert. `SPOF` beendet alle stanz- und nibbelspezifischen Funktionen. Die modal wirksamen Befehle `PON` und `SON` schließen sich gegenseitig aus, d. h. `PON` deaktiviert `SON` und umgekehrt.

#### Stanzen/Nibbeln mit Vorspann

Die Funktionen `SONS` und `PONS` schalten ebenfalls die Stanz- bzw. Nibbelfunktionen ein.

Im Gegensatz zu der bei `SON`/`PON` wirksamen Hubsteuerung auf Interpolationsebene erfolgt bei diesen Funktionen die signaltechnische Steuerung der Hubauslösung auf Servoebene. Hierdurch kann mit höheren Hubfrequenzen und damit höherer Stanzleistung gearbeitet werden.

Während der Signalauswertung im Vorspann sind alle Funktionen verriegelt, die zur Positionsänderung der Nibbel- oder Stanzachsen führen (z. B. manuelles Verfaren per Handrad, Änderungen von Frames über PLC, Messfunktionen).

#### Stanzen mit Verzögerung

`PDELAYON` bewirkt eine verzögerte Ausgabe des Stanzhubs. Der modal wirksame Befehl hat vorbereitende Funktion und steht damit in der Regel vor `PON`. Nach `PDELAYOF` wird normal weitergestanzt.

---

**Hinweis**

Die Verzögerungszeit wird eingestellt im Settingdatum SD42400 $SC_PUNCH_DWELLTIME.

---

#### Wegabhängige Beschleunigung

Mit `PUNCHACC` kann eine Beschleunigungskennlinie festgelegt werden, die je nach Lochabstand unterschiedliche Beschleunigungen definiert.

#### Zweites Stanz-Interface

Maschinen, die abwechselnd ein zweites Stand-Interface (zweite Stanzeinheit oder ein vergleichbares Medium) nutzen sollen, können auf ein zweites Paar der schnellen digitalen Ein- und Ausgänge der Steuerung (I/O-Paar) umgeschaltet werden. Für beide Stand-

Interfaces ist die volle Stanz-/Nibbel-Funktionalität nutzbar. Die Umschaltung zwischen erstem und zweitem Stanz-Interface erfolgt über die Befehle `SPIF1` und `SPIF2`.

**Hinweis**

Voraussetzung:Über Maschinendaten muss ein zweites I/O-Paar für die Stanzfunktionalität definiert sein (→ siehe Angaben des Maschinenherstellers!).

## Syntax

```
PON G... X... Y... Z...
SON G... X... Y... Z...
SONS G... X... Y... Z...
PONS G... X... Y... Z...
PDELAYON
PDELAYOF
PUNCHACC(<Smin>,<Amin>,<Smax>,<Amax>)
SPIF1/SPIF2
SPOF
```

## Bedeutung

| | | |
|---|---|---|
| `PON`: | Stanzen aktivieren | |
| `SON`: | Nibbeln aktivieren | |
| `PONS`: | Stanzen mit Vorspann aktivieren | |
| `SONS`: | Nibbeln mit Vorspann aktivieren | |
| `SPOF`: | Stanzen/Nibbeln deaktivieren | |
| `PDELAYON`: | Stanzen mit Verzögerung aktivieren | |
| `PDELAYOF`: | Stanzen mit Verzögerung deaktivieren | |
| `PUNCHACC`: | Wegabhängige Beschleunigung aktivieren<br>Parameter: | |
| | `<Smin>` | Kleinster Lochabstand |
| | `<Amin>` | Anfangsbeschleunigung<br>`<Amin>` kann größer als `<Amax>` sein. |
| | `<Smax>` | Größter Lochabstand |
| | `<Amax>` | Endbeschleunigung<br>`<Amax>` kann größer als `<Amin>` sein. |
| `SPIF1`: | **Erstes** Stanz-Interface aktivieren<br>Die Hubsteuerung erfolgt über das erste Paar der schnellen I/O. | |
| `SPIF2`: | **Zweites** Stanz-Interface aktivieren<br>Die Hubsteuerung erfolgt über das zweite Paar der schnellen I/O. | |
| | **Hinweis:**<br>Nach RESET oder Steuerungshochlauf ist immer das erste Stanz-Interface aktiv.<br>Wird nur ein Stanz-Interface benutzt, so muss dieses nicht programmiert werden. | |

## Beispiele

### Beispiel 1: Nibbeln aktivieren

| Programmcode | Kommentar |
|---|---|
| ... | |
| N70 X50 SPOF | ; Positionieren ohne Stanzauslösung. |
| N80 X100 SON | ; Nibbeln aktivieren, Auslösung eines Hubs vor der Bewegung (X=50) und am Ende der programmierten Bewegung (X=100). |
| ... | |

### Beispiel 2: Stanzen mit Verzögerung

| Programmcode | Kommentar |
|---|---|
| ... | |
| N170 PDELAYON X100 SPOF | ; Positionieren ohne Stanzauslösung, Aktivierung der verzögerten Stanzauslösung. |
| N180 X800 PON | ; Stanzen aktivieren. Nach Erreichen der Endposition wird Stanzhub verzögert ausgegeben. |
| N190 PDELAYOF X700 | ; Stanzen mit Verzögerung deaktivieren, normale Stanzauslösung am Ende der programmierten Bewegung. |
| ... | |

### Beispiel 3: Stanzen mit zwei Stand-Interfaces

| Programmcode | Kommentar |
|---|---|
| ... | |
| N170 SPIF1 X100 PON | ; Am Ende des Satzes erfolgt eine Hubauslösung auf dem ersten schnellen Ausgang. Das Signal "Hub aktiv" wird auf dem ersten Eingang überwacht. |
| N180 X800 SPIF2 | ; Die zweite Hubauslösung erfolgt auf dem zweiten schnellen Ausgang. Das Signal "Hub aktiv" wird auf dem zweiten Eingang überwacht. |
| N190 SPIF1 X700 | ; Die Hubsteuerung für alle weiteren Hübe erfolgt mit dem ersten Interface. |
| ... | |

## Weitere Informationen

### Stanzen und Nibbeln mit Vorspann (PONS/SONS)

Stanzen und Nibbeln mit Vorspann ist nicht gleichzeitig in mehreren Kanälen möglich. `PONS` bzw. `SONS` kann nur jeweils in einem Kanal aktiviert werden.

### Wegabhängige Beschleunigung (PUNCHACC)

Beispiel:
`PUNCHACC(2,50,10,100)`

*Lochabstände unter 2mm:*

Es wird mit einer Beschleunigung von 50% der Maximalbeschleunigung verfahren.

*Lochabstände von 2mm bis 10mm:*

Die Beschleunigung wird proportional zum Abstand auf 100% gesteigert.

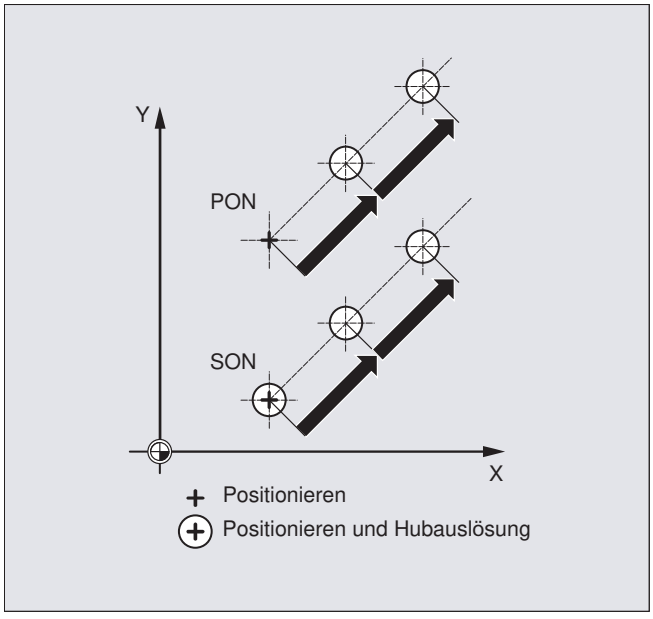
*Lochabstände größer als 10mm:*

Verfahren mit einer Beschleunigung von 100%.

### Auslösung des ersten Hubs

Die Auslösung des ersten Hubs nach Aktivierung der Funktion erfolgt beim Nibbeln und Stanzen zeitlich unterschiedlich:

- `PON/PONS`:
  - Alle Hübe - auch der des ersten Satzes nach Aktivierung - erfolgen im Satzende.
- `SON/SONS`:
  - Der erste Hub nach Aktivierung des Nibbelns erfolgt bereits im Satzanfang.
  - Alle weiteren Hübe werden jeweils im Satzende ausgelöst.




### Stanzen und Nibbeln auf der Stelle

Eine Hubauslösung erfolgt nur dann, wenn der Satz eine Verfahrinformation für die Stanz- oder Nibbelachsen (Achsen der aktiven Ebene) enthält.

Um dennoch einen Hub an gleicher Stelle auszulösen, wird eine der Stanz-/Nibbelachsen mit Verfahrweg 0 programmiert.

### Arbeiten mit drehbaren Werkzeugen

**Hinweis**

Um drehbare Werkzeuge tangential an die programmierte Bahn anzustellen, verwenden Sie die Tangentialsteuerung.

### Verwendung von M-Befehlen

Mit Hilfe der Makrotechnik ist es nach wie vor möglich, spezielle M-Funktionen statt der Sprachbefehle zu benutzen (Kompatibilität). Dabei gelten die folgenden Entsprechungen zu älteren Systemen:

```
M20, M23   ≙  SPOF
M22        ≙  SON
M25        ≙  PON
M26        ≙  PDELAYON
```

Beispiel für Makrodatei:

| **Programmcode** | **Kommentar** |
|---|---|
| DEFINE M25 AS PON | ; Stanzen ein |
| DEFINE M125 AS PONS | ; Stanzen mit Vorspann ein |
| DEFINE M22 AS SON | ; Nibbeln ein |
| DEFINE M122 AS SONS | ; Nibbeln mit Vorspann ein |
| DEFINE M26 AS PDELAYON | ; Stanzen mit Verzögerung ein |
| DEFINE M20 AS SPOF | ; Stanzen, Nibbeln aus |
| DEFINE M23 AS SPOF | ; Stanzen, Nibbeln aus |

Programmierbeispiel:

| **Programmcode** | **Kommentar** |
|---|---|
| ... | |
| N100 X100 M20 | ; Positionieren ohne Stanzauslösung. |
| N110 X120 M22 | ; Nibbeln aktivieren, vor und nach Bewegung Hubauslösung. |
| N120 X150 Y150 M25 | ; Stanzen aktivieren, Hubauslösung am Ende der Bewegung. |
| ... | |

## 3.19.2 Automatische Wegaufteilung

### Unterteilung in Teilstrecken

Bei aktiviertem Stanzen bzw. Nibbeln bewirken sowohl SPP als auch SPN eine Aufteilung der für die Bahnachsen programmierten Gesamtverfahrstrecke in eine Anzahl von gleichlangen Teilstrecken (äquidistante Wegaufteilung). Intern entspricht jede Teilstrecke einem Satz.

### Anzahl der Hübe

Beim Stanzen erfolgt der erste Hub am Endpunkt der ersten Teilstrecke, beim Nibbeln dagegen am Startpunkt der ersten Teilstrecke. Über die Gesamtfahrstrecke ergeben sich damit folgende Zahlen:

Stanzen: Anzahl der Hübe = Anzahl der Teilstrecken

Nibbeln: Anzahl der Hübe = Anzahl der Teilstrecken + 1

### Hilfsfunktionen

Hilfsfunktionen werden im ersten der erzeugten Sätze ausgeführt.

## Syntax

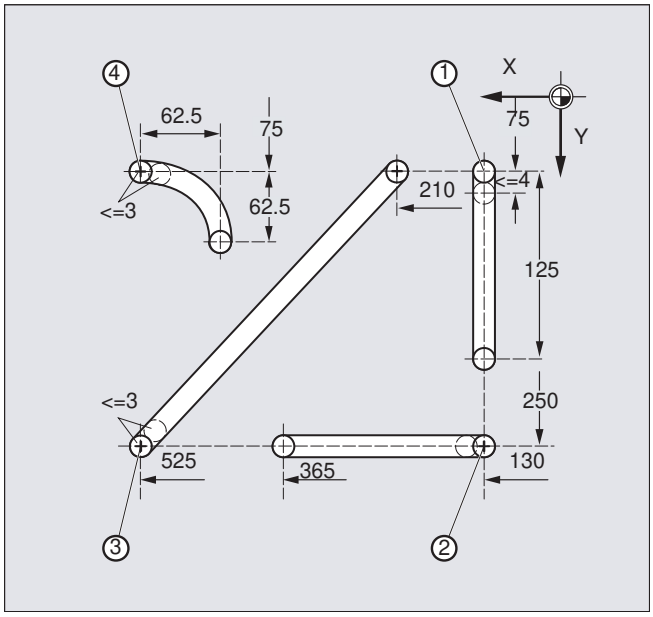
```
SPP=
SPN=
```

## Bedeutung

| | |
|---|---|
| SPP: | Größe der Teilstrecke (maximaler Hubabstand); modal wirksam |
| SPN: | Anzahl der Teilstrecken pro Satz; satzweise wirksam |

## Beispiel 1

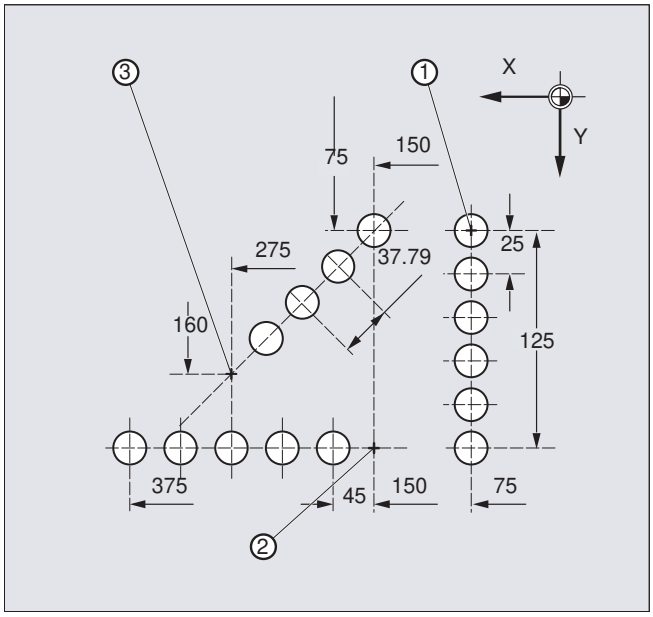
Die programmierten Nibbelstrecken sollen automatisch in gleichgroße Teilstrecken aufgeteilt werden.




| **Programmcode** | **Kommentar** |
|---|---|
| N100 G90 X130 Y75 F60 SPOF | ; Positionieren auf Startpunkt 1 |
| N110 G91 Y125 SPP=4 SON | ; Nibbeln ein; maximale Teilstreckenlänge für automatische Wegaufteilung: 4 mm |
| N120 G90 Y250 SPOF | ; Nibbeln aus; Positionieren auf Startpunkt 2 |
| N130 X365 SON | ; Nibbeln ein; maximale Teilstreckenlänge für automatische Wegaufteilung: 4 mm |
| N140 X525 SPOF | ; Nibbeln aus; Positionieren auf Startpunkt 3 |
| N150 X210 Y75 SPP=3 SON | ; Nibbeln ein; maximale Teilstreckenlänge für automatische Wegaufteilung: 3 mm |
| N160 X525 SPOF | ; Nibbeln aus; Positionieren auf Startpunkt 4 |
| N170 G02 X-62.5 Y62.5 I J62.5 SPP=3 SON | ; Nibbeln ein; maximale Teilstreckenlänge für automatische Wegaufteilung: 3 mm |
| N180 G00 G90 Y300 SPOF | ; Nibbeln aus |

### Beispiel 2

Für die einzelnen Lochreihen soll eine automatische Wegaufteilung erfolgen. Für die Aufteilung wird jeweils die maximale Teilstreckenlänge (SPP-Wert) angegeben.




| **Programmcode** | **Kommentar** |
|---|---|
| N100 G90 X75 Y75 F60 PON | ; Positionieren auf Startpunkt 1;<br>Stanzen ein Einzelloch stanzen |
| N110 G91 Y125 SPP=25 | ; Maximale Teilstreckenlänge für<br>automatische Wegaufteilung: 25 mm |
| N120 G90 X150 SPOF | ; Stanzen aus; Positionieren auf<br>Startpunkt 2 |
| N130 X375 SPP=45 PON | ; Stanzen ein; maximale Teilstreckenlänge<br>für automatische Wegaufteilung: 45 mm |
| N140 X275 Y160 SPOF | ; Stanzen aus; Positionieren auf<br>Startpunkt 3 |
| N150 X150 Y75 SPP=40 PON | ; Stanzen ein; anstelle der programmierten Teilstre-<br>ckenlänge von 40 mm wird die ;berechnete Teilstrecken-<br>länge von<br>37,79 mm verwendet. |
| N160 G00 Y300 SPOF | ; Stanzen aus; Positionieren |

## 3.19.2.1 Wegaufteilung bei Bahnachsen

### Länge der Teilstrecke SPP

Mit `SPP` geben Sie den maximalen Hubabstand und damit die maximale Länge der Teilstrecken an, in die die Gesamtverfahrstrecke aufgeteilt werden soll. Das Ausschalten des Befehls erfolgt mit `SPOF` oder `SPP=0`.

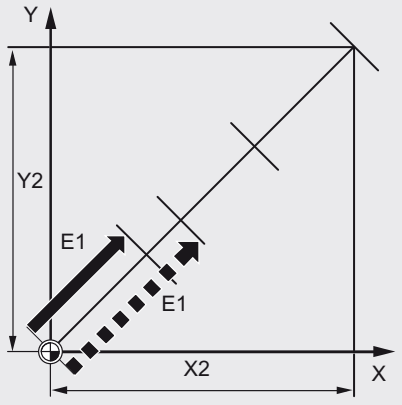
Beispiel:

```
N10 SON X0 Y0
N20 SPP=2 X10
```

Die Gesamtverfahrstrecke von 10 mm wird in 5 Teilstrecken von je 2 mm (`SPP=2`) aufgeteilt.

**Hinweis**

Die Wegaufteilung mit `SPP` erfolgt immer äquidistant: alle Teilstrecken sind gleich lang. Das heißt, die programmierte Teilstreckengröße (Wert von `SPP`) ist nur dann gültig, wenn der Quotient aus Gesamtverfahrstrecke und SPP-Wert ganzzahlig ist. Ist das nicht der Fall, so wird die Größe der Teilstrecke intern so reduziert, dass sich ein ganzzahliger Quotient ergibt.




X2/Y2 Programmierter Verfahrweg (Nibbel- oder Stanzsatz)
E1 Programmierte Teilstreckenlänge
E1' Automatisch abgerundete Teilstreckenlänge

Beispiel:

```
N10 G1 G91 SON X10 Y10
N20 SPP=3.5 X15 Y15
```

Bei der Gesamtverfahrstrecke von 15 mm und einer Teilstreckenlänge von 3,5 mm ergibt sich ein nicht ganzzahliger Quotient (4.28). Somit erfolgt eine Reduktion des SPP-Werts bis zum nächstmöglichen ganzzahligen Quotienten. In diesem Fall ergibt sich eine Teilstreckenlänge von 3 mm.

## Anzahl der Teilstrecken SPN

Mit `SPN` definieren Sie die Anzahl der Teilstrecken, die aus der Gesamtverfahrstrecke erzeugt werden soll. Die Länge der Teilstrecken wird automatisch berechnet. Da `SPN` satzweise wirksam ist, muss vorher Stanzen oder Nibbeln mit `PON` oder `SON` aktiviert werden.

### 3.19.2.2 Wegaufteilung bei Einzelachsen

Sind neben den Bahnachsen auch Einzelachsen als Stanz-Nibbel-Achse definiert, so können auch sie der automatischen Wegaufteilung unterliegen.

#### Verhalten der Einzelachse bei SPP

Die programmierte Länge der Teilstrecke (`SPP`) bezieht sich grundsätzlich auf die Bahnachsen. Daher wird in einem Satz, in dem neben der Einzelachsbewegung und dem SPP-Wert keine Bahnachse programmiert ist, der SPP-Wert ignoriert.

Sind sowohl Einzel- als auch Bahnachse im Satz programmiert, so richtet sich das Verhalten der Einzelachse nach der Einstellung des entsprechenden Maschinendatums.

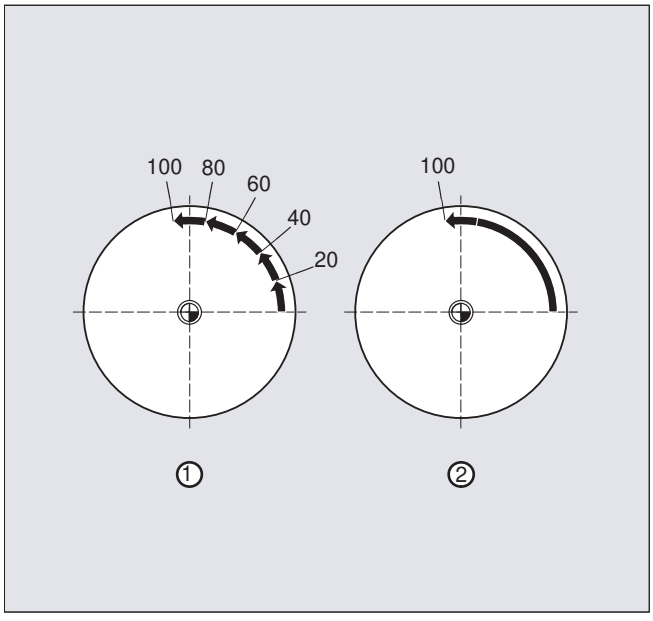
1. Standardeinstellung
   Der Weg der Einzelachse wird gleichmäßig auf die durch `SPP` erzeugten Zwischensätze verteilt.

Beispiel:

```
N10 G1 SON X10 A0
N20 SPP=3 X25 A100
```

Durch die Hubstrecke von 3 mm werden bei der Gesamtverfahrstrecke der X-Achse (Bahnachse) von 15 mm 5 Sätze erzeugt.

Die A-Achse dreht sich damit in jedem Satz um 20°.




1. Einzelachse ohne Wegaufteilung
   Die Einzelachse verfährt ihren Gesamtweg im ersten der erzeugten Sätze.
2. Unterschiedliche Wegaufteilung
   Das Verhalten der Einzelachse ist abhängig von der Interpolation der Bahnachsen:

- Kreisinterpolation: Wegaufteilung
- Linearinterpolation: keine Wegaufteilung

## Verhalten bei SPN

Die programmierte Anzahl von Teilstrecken gilt auch, wenn nicht gleichzeitig eine Bahnachse programmiert ist.

Voraussetzung: Einzelachse ist als Stanz-Nibbel-Achse definiert.

# 3.20 Schleifen

## 3.20.1 Schleifspezifische Werkzeugüberwachung ein-/ausschalten (TMON, TMOF)

Mit den vordefinierten Prozeduren TMON(...) und TMOF(...) wird die schleifspezifische Werkzeugüberwachung (Geometrie- und Drehzahlüberwachung) ein- bzw. ausgeschaltet.

### Voraussetzung

Die werkzeugspezifischen Parameter $TC_TPG1 bis $TC_TPG9 müssen gesetzt sein.

### Syntax

```
TMON(<TNr>)
...
TMOF(<TNr>)
```

### Bedeutung

| | |
|---|---|
| `TMON(...)`: | Schleifspezifische Werkzeugüberwachung einschalten<br>Der Befehl muss in dem Kanal programmiert werden, in dem die schleifspezifische Werkzeugüberwachung eingeschaltet werden soll. |
| `TMOF(...)`: | Schleifspezifische Werkzeugüberwachung ausschalten<br>Der Befehl muss in dem Kanal programmiert werden, in dem die schleifspezifische Werkzeugüberwachung ausgeschaltet werden soll. |
| `<TNr>`: | T-Nummer<br>**Hinweis:**<br>Nur erforderlich, wenn die Überwachung statt für das aktive, im Einsatz befindliche Werkzeug für eine nicht aktive Schleifscheibe ein- bzw. ausgeschaltet werden soll. |
| `TMOF(0)`: | Überwachung für alle Werkzeuge ausschalten |

# 3.21 Erweitertes Stillsetzen und Rückziehen (ESR)

Die Funktion "Erweitertes Stillsetzen und Rückziehen", im weiteren Verlauf mit ESR bezeichnet, bietet die Möglichkeit, in Fehlersituationen prozessabhängig flexibel zu reagieren:

- **Erweitertes Stillsetzen**
  Soweit es die spezifische Fehlersituation erlaubt, werden alle für das Erweiterte Stillsetzen freigegebenen Achsen geordnet stillgesetzt.
- **Rückziehen**
  Das sich im Eingriff befindliche Werkzeug wird schnellstmöglich vom Werkstück zurückgezogen.
- **Generatorbetrieb (SINAMICS-Antriebsfunktion "Vdc-Regelung")**
  Bei Unterschreitung eines parametrierbaren Werts der Zwischenkreisspannung, z. B. bei Ausfall der Netzspannung, wird die für das Rückziehen benötigte elektrische Energie durch Rückspeisen der Bremsenergie eines dafür vorgesehenen Antriebs (Generatorbetrieb) erzeugt.

## Auslösequellen

**Allgemeine Quellen (NC-extern/global oder BAG-/kanalspezifisch)**

- Digitale Eingänge (z. B. auf NCU-Baugruppe) bzw. das steuerungsinterne, rücklesbare Abbild digitaler Ausgänge ($A_IN, $A_OUT)
- Kanalzustand ($AC_STAT)
- VDI-Signale ($A_DBB)
- Sammelmeldungen einer Anzahl von Alarmen ($AC_ALARM_STAT)

**Axiale Quellen**

- Notrückzugsschwelle der Folgeachse (Synchronlauf der elektronischen Kopplung, $VC_EG_SYNCDIFF[<Folgeachse>])
- Antrieb: Zwischenkreis-Warnschwelle (drohende Unterspannung), $AA_ESR_STAT[<Achse>]
- Antrieb: Generator-Minimaldrehzahl-Schwelle (keine rückspeisbare Rotationsenergie mehr vorhanden), $AA_ESR_STAT[<Achse>].

## Verknüpfungslogik der statischen Synchronaktionen: Quellen-/Reaktions-Verknüpfung

Die flexiblen Verknüpfungsmöglichkeiten der statischen Synchronaktionen werden genutzt, um aufgrund von Quellen relativ zeitnah bestimmte Reaktionen auszulösen.

Die Verknüpfung aller relevanten Quellen mit Hilfe statischer Synchronaktionen liegt in den Händen des Anwenders. Er kann die Quellen-Systemvariablen als Ganzes oder mit Hilfe von Bit-Masken auch selektiv auswerten und hieran seine gewünschten Reaktionen knüpfen. Die statischen Synchronaktionen sind in allen Betriebsarten wirksam.

## Aktivierung

### Funktionsfreigabe

Die Funktionen Generatorbetrieb, Stillsetzen, Rückziehen werden durch Setzen des zugehörigen Steuersignals $AA_ESR_ENABLE freigegeben. Dieses Steuersignal kann von Synchronaktionen verändert werden.

### Funktionsauslösung

ESR wird gemeinsam für alle freigegebenen Achsen ausgelöst durch Setzen der Systemvariablen $AC_ESR_TRIGGER.

Der Generatorbetrieb wird "automatisch" im Antrieb bei Erkennung drohender Zwischenkreis-Unterspannung aktiv.

Antriebsautarkes Stillsetzen und/oder Rückziehen werden aktiv bei Erkennung eines Kommunikationsausfalls (zwischen NC und Antrieb) sowie bei Erkennung einer Zwischenkreis-Unterspannung im Antrieb (Konfiguration und Freigabe vorausgesetzt).

Antriebsautarkes Stillsetzen und/oder Rückziehen können zusätzlich auch von der NC-Seite her ausgelöst werden durch das Setzen des entsprechenden Steuersignals $AN_ESR_TRIGGER (Broadcast-Kommando an alle Antriebe).

## Weitere Informationen

Funktionshandbuch Achsen und Spindeln

# 3.21.1 NC-geführtes ESR

## 3.21.1.1 NC-geführtes Rückziehen (POLF, POLFA, POLFMASK, POLFMLIN)

Für NC-geführtes Rückziehen sind bestimmte Ausgangsbedingungen erforderlich (siehe "NC-geführtes Rückziehen (POLF, POLFA, POLFMASK, POLFMLIN) (Seite 993)"). Sind diese Voraussetzungen erfüllt, dann wird für die im Kanal konfigurierte(n) Rückzugsachse(n) durch Setzen der Systemvariablen $AC_ESR_TRIGGER (bzw. $AA_ESR_TRIGGER für Einzelachsen) Schnellabheben (LIFTFAST) aktiviert.

## Syntax

```
POLF(<Achse>)=<Position>
POLFA(<Achse>,<Typ>,<Position>)
POLFMASK(<Achse_1>,<Achse_2>,...)
POLFMLIN(<Achse_1>,<Achse_2>,...)
```

Für `POLFA` sind folgende Kurzformen erlaubt:

```
POLFA(<Achse>,<Typ>) ; Kurzform für Einzelachs-Rückziehen
POLFA(Achse,0/1/2) ; Schnelle Deaktivierung oder Aktivierung
POLFA(Achse,0,$AA_POLFA[Achse]) ; Bewirkt einen Vorlaufstopp
POLFA(Achse,0) ; Bewirkt keinen Vorlaufstopp
```

## Bedeutung

| | | | | |
|---|---|---|---|---|
| POLF: | Adresse zur Angabe der Zielposition der Rückzugsachse<br>POLF ist modal wirksam. | | | |
| | <Achse>: | Name der Geometrie- oder Kanal-/Maschinenachse, die zurückzieht | | |
| | <Position>: | Rückzugsposition | | |
| | | Typ: | REAL | |
| | | Für Geometrieachse gilt WKS, sonst MKS.<br>Bei gleichen Bezeichnern für Geometrie- und Kanal-/Maschinenachse wird im WKS zurückgezogen. | | |
| POLFA: | Vordefinierter Unterprogrammaufruf zur Angabe der Rückzugsposition von Einzelachsen | | | |
| | <Achse>: | Kanalachsbezeichner | | |
| | <Typ>: | Positionsangabemodus | | |
| | | Typ: | INT | |
| | | Wert: | 0: | Positionswert als ungültig markieren |
| | | | 1: | Positionswert ist absolut |
| | | | 2: | Positionswert ist inkrementell (Distanz) |
| | | **Hinweis:**<br>Ist eine Achse keine Einzelachse oder fehlt der Typ bzw. Typ=0, dann wird ein entsprechender Alarm gemeldet. | | |
| | <Position>: | Rückzugsposition (s. o.) | | |
| | | **Hinweis:**<br>Der Positionswert wird auch mit Typ=0 übernommen. Nur ist dieser Wert dann als ungültig markiert und muss für das Rückziehen neu programmiert werden. | | |
| POLFMASK: | Vordefinierter Unterprogrammaufruf zur Auswahl der Achsen, die nach Auslösung des Schnellabhebens **unabhängig voneinander** zurückgezogen werden sollen. | | | |
| | <Achse_1>,…: | Namen der Achsen, die beim Schnellabheben auf ihre mit POLF definierten Positionen fahren sollen.<br>Alle angegebenen Achsen müssen sich im gleichen Koordinatensystem befinden. | | |
| | POLFMASK() ohne Angabe einer Achse deaktiviert das Schnellabheben für alle Achsen, die unabhängig voneinander zurückgezogen wurden. | | | |
| POLFMLIN: | Vordefinierter Unterprogrammaufruf zur Auswahl der Achsen, die nach Auslösung des Schnellabhebens **im linearen Zusammenhang** zurückgezogen werden sollen. | | | |
| | <Achse_1>,...: | s. o. | | |
| | POLFMLIN() ohne Angabe einer Achse deaktiviert das Schnellabheben für alle Achsen, die im linearen Zusammenhang zurückgezogen wurden. | | | |

### Hinweis

Bevor über POLFMASK oder POLFMLIN das Schnellabheben auf eine feste Position freigegeben werden kann, muss für die ausgewählten Achsen eine Position mit POLF programmiert worden sein.

**Hinweis**

Werden Achsen nacheinander mit POLFMASK, POLFMLIN oder POLFMLIN, POLFMASK freigegeben, gilt für die jeweilige Achse immer die letzte Festlegung.

**Hinweis**

Die mit POLF programmierten Positionen und die Aktivierung durch POLFMASK oder POLFMLIN werden bei Teileprogrammstart gelöscht. Das heißt, der Anwender muss in jedem Teileprogramm die Werte für POLF und die selektierten Achsen in POLFMASK bzw. POLFMLIN neu programmieren.

**Hinweis**

Wenn bei Verwendung der Kurzformen POLFA nur der Typ geändert wird, dann muss der Anwender sicherstellen, dass entweder die Rückzugsposition oder der Rückzugsweg einen sinnvollen Wert enthält. Insbesondere sind die Rückzugsposition und der Rückzugsweg nach POWER ON neu zu setzen.

## Beispiel

Rückziehen einer Einzelachse:

| Programmcode | Kommentar |
|---|---|
| `MD37500 $MA_ESR_REACTION[AX1]=21` | `; NC-geführtes Rückziehen.` |
| `...` | |
| `$AA_ESR_ENABLE[AX1]=1` | |
| `POLFA(AX1,1,20.0)` | `; AX1 wird die axiale Rückzugsposition 20.0 (absolut) zugeteilt.` |
| `$AA_ESR_TRIGGER[AX1]=1` | `; Ab hier beginnt der Rückzug.` |

## Weitere Informationen

### Voraussetzungen für NC-geführtes Rückziehen

- Im Kanal ist eine Rückzugsachse für das NC-geführte Rückziehen projektiert:
  MD37500 $MA_ESR_REACTION = 21
- ESR muss für diese Achse freigegeben sein:
  $AA_ESR_ENABLE = 1
- Verzögerungszeiten sind definiert:
  MD21380 $MC_ESR_DELAY_TIME1
  MD21381 $MC_ESR_DELAY_TIME2
- Im Teileprogramm sind die achsspezifischen Rückzugspositionen mit POLF programmiert.
- Die Achsen sind mit POLFMASK/POLFMLIN für das NC-geführte Rückziehen ausgewählt.
- Für die Rückzugbewegung müssen die Freigabesignale gesetzt sein und gesetzt bleiben.

### NC-geführtes Rückziehen freigeben und starten

Wenn die Systemvariable $AC_ESR_TRIGGER = 1 gesetzt wird, und wenn in diesem Kanal eine Rückzugsachse konfiguriert ist (d. h. MD37500 $MA_ESR_REACTION = 21) und für diese Achse $AA_ESR_ENABLE = 1 gesetzt ist, dann wird in diesem Kanal Schnellabheben (LIFTFAST) aktiviert.

Die mit `POLF` (bzw. `LFPOS`) konfigurierte Abhebebewegung der mit `POLFMASK` oder `POLFMLIN` selektierte(n) Achse(n) ersetzt die für diese Achse(n) im Teileprogramm festgelegte Bahnbewegung.

Für den Rückzug steht maximal die Summe der Zeiten MD21380 $MC_ESR_DELAY_TIME1 und MD21381 $MC_ESR_DELAY_TIME2 zur Verfügung. Nach Ablauf dieser Zeitspanne wird auch für die Rückzugachse Schnellbremsen eingeleitet mit anschließendem Nachführen.

**Hinweis**

Die Erweiterte Rückzugbewegung (d. h. durch $AC_ESR_TRIGGER ausgelöstes LIFTFAST/ LFPOS) ist **nicht unterbrechbar** und kann nur durch Not-Halt vorzeitig beendet werden.

**Hinweis**

Durch $AC_ESR_TRIGGER ausgelöstes Rückziehen ist gegen Mehrfach-Rückzug verriegelt.

### Einzelachs-Rückziehen

Beim Einzelachs-Rückziehen muss mit `POLFA` die Rückzugsposition der Einzelachse programmiert worden sein und die folgenden Bedingungen müssen eingehalten werden:

- $AA_ESR_ENABLE = 1
- <Achse> muss zum Triggerzeitpunkt ($AA_ESR_TRIGGER = 1) eine Einzelachse sein.
- <Typ> muss entweder 1 oder 2 sein.

### Rückzugsrichtung beim Schnellabheben

Zum Zeitpunkt der Aktivierung des Schnellabhebens wird der gültige Frame berücksichtigt.

**Hinweis**

Frames mit Drehung beeinflussen über `POLF` auch die Richtung, in welche abgehoben wird.

### Achstausch

Rückzugsachsen sind immer genau einem NC-Kanal zuzuordnen und dürfen nicht zwischen den Kanälen getauscht werden. Beim Versuch, eine Rückzugsachse in einen anderen Kanal zu tauschen, wird ein Alarm gemeldet. Erst nachdem diese Achse mit $AA_ESR_ENABLE[AX] = 0 wieder deaktiviert wurde, kann die Achse in einen neuen Kanal getauscht werden. Nach dem erfolgten Achstausch können Achsen wieder mit $AA_ESR_ENABLE[AX] = 1 beaufschlagt werden.

### Neutrale Achsen

Neutrale Achsen können kein NC-geführtes ESR ausführen.

### 3.21.1.2 NC-geführtes Stillsetzen

Für die im Kanal konfigurierte(n) Stillsetzachse(n) wird durch Setzen der Systemvariablen $AC_ESR_TRIGGER (bzw. $AA_ESR_TRIGGER für Einzelachsen) das NC-geführte Stillsetzen aktiviert.

#### Voraussetzungen

- Im Kanal ist eine Stillsetzachse für das NC-geführte Stillsetzen projektiert:
  MD37500 $MA_ESR_REACTION = 22
- ESR muss für diese Achse freigegeben sein:
  $AA_ESR_ENABLE = 1
- Verzögerungszeiten sind definiert:
  MD21380 $MC_ESR_DELAY_TIME1 (Verzögerungszeit ESR-Achsen)
  MD21381 $MC_ESR_DELAY_TIME2 (ESR-Zeit für interpolatorisches Bremsen)

#### Ablauf

Für die Dauer der Zeitspanne in MD21380 interpoliert die Achse ungestört weiter wie programmiert. Nach Ablauf der Zeitspanne in MD21380 wird interpolatorisch geführtes Bremsen (Rampenstopp) eingeleitet. Für das interpolatorisch geführte Bremsen steht dann maximal die Zeitspanne in MD21381 zur Verfügung. Nach Ablauf dieser Zeitspanne wird Schnellbremsen mit anschließendem Nachführen eingeleitet.

#### Beispiel

Stillsetzen einer Einzelachse:

```
Programmcode                                          Kommentar
MD37500 $MC_ESR_REACTION[AX1]=22                      ; NC-geführtes Stillsetzen.
MD21380 $MC_ESR_DELAY_TIME1[AX1]=0.3
MD21381 $MC_ESR_DELAY_TIME2[AX1]=0.06
...
$AA_ESR_ENABLE[AX1]=1
$AA_ESR_TRIGGER[AX1]=1                                ; Ab hier beginnt das Stillsetzen.
```

## 3.21.2 Antriebsautarkes ESR

### 3.21.2.1 Antriebsautarkes Stillsetzen projektieren (ESRS)

Mit der Funktion `ESRS(...)` werden die Antriebsparameter für das "Stillsetzen" der antriebsautarken ESR-Funktion projektiert.

#### Syntax

```
ESRS(<Achse_1>,<Stillsetzzeit_1>[,...,<Achse_n>,<Stillsetzzeit_n>])
```

## Bedeutung

| | | |
|---|---|---|
| `ESRS(...)`: | Funktion zum Schreiben der Antriebsparameter für die ESR-Funktion "Stillsetzen"<br>Die Funktion:<br>• muss alleine im Satz stehen.<br>• löst Vorlaufstopp aus.<br>• kann nicht in Synchronaktionen verwendet werden. | |
| `<Achse_1>,`<br>`...,`<br>`<Achse_n>`: | Achse, für die antriebsautarkes Stillsetzen projektiert werden soll<br>Im Antrieb wird für diese Achse der Antriebsparameter p0888 (Konfiguration) geschrieben:<br>p0888 = 1 | |
| | Typ: | AXIS |
| | Wertebereich: | Kanalachsbezeichner |
| `<Stillsetzzeit_1>,`<br>`...,`<br>`<Stillsetzzeit_n>`: | Zeitdauer, über die der Antrieb nach dem Auftreten eines Fehlers mit dem aktuellen Drehzahlsollwert konstant weiter fährt<br>Im Antrieb wird für die angegebene Achse der Antriebsparameter p0892 (Zeitstufe) geschrieben:<br>p0892 = <Stillsetzzeit> | |
| | Einheit: | s |
| | Typ: | REAL |
| | Wertebereich: | 0.00 - 20.00 |
| In einem Funktionsaufruf können maximal 5 Achsen programmiert werden; n = 5 | | |

### 3.21.2.2 Antriebsautarkes Rückziehen projektieren (ESRR)

Mit der Funktion `ESRR(...)` werden die Antriebsparameter für das "Rückziehen" der antriebsautarken ESR-Funktion projektiert.

## Syntax

```
ESRR(<Achse_1>,<Rückzugsweg_1>,<Rückzugsgeschwindigkeit_1>[,...,<Achse_n>,<Rückzugsweg_n>,<Rückzugsgeschwindigkeit_n>])
```

## Bedeutung

| | | |
|---|---|---|
| `ESRR(...)`: | Funktion zum Schreiben der Antriebsparameter für die ESR-Funktion "Rückziehen"<br>Die Funktion:<br>• muss alleine im Satz stehen.<br>• löst Vorlaufstopp aus.<br>• kann nicht in Synchronaktionen verwendet werden. | |
| `<Achse_1>,`<br>`...,`<br>`<Achse_n>`: | Achse, für die antriebsautarkes Rückziehen projektiert werden soll<br>Im Antrieb wird für diese Achse der Antriebsparameter p0888 (Konfiguration) geschrieben:<br>p0888 = 2 | |
| | Typ: | AXIS |
| | Wertebereich: | Kanalachsbezeichner |
| `<Rückzugsweg_1>,`<br>`...,`<br>`<Rückzugsweg_n>`: | Der Rückzugsweg wird für den Antrieb in eine Rückzugsdrehzahl umgerechnet. Der Wert wird für die angegebene Achse in den Antriebsparameter p0893 (Drehzahl) geschrieben:<br>p0893 = (<Rückzugsweg _n> umgerechnet in Rückzugsdrehzahl) | |
| | Einheit: | mm/min, inch/min, Grad/min (abhängig von der Einheit der Achse) |
| | Typ: | REAL |
| | Wertebereich: | MIN - MAX |
| `<Rückzugs-`<br>`geschwindigkeit_1>,`<br>`...,`<br>`<Rückzugs-`<br>`geschwindigkeit_n>`: | Die Rückzugsgeschwindigkeit wird für den Antrieb in eine Zeitdauer umgerechnet. Der Wert wird für die angegebene Achse in den Antriebsparameter p0892 (Zeitstufe) [s] geschrieben:<br>p0892 = <Rückzugsweg_n> / <Rückzugsgeschwindigkeit _n> | |
| | Einheit: | mm/min, inch/min, Grad/min (abhängig von der Einheit der Achse) |
| | Typ: | REAL |
| | Wertebereich: | 0.00 - MAX |
| In einem Funktionsaufruf können maximal 5 Achsen programmiert werden; n = 5 | | |

# 3.22 Programmlaufzeit / Werkstückzähler

Zur Unterstützung des Werkzeugmaschinenbedieners werden Informationen zur Programmlaufzeit und Werkstückzahl bereitgestellt.

Diese Informationen können als Systemvariablen im NC- und/oder PLC-Programm bearbeitet werden. Gleichzeitig stehen diese Informationen für die Anzeige auf der Bedienoberfläche zur Verfügung.

## 3.22.1 Programmlaufzeit

Die Funktion "Programmlaufzeit" stellt NC-interne Timer zur Überwachung technologischer Prozesse zur Verfügung, die über NC- und Kanal-spezifische Systemvariablen im Teileprogramm und in Synchronaktionen gelesen werden können.

Der Trigger zur Laufzeitmessung ($AC_PROG_NET_TIME_TRIGGER) ist die einzige schreibbare Systemvariable der Funktion und dient zur selektiven Messung von Programmabschnitten. D. h. durch Beschreiben des Triggers im NC-Programm kann die Zeitmessung ein- und wieder ausgeschaltet werden.

| Systemvariable | Bedeutung | Aktivität |
|---|---|---|
| **NC-spezifisch** | | |
| $AN_SETUP_TIME | Zeit seit dem letzten Steuerungshochlauf mit Standardwerten ("Kaltstart") in Minuten<br>Wird bei jedem Steuerungshochlauf mit Standardwerten automatisch auf "0" zurückgesetzt. | • immer aktiv |
| $AN_POWERON_TIME | Zeit seit dem letzten Normalhochlauf der Steuerung ("Warmstart") in Minuten<br>Wird bei jedem Normalhochlauf der Steuerung automatisch auf "0" zurückgesetzt. | |
| **Kanal-spezifisch** | | |
| $AC_OPERATING_TIME | Gesamtlaufzeit von NC-Programmen in der Betriebsart Automatik in Sekunden<br>Der Wert wird mit jedem Steuerungshochlauf automatisch auf "0" zurückgesetzt. | • Aktivierung über MD27860<br>• nur Betriebsart AUTOMATIK |
| $AC_CYCLE_TIME | Laufzeit des angewählten NC-Programms in Sekunden<br>Der Wert wird mit dem Start eines neuen NC-Programms automatisch auf "0" zurückgesetzt. | |
| $AC_CUTTING_TIME | Bearbeitungszeit in Sekunden<br>Gemessen wird die Laufzeit der Bahnachsen (mindestens eine ist aktiv) ohne aktiven Eilgang in allen NC-Programmen zwischen NC-Start und Programmende / NC-Reset. Die Messung wird zusätzlich bei aktiver Verweilzeit unterbrochen.<br>Der Wert wird bei jedem Steuerungshochlauf mit Standardwerten automatisch auf "0" zurückgesetzt. | |
| | | |

| Systemvariable | Bedeutung | | Aktivität |
|---|---|---|---|
| $AC_ACT_PROG_NET_TIME | Aktuelle Netto-Laufzeit des aktuellen NC-Programms in Sekunden<br>Wird mit dem Start eines NC-Programms automatisch auf "0" zurückgesetzt. | | • immer aktiv<br>• nur Betriebsart AUTOMATIK |
| $AC_OLD_PROG_NET_TIME | Netto-Laufzeit des gerade korrekt mit M30 beendeten Programms in Sekunden | | |
| $AC_OLD_PROG_NET_TIME_COUNT | Änderungen auf $AC_OLD_PROG_NET_TIME<br>Nach POWER ON steht $AC_OLD_PROG_NET_TIME_COUNT auf "0".<br>$AC_OLD_PROG_NET_TIME_COUNT wird immer dann erhöht, wenn die Steuerung $AC_OLD_PROG_NET_TIME neu geschrieben hat. | | |
| $AC_PROG_NET_TIME_TRIGGER | Trigger zur Laufzeitmessung: | | • nur Betriebsart AUTOMATIK |
| | 0 | Neutraler Zustand<br>Der Trigger ist nicht aktiv. | |
| | 1 | Beenden<br>Beendet die Messung und kopiert den Wert aus $AC_ACT_PROG_NET_TIME in $AC_OLD_PROG_NET_TIME. $AC_ACT_PROG_NET_TIME wird auf "0" gesetzt und läuft danach weiter. | |
| | 2 | Start<br>Startet die Messung und setzt dabei $AC_ACT_PROG_NET_TIME auf "0". $AC_OLD_PROG_NET_TIME wird nicht verändert. | |
| | 3 | Stopp<br>Stoppt die Messung. Verändert $AC_OLD_PROG_NET_TIME nicht und hält $AC_ACT_PROG_NET_TIME bis zum Fortsetzen konstant. | |
| | 4 | Fortsetzen<br>Fortsetzen der Messung, d. h. eine vorher gestoppte Messung wird wieder aufgenommen. $AC_ACT_PROG_NET_TIME läuft weiter. $AC_OLD_PROG_NET_TIME wird nicht verändert. | |
| Durch POWER ON werden alle Systemvariablen auf "0" zurückgesetzt! | | | |

**Hinweis**

**Maschinenhersteller**

Das Einschalten der aktivierbaren Timer erfolgt über das Maschinendatum MD27860 $MC_PROCESSTIMER_MODE.

Das Verhalten der aktiven Zeitmessungen bei bestimmten Funktionen (z. B. GOTOS, Override = 0%, aktiver Probelaufvorschub, Programmtest, ASUP, PROG_EVENT, …) wird konfiguriert über die Maschinendaten MD27850 $MC_PROG_NET_TIMER_MODE und MD27860 $MC_PROCESSTIMER_MODE.

**Weitere Informationen:** Funktionshandbuch Basisfunktionen

**Hinweis**

**Restzeit für ein Werkstück**

Wenn nacheinander gleiche Werkstücke produziert werden, kann mit den folgenden Timerwerten die verbleibende Restzeit für ein Werkstück ermittelt werden:

- Bearbeitungszeit für das zuletzt produzierte Werkstück (siehe $AC_OLD_PROG_NET_TIME)
- Aktuelle Bearbeitungszeit (siehe $AC_ACT_PROG_NET_TIME)

Die Restzeit wird zusätzlich zur aktuellen Bearbeitungszeit auf der Bedienoberfläche anzeigt.

**Hinweis**

**Verwendung von STOPRE**

Die Systemvariablen $AC_OLD_PROG_NET_TIME und $AC_OLD_PROG_NET_TIME_COUNT erzeugen keinen impliziten Vorlaufstopp. Bei der Verwendung im Teileprogramm ist das unkritisch, wenn der Wert der Systemvariablen aus dem vorangegangen Programmlauf stammt. Wenn aber der Trigger zur Laufzeitmessung ($AC_PROG_NET_TIME_TRIGGER) hochfrequent geschrieben wird und sich dadurch $AC_OLD_PROG_NET_TIME sehr oft ändert, dann sollte im Teileprogramm ein explizites STOPRE verwendet werden.

## Randbedingungen

- **Satzsuchlauf**
  Bei Satzsuchlauf werden keine Programmlaufzeiten ermittelt.
- **REPOS**
  Die Zeitdauer eines REPOS-Vorgangs wird der aktuellen Bearbeitungszeit ($AC_ACT_PROG_NET_TIME) angerechnet.

## Beispiele

### Beispiel 1: Zeitdauer von "mySubProgrammA" messen

```
Programmcode
...
```

**Programmcode**

```
N50 DO $AC_PROG_NET_TIME_TRIGGER=2
N60 FOR ii= 0 TO 300
N70 mySubProgrammA
N80 DO $AC_PROG_NET_TIME_TRIGGER=1
N95 ENDFOR
N97 mySubProgrammB
N98 M30
```

Nachdem das Programm die Zeile N80 verarbeitet hat, steht in $AC_OLD_PROG_NET_TIME die Nettolaufzeit von "mySubProgrammA".

Der Wert von $AC_OLD_PROG_NET_TIME:

- beibt über M30 hinaus erhalten.
- wird nach jedem Schleifendurchlauf aktualisiert.

**Beispiel 2: Zeitdauer von "mySubProgrammA" und "mySubProgrammC" messen**

**Programmcode**

```
...
N10 DO $AC_PROG_NET_TIME_TRIGGER=2
N20 mySubProgrammA
N30 DO $AC_PROG_NET_TIME_TRIGGER=3
N40 mySubProgrammB
N50 DO $AC_PROG_NET_TIME_TRIGGER=4
N60 mySubProgrammC
N70 DO $AC_PROG_NET_TIME_TRIGGER=1
N80 mySubProgrammD
N90 M30
```

## 3.22.2 Werkstückzähler

Die Funktion "Werkstückzähler" stellt diverse Zähler zur Verfügung, die insbesondere für die steuerungsinterne Zählung von Werkstücken verwendet werden können.

Die Zähler existieren als kanalspezifische Systemvariablen mit Schreib- und Lese-Zugriff im Wertebereich von 0 bis 999 999 999.

| Systemvariable | Bedeutung |
|---|---|
| $AC_REQUIRED_PARTS | Anzahl der zu fertigenden Werkstücke (Soll-Werkstückzahl)<br>In diesem Zähler kann die Anzahl der Werkstücke definiert werden, bei dessen Erreichen die Ist-Werkstückzahl ($AC_ACTUAL_PARTS) auf "0" zurückgesetzt wird. |
| $AC_TOTAL_PARTS | Anzahl der insgesamt gefertigten Werkstücke (Ist-Werkstückzahl Gesamt)<br>Dieser Zähler gibt die Anzahl aller ab Startzeitpunkt gefertigten Werkstücke an. Der Wert wird nur bei einem Steuerungshochlauf mit Standardwerten automatisch auf "0" zurückgesetzt. |

| Systemvariable | Bedeutung |
|---|---|
| $AC_ACTUAL_PARTS | Anzahl der gefertigten Werkstücke (Ist-Werkstückzahl)<br>In diesem Zähler wird die Anzahl aller ab Startzeitpunkt gefertigten Werkstücke registriert. Bei einem Erreichen der Soll-Werkstückzahl ($AC_REQUIRED_PARTS) wird der Zähler automatisch auf "0" zurückgesetzt ($AC_REQUIRED_PARTS > 0 vorausgesetzt). |
| $AC_SPECIAL_PARTS | Anzahl der vom Anwender gezählten Werkstücke<br>Dieser Zähler erlaubt dem Anwender eine Werkstückzählung nach eigener Definition. Definiert werden kann eine Alarmausgabe bei Erreichen der Soll-Werkstückzahl ($AC_REQUIRED_PARTS). Eine Nullung des Zählers muss der Anwender selbst vornehmen. |

**Hinweis**

Alle Werkstückzähler werden bei einem Steuerungshochlauf mit Standardwerten auf "0" gesetzt und können unabhängig von ihrer Aktivierung gelesen und geschrieben werden.

**Hinweis**

Über kanalspezifische Maschinendaten kann auf die Zähler-Aktivierung, den Zeitpunkt der Nullung und den Zählalgorithmus Einfluss genommen werden.

**Hinweis**

**Werkstückzählung mit anwenderdefiniertem M-Befehl**

Über Maschinendaten kann eingestellt werden, dass die Zählimpulse für die verschiedenen Werkstückzähler statt über das Programmende `M2`/`M30` über anwenderdefinierte M-Befehle ausgelöst werden.

# 3.23 Weitere Funktionen

## 3.23.1 Maschinendaten wirksam setzen (NEWCONF)

Mit dem Befehl NEWCONF werden alle Maschinendaten wirksam gesetzt. Die Funktion kann auch in der Bedienoberfläche HMI durch Betätigen des Softkeys "MD wirksam setzen" aktiviert werden.

Bei der Ausführung der Funktion "NEWCONF" erfolgt ein impliziter Vorlaufstopp, d. h. die Bahnbewegung wird unterbrochen.

### Syntax

```
NEWCONF
```

### Bedeutung

| | |
|---|---|
| NEWCONF: | Befehl zum Wirksamsetzen aller Maschinendaten der Wirksamkeitsstufe "NEW_CONFIG" |

### NEWCONF aus dem Teileprogramm kanalübergreifend ausführen

Werden axiale Maschinendaten aus dem Teileprogramm heraus verändert und anschließend mit NEWCONF aktiviert, so setzt der Befehl NEWCONF nur die Maschinendaten aktiv, die Änderungen für den Kanal des Teileprogramms bewirken.

**Hinweis**

Um alle Änderungen sicher wirksam werden zu lassen, muss der Befehl NEWCONF in jedem Kanal ausgeführt werden, in dem auch die von den veränderten Maschinendaten betroffenen Achsen oder Funktionen aktuell gerechnet werden.

Bei NEWCONF werden keine axialen Maschinendaten wirksam gesetzt.

Für PLC-kontrollierte Achsen muss ein axialer RESET ausgeführt werden.

### Beispiel

Fräsbearbeitung: Bohrposition mit verschiedenen Technologien bearbeiten

```
Programmcode                          Kommentar
N10 $MA_CONTOUR_TOL[AX]=1.0           ; Maschinendatum ändern.
N20 NEWCONF                           ; Maschinendaten wirksam setzen.
...
```

## 3.23.2 Vorhandenen NC-Sprachumfang prüfen (STRINGIS)

Mit der Funktion "STRINGIS(...)" kann geprüft werden, ob der angegebene String als Element der NC-Programmiersprache im aktuellen Sprachumfang zur Verfügung steht.

### Definition

```
INT STRINGIS(STRING <Name>)
```

### Syntax

```
STRINGIS(<Name>)
```

### Bedeutung

| | |
|---|---|
| STRINGIS: | Funktion mit Rückgabewert |
| <Name>: | Name des zu prüfenden Elementes der NC-Programmiersprache |
| Rückgabewert: | Das Format des Rückgabewertes ist yxx (Dezimal). |

#### Elemente der NC-Programmiersprache

Folgende Elemente der NC-Programmiersprache können geprüft werden:

- G-Befehle aller existierenden G-Gruppen z.B. "G0", "INVCW", "POLY", "ROT", "KONT", "SOFT", "CUT2D", "CDON", "RMBBL", "SPATH"
- DIN- oder NC-Adressen wie z.B. "ADIS", "RNDM", "SPN", "SR", "MEAS"
- Funktionen z.B. "TANG(...)" oder "GETMDACT"
- Prozeduren z.B. "SBLOF".
- Schlüsselworte z.B. "ACN", "DEFINE" oder "SETMS"
- Systemdaten z.B. Maschinendaten $M... , Settingdaten $S... oder Optionsdaten $O...
- Systemvariable $A... , $V... , $P...
- Rechenparameter R...
- Zyklennamen von aktivierten Zyklen
- GUD- und LUD-Variablen
- Makro-Namen
- Label-Namen

#### Rückgabewert

Der Rückgabewert ist nur in den erst 3 Dezimalstellen relevant. Das Format des Rückgabewertes ist yxx, mit y = Basisinformation und xx = Detailinformation.

| Rückgabewert | Bedeutung |
|---|---|
| 000 | Der String 'name' ist im vorliegenden System nicht bekannt [1)] |
| 100 | Der String 'name' ist ein Element der NC-Programmiersprache, aber aktuell nicht programmierbar (Option/ Funktion ist inaktiv) |

| Rückgabewert | Bedeutung | |
|---|---|---|
| 2xx | Der String 'name' ist ein programmierbares Element der NC-Programmiersprache (Option/Funktion ist aktiv). Die Detailinformation xx enthält weitere Informationen über die Art des Elements: | |
| | **xx** | **Bedeutung** |
| | 01 | DIN–Adresse oder NC–Adresse [2] |
| | 02 | G–Befehl (z.B. G04, INVCW) |
| | 03 | Funktion mit Rückgabewert |
| | 04 | Funktion ohne Rückgabewert |
| | 05 | Schlüsselwort (z.B. DEFINE) |
| | 06 | Maschinen- ($M...), Setting- ($S...) oder Optionsdatum ($O...) |
| | 07 | Systemparameter, z.B. Systemvariable ($...) oder Rechenparameter (R...) |
| | 08 | Zyklus (Der Zyklus muss im NC geladen und die Zyklenprogramme aktiv sein [3] ) |
| | 09 | GUD–Variable (Die GUD-Variable muss der in GUD–Definitionsdateien definierten und die GUD-Variablen aktiviert sein) |
| | 10 | Makroname (Das Makro muss in der Makro–Definitionsdateien definierten und Makros aktivierten sein) [4] |
| | 11 | LUD–Variable des aktuellen Teileprogramms |
| | 12 | ISO G–Befehl (ISO Sprachmodus muss aktiv ist) |
| 400 | Der String 'name' ist eine NC–Adresse, die nicht als xx == 01 oder xx == 10 erkannt wurde und die nicht G oder R ist [2] | |
| y00 | Keine spezifische Zuordnung möglich | |

**1**) Steuerungs-abhängig ist unter Umständen nur eine Untermenge der Siemens NC-Sprachbefehle bekannt, z.B. SINUMERIK 802D sl. Auf diesen Steuerungen wird für Strings, die prinzipiell Siemens NC-Sprachbefehle sind, der Wert 0 zurückgegeben. Dieses Verhalten kann über MD10711 $MN_NC_LANGUAGE_CONFIGURATION verändert werden. Bei MD10711 = 1 wird dann für Siemens NC-Sprachbefehle immer der Wert 100 zurückgegeben.

**2**) NC–Adressen sind folgende Buchstaben: A, B, C, E, I, J, K, Q, U, V, W, X, Y, Z. Diese NC–Adressen können auch mit einer Adresserweiterung programmiert werden. Die Adresserweiterung kann bei der Prüfung mit STRINGIS angegeben werden. Beispiel: 201 == STRINGIS("A1").
Die Buchstaben: D, F, H, L, M, N, O, P, S, T sind NC–Adressen oder Hilfsfunktionen die anwenderdefiniert verwendet werden. Für sie wird immer der Wert 400 zurückgegeben. Beispiel: 400 == STRINGIS( "D" ). Diese NC–Adressen können bei der Prüfung mit STRINGIS nicht mit Adresserweiterung angegeben werden.
Beispiel: 000 == STRINGIS("M02"), aber 400 == STRINGIS("M").

**3**) Namen von Zyklenparametern können mit STRINGIS nicht geprüft werden.
**4**) Als Makro definierte Adress z.B. G, H, M, L werden als Makro identifiziert

### Beispiele

In den folgenden Beispielen wird angenommen, dass die als String angegebenen NC-Sprachelement, sofern nicht besonders vermerkt, in der Steuerung prinzipiell programmierbar sind.

1. Der String "T" ist als Hilfsfunktion definiert:
```
400 == STRINGIS("T")
000 == STRINGIS ("T3")
```
2. Der String "X" ist als Achse definiert:
```
201 == STRINGIS("X")
201 == STRINGIS("X1")
```
3. Der String "A2" ist als NC–Adresse mit Erweiterung definiert:
```
201 == STRINGIS("A")
201 == STRINGIS("A2")
```

4. Der String "INVCW" ist als benamter G–Befehl definiert:
   `202 == STRINGIS("INVCW")`
5. Der String "$MC_GCODE_RESET_VALUES" ist als Maschinendatum definiert:
   `206 == STRINGIS("$MC_GCODE_RESET_VALUES")`
6. Der String "GETMDACT" ist eine NC–Sprachfunktion:
   `203 == STRINGIS("GETMDACT ")`
7. Der String "DEFINE" ist ein Schlüsselwort:
   `205 == STRINGIS("DEFINE")`
8. Der String "$TC_DP3" ist ein Systemparameter (Werkzeuglängenkomponente):
   `207 == STRINGIS("$TC_DP3")`
9. Der String "$TC_TP4" ist ein Systemparameter (Werkzeuggröße):
   `207 == STRINGIS("$TC_TP4")`
10. Der String "$TC_MPP4" ist ein Systemparameter (Magazinplatzzustand):
    - Die Werzeugmagazin–Verwaltung ist aktiv: `207 == STRINGIS("$TC_MPP4") ;`
    - Die Werzeugmagazin–Verwaltung ist nicht aktiv: `000 == STRINGIS("$TC_MPP4")`

    Siehe auch unten Absatz: Werkzeugmagazin-Verwaltung.
11. Der String "MACHINERY_NAME" ist als GUD-Variable definiert:
    `209 == STRINGIS("MACHINERY_NAME")`
12. Der String "LONGMACRO" ist als Makro definiert:
    `210 == STRINGIS("LONGMACRO")`
13. Der String "MYVAR" ist als LUD-Variable definiert:
    `211 == STRINGIS("MYVAR")`
14. Der String "XYZ" ist kein im NC bekannter Befehl, GUD-Variable, Makro- oder Zyklen-Name:
    `000 == STRINGIS("XYZ")`

#### Werkzeugmagazin-Verwaltung

Ist die Funktion Werkzeugmagazin-Verwaltung nicht aktiv, liefert STRINGIS für die Systemparameter der Werkzeugmagazin-Verwaltung, unabhängig vom Maschinendatum

- MD10711 $MN_NC_LANGUAGE_CONFIGURATION

immer den Wert 000.

#### ISO Modus

Ist die Funktion "ISO Modus" aktiv:

- MD18800 $MN_MM_EXTERN_LANGUAGE (Aktivierung externer NC-Sprachen)
- MD10880 $MN_ MM_EXTERN_CNC_SYSTEM (zu adaptierendes Steuerungssystems)

überprüft STRINGIS den angegebenen String zuerst als SINUMERIK G-Befehl. Ist der String kein SINUMERIK G-Befehl wird er anschließend als ISO G-Befehl überprüft.

Programmierte Umschaltungen (`G290` (SINUMERIK Mode), `G291` (ISO Mode)) haben auf STRINGIS keine Auswirkung.

### Beispiel

Die für die Funktion STRINGIS(...) relevanten Maschinendaten haben folgende Werte:

- MD10711 $MN_NC_LANGUAGE_CONFIGURATION = 2 (Es werden nur die NC-Sprachbefehle als bekannt angesehen, deren Optionen gesetzt sind)
- MD19410 $ON_TRAFO_TYPE_MASK = 'H0' (Option: Transformationen)
- MD10700 $MN_PREPROCESSING_LEVEL='H43' (Vorverarbeitung für Zyklen aktiv)

Das folgende Beispielprogramm wird ohne Fehlermeldung abgearbeitet:

| **Programmcode** | **Kommentar** |
|---|---|
| `N1 R1=STRINGIS("TRACYL")` | `; R1 == 0, da TRACYL wegen der fehlenden Transformations-Option als "nicht bekannt" erkannt wird` |
| `N2 IF STRINGIS("TRACYL") == 204` | |
| `N3  TRACYL(1,2,3)` | `; N3 wird übersprungen` |
| `N4 ELSE` | |
| `N5  G00` | `; und stattdessen N5 ausgeführt` |
| `N6 ENDIF` | |
| `N7 M30` | |

## 3.23.3 Fenster aus dem Teileprogramm interaktiv aufrufen (MMC)

Über das vordefinierte Unterprogramm MMC(...) können aus einem NC-Programm heraus anwenderspezifische Dialoge auf der Bedienoberfläche angezeigt werden.

Die Projektierung der Dialoge kann für folgende Dialogarten erfolgen:

- Run MyScreens
- Easy XML
- Anwender-XML

**Weitere Informationen:**

- Programmierhandbuch Run MyScreens
- Programmierhandbuch Easy XML

### Syntax

```
MMC("<ADDRESS>,<COMMAND>,<FILE>,<DIALOG>","<QUIT>")
```

### Bedeutung

| | |
|---|---|
| `MMC(...)`: | Unterprogrammbezeichner<br>Die Parameter werden platzcodiert und durch Komma getrennt innerhalb von zwei Strings, dem Kommando-String und dem Quittungs-String, angegeben. |
| Parameter innerhalb des Kommando-Strings: | |

| | | |
|---|---|---|
| <ADDRESS>: | Bedienbereich, in dem die projektierten Anwenderdialoge ausgeführt werden | |
| | **Funktion** | **Bedienbereiche** |
| | Anwenderdialog "Run MyScreens" | CYCLES |
| | Anwenderdialog "Easy XML" | CYCLES |
| | Anwender-XML | XML |
| | Popup "Run MyScreens" | POPUPDLG |
| | Popup "Easy XML" | POPUPDLG |
| <COMMAND>: | Auszuführendes Kommando | |
| | **Funktion** | **Kommandos** |
| | Anwenderdialog "Run MyScreens" | PICTURE_ON, PICTURE_OFF |
| | Anwenderdialog "Easy XML" | PICTURE_ON, PICTURE_OFF |
| | Anwender-XML | XML_ON, XML_OFF |
| | Popup "Run MyScreens" | PICTURE_ON, PICTURE_OFF |
| | Popup "Easy XML" | PICTURE_ON, PICTURE_OFF |
| <FILE>: | Name der Datei, in der der anzuzeigende Dialog programmiert ist | |
| | **Funktion** | **Dateien** |
| | Anwenderdialog "Run MyScreens" | <name>.com |
| | Anwenderdialog "Easy XML" | <name>.xml |
| | Anwender-XML | <name>.xml |
| | Popup "Run MyScreens" | <name>.com |
| | Popup "Easy XML" | <name>.xml |
| | Popup "Easy XML" mit Projektierung direkt im NC-Programm (siehe Beispiel 2) | xmldial_emb.xml |
| <DIALOG>: | Name des anzuzeigenden Dialogs | |
| | **Funktion** | **Dialogname** |
| | Alle Funktionen außer Popup "Easy XML" mit Projektierung direkt im NC-Programm | In der Datei <FILE> projektierter Name des Dialogs |
| | Popup "Easy XML" mit Projektierung direkt im NC-Programm (siehe Beispiel 3) | main |
| Parameter innerhalb des Quittungs-Strings: | | |
| <QUIT>: | Quittungstyp | |
| | N: | Keine Quittierung.<br>Die Programmbearbeitung wird nach Absenden des Kommandos fortgeführt. Es erfolgt keine Rückmeldung, wenn das Kommando nicht erfolgreich ausgeführt werden konnte.<br>**Hinweis**<br>Quittungstyp "N" ist zu verwenden, wenn im NC-Programm eine Anzeigezeit (Verweilzeit) programmiert wird (siehe unten Beispiel 2) |
| | A: | Asynchrone Quittierung<br>Die Programmabarbeitung wird nach Absetzen des Befehls fortgeführt. Der Rückgabewert wird in einer im Rahmen der Dialogprojektierung definierten anwenderspezifischen Quittungsvariablen (GUD-Variable) abgelegt und kann im NC-Programm gelesen werden. |

## Randbedingungen

- Die Definitionsdateien *.com der Dialoge müssen im Ordner "proj" abgelegt werden.
- Die Easy XML Definitionsdateien *.xml der Dialoge müssen im Ordner "appl" abgelegt werden.
  Werden die Definitionsdateien in einem anderen Verzeichnis abgelegt, muss die Pfadangabe indirekt, ausgehend vom Verzeichnis "appl", erfolgen.
- Anwenderdefinierte Dialoge können nicht gleichzeitig aus unterschiedlichen Kanälen heraus angezeigt werden.
- In der Simulation wird die MMC-Funktionalität nicht unterstützt.

## Beispiele

### Beispiel 1

Anzeige eines Dialogs und Reaktion auf die Anwenderbedienung in einem NC-Programm.

```
Programmcode                                    Kommentar
; Die Quittungsvariable QUIT wurde bereits als globale Anwendervariable (GUD)
; vom Typ STRING bei der Projektierung des Dialoges angelegt:
; DEF NCK STRING[20] QUIT
QUIT = "XXX"                                   ; Quittungsvariable initialisieren
G4 F5
MMC("CYCLES,PICTU-                             ; Dialog anzeigen
RE_ON,test.com,test1","A")                     ;  - Bedienbereich: CYCLES
                                               ;  - Bildstatus: PICTURE_ON (anzeigen)
                                               ;  - Dialogbild-Datei: test.com
                                               ;  - Dialogbild: test1
INPUT:                                         ; Warten auf Anwenderbedienung
   STOPRE                                      ;    Vorlaufstop
   IF MATCH (QUIT,"RUN") >= 0 GOTOF WORK       ;    Softkey "RUN"
   IF MATCH (QUIT,"CHK") >= 0 GOTOF CHECK      ;    Softkey "CHK"
GOTOB INPUT                                    ; => Warten

WORK:                                          ; Softkey "RUN" gedrückt
MSG("Weiter mit Bearbeitung -> NC-Start")      ; Meldung ausgeben
MMC("CYCLES,PICTURE_OFF","N")                  ; Dialog schließen
M0                                             ; Warten auf NC-Start
GOTOF END                                      ; => Programmende

CHECK:                                         ; Softkey "CHK" gedrückt
MSG("Position anfahren -> NC-Start")           ; Meldung ausgeben
MMC("CYCLES,PICTURE_OFF","N")                  ; Dialog schließen
M0                                             ; Warten auf NC-Start
GOTOF END                                      ; => Programmende
```

| Programmcode | Kommentar |
|---|---|
| END: | |
| ... | |

### Beispiel 2

Die Anzeigezeit eines Dialogs wird im NC-Programm z. B. über eine Verweilzeit festgelegt.

| Programmcode | Kommentar |
|---|---|
| F1000 G94 | |
| ... | |
| MMC("POPUPDLG,PICTURE_**ON**,xmldial_emb.xml,main","N") | ; Dialog anzeigen |
| X200 | |
| Z40 | |
| MMC("POPUPDLG,PICTURE_**OFF**","N") | ; Dialog schließen |

### Beispiel 3

Einbetten eines PopUp-Scripts in ein NC-Programm und dessen Verwendung.

**Programmcode**

```
PROC POPUP_TEST
; ----------------------------- Script -----------------------------
; <main_dialog entry="rpara_main">
;    <let name="xpos" />
;    <let name="ypos" />
;    <let name="field_name" type="string" />
;    <let name="num" />
;    <menu name="rpara_main">
;       <open_form name="rpara_form"/>
;       <softkey_back>
;          <close_form />
;       </softkey_back>
;    </menu>
;    <form name="rpara_form">
;       <init>
;          <caption>mask from NC part program</caption>
;          <let name="count" >0</let>
;          <op>
;             xpos = 120;
;             ypos = 34;
;             "nck/Channel/Parameter/R[10]" = 10;
;          </op>
;          <!-- load the number of controls -->
;          <op>
;             num = "nck/Channel/Parameter/R[10]";
;          </op>
;          <while>
```

**Programmcode**

```
;             <condition> count < num</condition>
;             <print name="field_name" text="edit%d">count</print>
;             <op>
;                ypos = ypos + 24;
;                count = count + 1;
;             </op>
;          </while>
;       </init>
;       <paint>
;          <op>
;             xpos =  8;
;             ypos = 36;
;             count = 0;
;          </op>
;          <while>
;             <condition>count < num</condition>
;             <print name="field_name" text="R-Parameter%d">count</print>
;             <text xpos = "$xpos" ypos = "$ypos" >$$$field_name</text>
;             <op>
;                ypos = ypos + 24;
;                count = count + 1;
;             </op>
;          </while>
;       </paint>
;    </form>
; </main_dialog>
; ========================= Programmteil ============================
...
G94 F100
MMC("POPUPDLG,PICTURE_ON,xmldial_emb.xml,main","N")
G4 F4
X200
MMC("POPUPDLG,PICTURE_OFF","N")
G4 F2
X0
...
```

## 3.23.4 Process DataShare - Ausgabe auf ein externes Gerät/Datei (EXTOPEN, WRITE, EXTCLOSE)

Das Schreiben von Daten aus einem Teileprogramm heraus auf ein externes Gerät/Datei erfolgt in drei Schritten:

1. Externes Gerät/Datei öffnen
   Mit dem Befehl EXTOPEN wird das externe Gerät/Datei für den Kanal zum Schreiben geöffnet.
2. Daten schreiben
   Das Ausgabedatum kann mit den Stringfunktionen der NC-Sprache, z. B. SPRINT, aufbereitet werden. Das Schreiben selbst erfolgt über den WRITE-Befehl.
3. Externes Gerät/Datei schließen
   Mit dem Befehl EXTCLOSE oder durch Erreichen des Programmendes (M30) sowie bei Kanal-Reset wird das im Kanal belegte externe Gerät/Datei wieder freigegeben.

### Syntax

```
DEF INT <Result>
DEF STRING[<n>] <Output>
…
EXTOPEN(<Result>,<ExtDev>,<SyncMode>,<AccessMode>,<WriteMode>)
…
<Output>="Ausgabe Daten"
WRITE(<Result>,<ExtDev>,<Output>)
…
EXTCLOSE(<Result>,<ExtDev>)
```

## Bedeutung

| | | | |
|---|---|---|---|
| EXTOPEN: | Vordefinierte Prozedur zum Öffnen eines externen Geräts/Datei | | |
| <Result>: | **Parameter 1:** Ergebnisvariable<br>Anhand des Ergebnisvariablenwerts kann im Programm das Gelingen der Operation ausgewertet und entsprechend fortgefahren werden. | | |
| | Typ: | INT | |
| | Werte: | 0 | kein Fehler |
| | | 1 | externes Gerät kann nicht geöffnet werden |
| | | 2 | externes Gerät ist nicht projektiert |
| | | 3 | externes Gerät mit ungültigem Pfad projektiert |
| | | 4 | keine Zugriffsrechte für externes Gerät |
| | | 5 | Benutzungsmodus: externes Gerät bereits "exklusiv" belegt |
| | | 6 | Benutzungsmodus: externes Gerät bereits "shared" belegt |
| | | 7 | Dateilänge größer als LOCAL_DRIVE_MAX_FILESIZE |
| | | 8 | maximale Anzahl externer Geräte überschritten |
| | | 9 | Option für LOCAL_DRIVE nicht gesetzt |
| | | 11 | reserviert |
| | | 12 | Schreibmodus: Angabe widersprüchlich zu extdev.ini |
| | | 16 | ungültiger externer Pfad programmiert |
| | | 22 | externes Gerät nicht gemountet |

<table>
<tr><td><ExtDev>:</td><td colspan="2">Parameter 2: Symbolischer Bezeichner für das zu öffnende externe Gerät/Datei</td></tr>
<tr><td></td><td>Typ:</td><td>STRING</td></tr>
<tr><td></td><td colspan="2">Der symbolische Bezeichner besteht aus:<br>1. dem logischen Gerätenamen<br>2. ggf. gefolgt von einem Dateipfad (angehängt mit "/").<br>Folgende logische Gerätenamen sind definiert:</td></tr>
<tr><td></td><td>"LOCAL_DRIVE":</td><td>Lokale CompactFlash Card (vordefiniert)</td></tr>
<tr><td></td><td>"CYC_DRIVE":</td><td>reservierte Laufwerksangabe zur Verwendung in SIEMENS-Zyklen (vordefiniert)</td></tr>
<tr><td></td><td>"/dev/ext/1", ...<br>"/dev/ext/9":</td><td>Verfügbare Netzlaufwerke<br>Hinweis:<br>Projektierung in der Datei extdev.ini erforderlich!</td></tr>
<tr><td></td><td>"/dev/cyc/1",<br>"/dev/cyc/2":</td><td>reservierte Laufwerksangaben zur Verwendung in SIEMENS-Zyklen<br>Hinweis:<br>Projektierung in der Datei extdev.ini erforderlich!</td></tr>
<tr><td></td><td colspan="2">Dateipfad:<br>• Zu "LOCAL_DRIVE" und "CYC_DRIVE" muss ein Dateipfad angegeben werden, z. B.:<br>"LOCAL_DRIVE/my_dir/my_file.txt"<br>• Die logischen Gerätenamen "/dev/ext/1...9" und "/dev/cyc/1...2" können per Projektierung:<br>– schon auf eine Datei verweisen, dann darf nur der logische Gerätenamen angegeben werden, z. B.:<br>"/dev/ext/4"<br>– oder auf ein Verzeichnis, dann muss ein Dateipfad angegeben werden, z. B.:<br>"/dev/ext/5/my_dir/my_file.txt"</td></tr>
<tr><td></td><td colspan="2">Hinweis:<br>Für die logischen Gerätenamen "/dev/ext/1...9", "/dev/v24" und "/dev/cyc/1...2" wird Groß-/Kleinschreibung ignoriert, bei der Pfadangabe zu einer Datei ist Groß-/Kleinschreibung signifikant. Für "LOCAL_DRIVE" und "CYC_DRIVE" ist nur Großschreibung zulässig.</td></tr>
</table>

| | | | |
|---|---|---|---|
| <SyncMode>: | **Parameter 3:** Bearbeitungsmodus für die WRITE-Befehle zu diesem Gerät/Datei | | |
| | Typ: | STRING | |
| | Werte: | "SYN": | Synchrones Schreiben<br>Die Programmausführung wird angehalten, bis der Schreibvorgang abgeschlossen ist.<br>Die erfolgreiche Beendigung des synchronen Schreibens kann durch Auswerten der Fehlervariablen des WRITE-Befehls überprüft werden. |
| | | "ASYN": | Asynchrones Schreiben<br>Die Programmausführung wird durch den WRITE-Befehl nicht unterbrochen.<br>**Hinweis:**<br>Die Ergebnisvariable des WRITE-Befehls ist in diesem Modus nicht aussagekräftig und hat immer den Wert 0 (kein Fehler). Es gibt in diesem Modus keine Sicherheit, dass der WRITE-Befehl erfolgreich war. |
| <AccessMode>: | **Parameter 4:** Benutzungsmodus für dieses Gerät/Datei | | |
| | Typ: | STRING | |
| | Werte: | "SHARED": | Gerät/Datei wird im "Shared"-Modus angefordert. Andere Kanäle können das Gerät mitverwenden, d. h. ebenfalls in diesem Modus öffnen. |
| | | "EXCL": | Gerät/Datei wird in dem Kanal exklusiv verwendet, kein anderer Kanal kann das Gerät mitverwenden. |
| <WriteMode>: | **Parameter 5:** Schreibmodus für die WRITE-Befehle zu diesem Gerät/Datei (optional) | | |
| | Typ: | STRING | |
| | Werte: | "APP": | Anhängen<br>Die Datei bleibt in ihrem Inhalt erhalten, Schreibaufrufe fügen an das Ende an. |
| | | "OVR": | Überschreiben<br>Der Inhalt der Datei wird gelöscht und durch nachfolgende Schreibaufrufe neu erstellt. |
| | **Hinweis:**<br>Mit diesem Parameter kann der in der Datei extdev.ini projektierte Schreibmodus nicht überschrieben werden. Im Konfliktfall wird der EXTOPEN-Aufruf mit Fehler quittiert. | | |

| | |
|---|---|
| WRITE: | Vordefinierte Prozedur zum Schreiben der Ausgabedaten |

| | | | |
|---|---|---|---|
| EXTCLOSE: | Vordefinierte Prozedur zum Schließen eines geöffneten externen Geräts/Datei | | |
| <Result>: | **Parameter 1:** Ergebnisvariable | | |
| | Typ: | INT | |
| | Werte: | 0 | kein Fehler |
| | | 16 | ungültiger externer Pfad programmiert |
| | | 21 | Fehler beim Schließen des externen Geräts |
| <ExtDev>: | **Parameter 2:** Symbolischer Bezeichner für das zu schließende externe Gerät/ DateiBeschreibung siehe unter EXTOPEN!<br>**Hinweis:**<br>Der Bezeichner muss identisch zu dem im EXTOPEN-Aufruf angegebenen Bezeichner sein! | | |

## Beispiel

**Programmcode**

```
N10       DEF INT RESULT
N20       DEF BOOL EXTDEVICE
N30       DEF STRING[80] OUTPUT
N40       DEF INT PHASE
N50       EXTOPEN(RESULT,"LOCAL_DRIVE/my_file.txt","SYN","SHARED")
N60       IF RESULT > 0
N70          MSG("Fehler bei EXTOPEN:" << RESULT)
N80       ELSE
N90          EXTDEVICE=TRUE
N100      ENDIF
…
N200      PHASE=4
N210      IF EXTDEVICE
N220         OUTPUT=SPRINT("Ende Phase: %D",PHASE)
N230         WRITE(RESULT,"LOCAL_DRIVE/my_file.txt",OUTPUT)
N240      ENDIF
…
```

## Siehe auch

Stringoperationen (Seite 452)

Datei schreiben (WRITE) (Seite 574)

## 3.23.5 Alarme (SETAL)

In einem NC-Programm können Alarme gesetzt werden. Diese werden in der Bedienoberfläche in einem besonderen Feld dargestellt. Mit einem Alarm ist jeweils eine Reaktion der Steuerung entsprechend der Alarmkategorie verbunden.

## Syntax

```
SETAL(<Nr>[,<String>])
```

## Bedeutung

| | | | |
|---|---|---|---|
| SETAL(...): | Vordefinierte Prozedur zum Setzen eines Alarms<br>SETAL muss in einem eigenen NC-Satz programmiert werden. | | |
| <Nr>: | Alarmnummer | | |
| | Datentyp: | INT | |
| | Wertebereich: | 60000 ... 64999 (reserviert) | Alarme für SIEMENS-Zyklen |
| | | 65000 ... 69999 | Alarme für Anwenderzyklen |
| <String>: | Zeichenkette (optional) | | |
| | Datentyp: | STRING | |
| | Bei der Programmierung von Anwenderzyklenalarmen kann eine Zeichenkette mit bis zu 4 Parametern angegeben werden.<br>In diesen Parametern können variable Anwendertexte definiert werden.<br>Es stehen auch vordefinierte Parameter zur Verfügung: | | |
| | %1 | Kanalnummer | |
| | %2 | Satznummer, Label | |
| | %3 | Textindex für Zyklenalarme | |
| | %4 | zusätzlicher Alarmparameter | |

### Hinweis

Alarmtexte müssen in der Bedienoberfläche projektiert werden.

### Hinweis

Soll eine Alarmausgabe in der auf der Bedienoberfläche aktiven Sprache erfolgen, benötigt der Anwender Informationen über die aktuell auf dem HMI eingestellte Sprache. Diese Information kann im Teileprogramm und in Synchronaktionen über die Systemvariable $AN_LANGUAGE_ON_HMI (Seite 1326) abgefragt werden.

## Beispiel

```
Programmcode                          Kommentar
...
N100 SETAL(65000)                     ; Alarm Nr. 65000 setzen
...
```

## Weitere Informationen

### Alarmreaktion und -quittierung

Anwenderzyklenalarme sind Nummernbereichen zugeordnet, die sich bezüglich Alarmreaktion und -quittierung unterscheiden:

| Nummernbereich | Alarmreaktion | Alarmquittierung |
|---|---|---|
| 65000 - 65499 | Anzeige, Verriegelung NC-Start | Reset |
| 65500 - 65999 | Anzeige, Verriegelung NC-Start (nicht für ASUPs bei gesetztem MD20194) | Reset |
| 66000 - 66999 | Anzeige, Verriegelung NC-Start, Bewegungsstillstand nach Abarbeitung der vordecodierten Sätze | Reset |
| 67000 - 67999 | Anzeige | Cancel |
| 68000 - 68999 | Anzeige, Verriegelung NC-Start, sofortiger Interpolator-Stopp | Reset |
| 69000 - 69999 | Anzeige, Verriegelung NC-Start, Stopp am nächsten Satzende | Reset |

## 3.23.6 Rohteil definieren (WORKPIECE)

Die Steuerung muss Form und Größe eines Rohteils kennen, um es in der grafischen Simulation anzeigen zu können. Der Anwender hat daher die Möglichkeit, Rohteile über die Bedienoberfläche oder direkt im NC-Programm zu definieren. Die Rohteildefinitionen bleiben über (Programmende-/Kanal-/BAG-)Reset hinaus erhalten. Mit dem nächsten Hochlauf der Steuerung werden sie automatisch gelöscht.

## Syntax

```
WORKPIECE("<WP>", "<RefP>", "<ZeroOffset>", "<Type>", <Par5>,
<Par6>, ..., <Par12>)
```

## Bedeutung

| | | | |
|---|---|---|---|
| `WORKPIECE(...)`: | | Vordefinierte Prozedur zum Definieren eines Rohteils | |
| | | Vorlaufstopp: | ja |
| | | Alleine im Satz: | ja |
| **Parameter:** | | | |
| 1 | `"<WP>"`: | Name des Werkstücks (optional) | |
| | | Datentyp: | STRING |
| | | Eine Angabe ist nur notwendig, wenn es in einem Kanal mehrere Werkstücke geben kann. Ohne Angabe wird automatisch "WORKP<n>" angenommen, wobei <n> die Nummer des deklarierenden Kanals ist. | |

| | | | | |
|---|---|---|---|---|
| 2 | "<RefP>": | Aufspannung (optional; nur bei Fräsmaschinen) | | |
| | | Datentyp: | STRING | |
| | | Wertebereich: | "Tisch" | Aufspannung auf dem festen Tisch |
| | | | "A" | Aufspannung auf Rundachse A |
| | | | "B" | Aufspannung auf Rundachse B |
| | | | "C" | Aufspannung auf Rundachse C |
| | | **Voraussetzung:**<br>Der Tisch bzw. die Rundachse muss über das entsprechende Maschinendatum für die Rohteilaufspannung freigeschaltet sein (siehe Inbetriebnahmehandbuch SINUMERIK Operate). | | |
| 3 | "<ZeroOffset>": | Einstellbare Nullpunktverschiebung zur Platzierung des Rohteils (nicht programmierbar)<br>Die Auswahl einer einstellbaren Nullpunktverschiebung zur Platzierung des Rohteils wird nur bei der Rohteileingabe über die Bedienoberfläche angeboten. Bei der direkten Rohteildefinition im Teileprogramm bezieht sich das Rohteil dagegen immer auf die aktuell gültige Nullpunktverschiebung. | | |
| 4 | "<Type>": | Rohteilform | | |
| | | Datentyp: | STRING | |
| | | Wertebereich: | "CYLINDER": | Zylinder |
| | | | "PIPE": | Rohr |
| | | | "RECTANGLE": | Quader mittig |
| | | | "BOX": | Quader |
| | | | "N_CORNER": | N-Eck |
| 5 ... 12 | <Par5> ... <Par12>: | Parameter zur Beschreibung der Rohteilform | | |
| | | Datentyp: | REAL | |
| | | Die Anzahl der notwendigen Parameter und deren Bedeutung sind von der jeweiligen Rohteilform und vom Wert des Bitparameters abhängig.<br>Siehe:<br>• Tabelle "Parameter zur Beschreibung der Rohteilform"<br>• Tabelle "Bitparameter" | | |
| | | | | |
| WORPIECE(): | | Ein WORKPIECE-Aufruf ohne Parameter löscht alle Rohteildefinitionen. | | |
| WORPIECE(<WP>): | | Ein WORKPIECE-Aufruf mit Werkstücknamen löscht nur diese Rohteildefinition. | | |

Tabelle 3-6 Parameter zur Beschreibung der Rohteilform

| Rohteilform | Parameter | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | <Par5> | <Par6> | <Par7> | <Par8> | <Par9> | <Par10> | <Par11> | <Par12> |
| Zylinder | Bitparameter<br>Realwert, der als bitcodierte Integerzahl interpretiert wird. Die Bits definieren die Bedeutung der folgenden Parameter (siehe Tabelle "Bitparameter"). | Bezugspunkt $Z_0$ | Länge $Z_1$ | Bearbeitungsmaß $Z_B$ | Außendurchmesser $d_0$ | - | Drehung um Rundachse | - |
| Rohr | | Bezugspunkt $Z_0$ | Länge $Z_1$ | Bearbeitungsmaß $Z_B$ | Außendurchmesser $d_0$ | Wandstärke (ink) / Innendurchmesser $d_1$ (abs) | Drehung um Rundachse | - |
| Quader mittig | | Bezugspunkt $Z_0$ | Länge $Z_1$ | Bearbeitungsmaß $Z_B$ | Breite W | Länge L | Drehung um Rundachse | - |
| Quader | | Bezugspunkt $Z_0$ | Länge $Z_1$ | Bearbeitungsmaß $Z_B$ | $X_0$ | $Y_0$ | $X_1$ | $Y_1$ |
| N-Eck | | Bezugspunkt $Z_0$ | Länge $Z_1$ | Bearbeitungsmaß $Z_B$ | Anzahl Ecken | Schlüsselweite | Drehung um Rundachse | - |

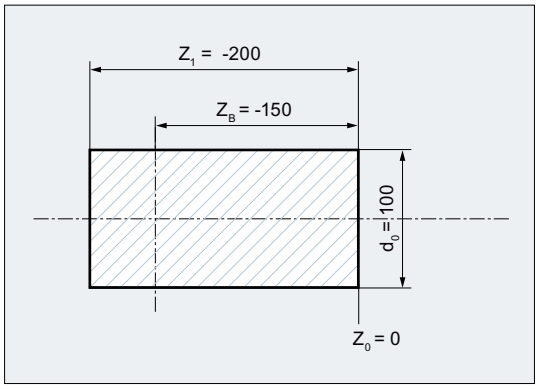
Tabelle 3-7 Bitparameter

| Bit | Bedeutung | |
|---|---|---|
| 4 (0x0010) | Quader: $X_1$ | |
| | = 0 | ink |
| | = 1 | abs |
| 5 (0x0020) | Quader: $Y_1$ | |
| | = 0 | ink |
| | = 1 | abs |
| 6 (0x0040) | Länge $Z_1$ (Endmaß) | |
| | = 0 | ink |
| | = 1 | abs |
| Bit 7 (0x0080) | Bearbeitungsmaß $Z_B$ | |
| | = 0 | ink |
| | = 1 | abs |
| Bit 8 (0x0100) | Rohr: Wandstärke / Innendurchmesser | |
| | = 0 | ink |
| | = 1 | abs |
| 9 (0x0200) | N-Eck | |
| | = 0 | Schlüsselweite |
| | = 1 | Kantenlänge |
| 12 (0x1000) | Aufspannung für Drehmaschinen | |
| | = 0 | Hauptspindel |
| | = 1 | Gegenspindel |

| Bit | Bedeutung | |
|---|---|---|
| 13 (0x2000) | Gegenspindel | |
| | = 0 | mit Spiegelung |
| | = 1 | ohne Spiegelung |

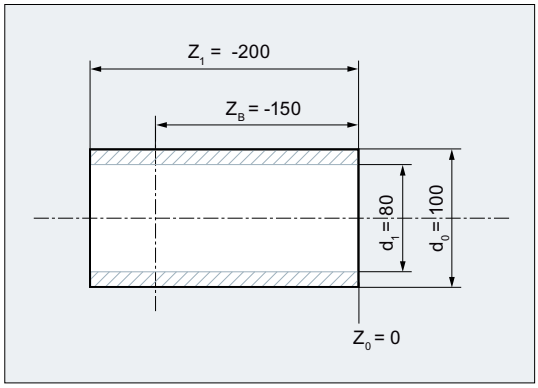
## Beispiele

### Beispiel 1: Zylinderförmiges Rohteil auf einer Drehmaschine




| Programmcode | Kommentar |
|---|---|
| ... | |
| WORKPIECE(,,,"CYLINDER",0,0,-200,-150,100) | ; **Rohteildefinition:**<br>; Rohteilform: Zylinder<br>; Bitparameter=0(kein Bit gesetzt) → Werte für Länge und Bearbeitungsmaß sind inkrementell, Rohteil auf Hauptspindel<br>; Bezugspunkt(Z0)=0<br>; Länge(Z1)=-200<br>; Bearbeitungsmaß(ZB)=-150<br>; Außendurchmesser(d0)=100 |
| ... | |

**Beispiel 2: Röhrenförmiges Rohteil auf einer Drehmaschine**




| Programmcode | Kommentar |
|---|---|
| ... | |
| WORKPIECE(,,,"PIPE",256,0,-200,-150,100,80) | ; **Rohteildefinition:** |
| | ; Rohteilform: Rohr |
| | ; Bitparameter=256(Bit8=1) → Innendurchmesser ist absolut, Länge und Bearbeitungsmaß sind inkrementell, Rohteil auf Hauptspindel |
| | ; Bezugspunkt(Z0)=0 |
| | ; Länge(Z1)=-200 |
| | ; Bearbeitungsmaß(ZB)=-150 |
| | ; Außendurchmesser(d0)=100 |
| | ; Innendurchmesser(d1)=80 |
| ... | |

## 3.23.7 Sprachmodus umschalten (G290, G291)

Die Steuerung bietet die Möglichkeit, Teileprogramme von externen CNC-Systemen einzulesen und abzuarbeiten. Voraussetzung ist, dass der entsprechende NC-Sprachmodus (ISO-Dialekt) bei der Inbetriebnahme festgelegt wurde.

Der ISO-Dialekt-Modus kann getrennt für jeden Kanal aktiviert werden. Zum Beispiel kann Kanal 1 im ISO-Dialekt-Modus laufen, während Kanal 2 im SINUMERIK-Modus aktiv ist.

Das Umschalten zwischen SINUMERIK-Modus und ISO-Dialekt-Modus erfolgt im NC-Programm über die Befehle der G-Gruppe 47. Das aktive Werkzeug, die Werkzeugkorrekturen und Nullpunktverschiebungen werden durch das Umschalten nicht beeinflusst.

### Syntax

```
G291
...
G290
```

## Bedeutung

| | | |
|---|---|---|
| G290: | SINUMERIK-Sprachmodus aktivieren | |
| | Alleine im Satz: | ja |
| | Wirksamkeit: | modal |
| G291: | ISO-Sprachmodus aktivieren | |
| | Alleine im Satz: | ja |
| | Wirksamkeit: | modal |

## Bedingungen

### SINUMERIK-Modus

- Die Voreinstellung der G-Befehle kann für jeden Kanal über ein Maschinendatum festgelegt werden.
- Im SINUMERIK-Modus können keine Sprachbefehle aus den ISO-Dialekten programmiert werden.

### ISO-Dialekt-Modus

- Der ISO-Dialekt-Modus als Grundeinstellung der Steuerung kann mit Maschinendaten eingestellt werden. Die Steuerung fährt danach standardmäßig im ISO-Dialekt-Modus hoch.
- Es können nur G-Befehle aus dem ISO-Dialekt programmiert werden. Die Programmierung von SINUMERIK-Funktionen ist im ISO-Dialekt-Modus nicht möglich.
- Eine Mischung aus ISO-Dialekt- und SINUMERIK-Sprache im selben NC-Satz ist nicht möglich.
- Eine Umschaltung über G-Befehle zwischen ISO-Dialekt-M (Fräsen) und ISO-Dialekt-T (Drehen) ist nicht möglich.
- Es können Unterprogramme aufgerufen werden, die im SINUMERIK-Modus programmiert sind.
- Wenn SINUMERIK-Funktionen verwendet werden sollen, muss zuerst in den SINUMERIK-Modus umgeschaltet werden (siehe Beispiel).

## Beispiel

### Kompression von Linearsätzen im ISO-Dialekt-Modus

```
Programmcode                    Kommentar
N5 G290                         ; SINUMERIK-Sprachmodus aktivieren.
N10 COMPCAD                     ; COMPCAD ist ein Befehl der Siemens-Spra-
                                che, der eine Kompressorfunktion akti-
                                viert, die aufeinanderfolgende Linearsät-
                                ze durch Polynom-Sätze mit möglichst gro-
                                ßen Bahnlängen ersetzt.
N15 G291                        ; ISO-Sprachmodus aktivieren.
```

| **Programmcode** | **Kommentar** |
|---|---|
| N20 G01 X100 Y100 F1000 | ; Da COMPCAD im SINUMERIK-Modus aktiviert wurde, können mit dieser Funktion auch Linearsätze im ISO-Dialekt-Modus komprimiert werden. |
| ... | |

## Weitere Informationen

Funktionshandbuch ISO-Dialekte

# 3.24 Eigene Abspanprogramme

## 3.24.1 Unterstützende Funktionen für das Abspanen

Für das Abspanen werden Ihnen fertige Bearbeitungszyklen angeboten. Darüber hinaus haben Sie die Möglichkeit, mit den nachfolgend aufgeführten Funktionen eigene Abspanprogramme zu erstellen:

- Konturtabelle erstellen (CONTPRON)
- Codierte Konturtabelle erstellen (CONTDCON)
- Konturaufbereitung ausschalten (EXECUTE)
- Schnittpunkt zwischen zwei Konturelementen ermitteln (INTERSEC)
  (Nur für Tabellen, die durch CONTPRON erstellt wurden.)
- Konturelemente einer Tabelle satzweise abarbeiten (EXECTAB)
  (Nur für Tabellen, die durch CONTPRON erstellt wurden.)
- Kreisdaten berechnen (CALCDAT)

**Hinweis**

Sie können diese Funktionen nicht nur zum Abspanen, sondern universell einsetzen.

### Voraussetzungen

Vor dem Aufruf der Funktionen CONTPRON oder CONTDCON müssen:

- ein Startpunkt angefahren werden, der eine kollisionsfreie Bearbeitung erlaubt.
- die Schneidenradiuskorrektur mit `G40` ausgeschaltet sein.

## 3.24.2 Konturtabelle erstellen (CONTPRON)

Mit `CONTPRON` schalten Sie die Konturaufbereitung ein. Die nachfolgend aufgerufenen NC-Sätze werden nicht abgearbeitet, sondern in einzelne Bewegungen aufgeteilt und in der Konturtabelle abgelegt. Jedem Konturelement entspricht eine Tabellenzeile im zweidimensionalen Feld der Konturtabelle. Die Anzahl der ermittelten Hinterschnitte wird zurückgeliefert.

### Syntax

Konturaufbereitung einschalten:
```
CONTPRON(<Konturtabelle>,<Bearbeitungsart>,<Hinterschnitte>,
<Bearbeitungsrichtung>)
```

Konturaufbereitung ausschalten und in den normalen Abarbeitungsmodus zurückkehren:
```
EXECUTE(<FEHLER>)
```

Siehe " Konturaufbereitung ausschalten (EXECUTE) (Seite 1040) "

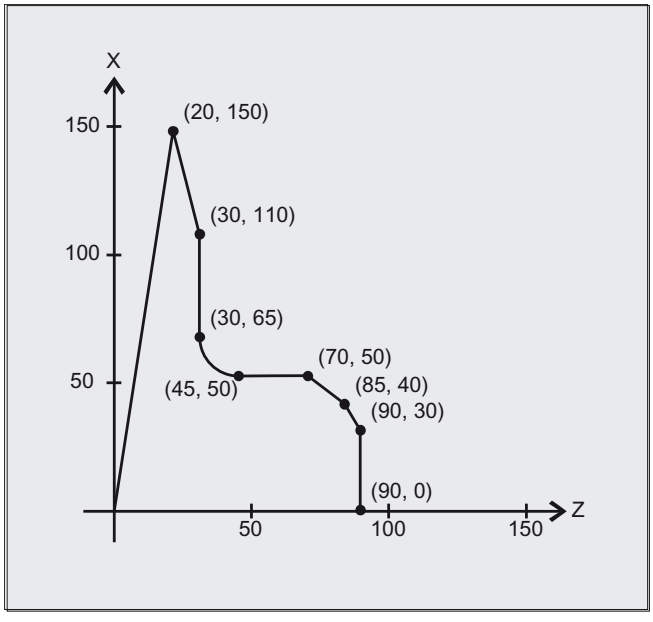
## Bedeutung

| | | | |
|---|---|---|---|
| CONTPRON: | Vordefinierte Prozedur zum Einschalten der Konturaufbereitung zur Erstellung einer Konturtabelle | | |
| <Konturtabelle>: | Name der Konturtabelle | | |
| <Bearbeitungsart>: | Parameter für die Bearbeitungsart | | |
| | Typ: | CHAR | |
| | Wert: | "G": | Längsdrehen: Innenbearbeitung |
| | | "L": | Längsdrehen: Außenbearbeitung |
| | | "N": | Plandrehen: Innenbearbeitung |
| | | "P": | Plandrehen: Außenbearbeitung |
| <Hinterschnitte>: | Ergebnisvariable für die Anzahl auftretender Hinterschnittelemente | | |
| | Typ: | INT | |
| <Bearbeitungsrichtung>: | Parameter für die Bearbeitungsrichtung | | |
| | Typ: | INT | |
| | Wert: | 0 | Konturaufbereitung vorwärts (Standardwert) |
| | | 1 | Konturaufbereitung in beiden Richtungen |

## Beispiel 1

Erstellen einer Konturtabelle mit:

- Namen "KTAB"
- max. 30 Konturelementen (Kreise, Geraden)
- einer Variablen für die Anzahl auftretender Hinterschnittelemente
- einer Variablen für Fehlermeldungen



### NC-Programm:

```
Programmcode                                    Kommentar
N10 DEF REAL KTAB[30,11]                        ; Konturtabelle mit Namen KTAB und max.
                                                30 Konturelementen, Parameterwert 11
                                                (Spaltenzahl der Tabelle) ist eine feste
                                                Größe.
N20 DEF INT ANZHINT                             ; Variable für die Anzahl der Hinter-
                                                schnittelemente mit Namen ANZHINT.
N30 DEF INT FEHLER                              ; Variable für die Fehlerrückmeldung
                                                (0=kein Fehler, 1=Fehler).
N40 G18
N50 CONTPRON(KTAB,"G",ANZHINT)                  ; Konturaufbereitung einschalten.
N60 G1 X150 Z20                                 ; N60 bis N120: Konturbeschreibung
N70 X110 Z30
N80 X50 RND=15
N90 Z70
N100 X40 Z85
N110 X30 Z90
N120 X0
N130 EXECUTE(FEHLER)                            ; Füllen der Konturtabelle beenden, Um-
                                                schalten auf normalen Programmbetrieb.
N140 …                                          ; Weitere Bearbeitung der Tabelle.
```

### Konturtabelle KTAB:

| Index Zeile | Spalte | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| (0) | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) | (10) |
| 7 | 7 | 11 | 0 | 0 | 20 | 150 | 0 | 82.40535663 | 0 | 0 |
| 0 | 2 | 11 | 20 | 150 | 30 | 110 | -1111 | 104.0362435 | 0 | 0 |
| 1 | 3 | 11 | 30 | 110 | 30 | 65 | 0 | 90 | 0 | 0 |
| 2 | 4 | 13 | 30 | 65 | 45 | 50 | 0 | 180 | 45 | 65 |
| 3 | 5 | 11 | 45 | 50 | 70 | 50 | 0 | 0 | 0 | 0 |
| 4 | 6 | 11 | 70 | 50 | 85 | 40 | 0 | 146.3099325 | 0 | 0 |
| 5 | 7 | 11 | 85 | 40 | 90 | 30 | 0 | 116.5650512 | 0 | 0 |
| 6 | 0 | 11 | 90 | 30 | 90 | 0 | 0 | 90 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### Erläuterung der Spalteninhalte:

(0) Zeiger auf nächstes Konturelement (auf die Zeilennummer desselben)

(1) Zeiger auf vorhergehendes Konturelement

(2) Codierung des Konturmodus für die Bewegung
Mögliche Werte für X = abc

| | | | | |
|---|---|---|---|---|
| a = $10^2$ | G90 = 0 | G91 = 1 | | |
| b = $10^1$ | G70 = 0 | G71 = 1 | | |
| c = $10^0$ | G0 = 0 | G1 = 1 | G2 = 2 | G3 = 3 |

(3), (4) Anfangspunkt der Konturelemente
(3) = Abszisse, (4) = Ordinate in der aktuellen Ebene

(5), (6) Endpunkt der Konturelemente
(5) = Abszisse, (6) = Ordinate in der aktuellen Ebene

(7) Max-/min-Anzeiger: kennzeichnet lokale Maxima und Minima in der Kontur

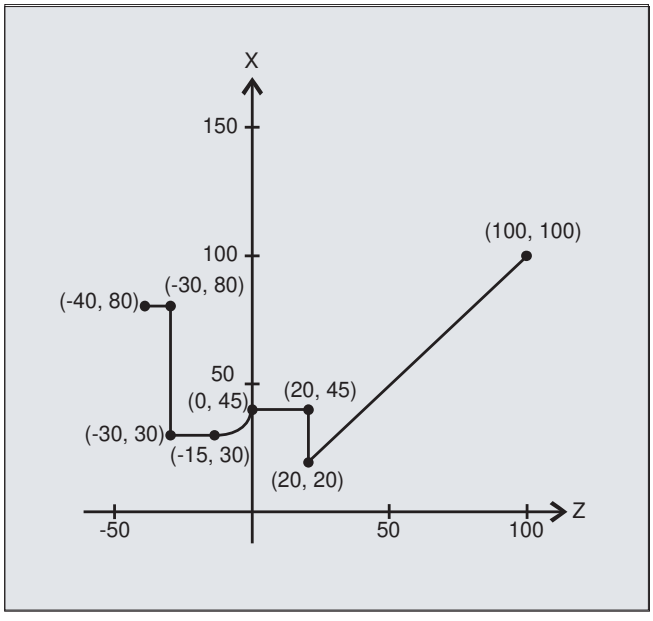
(8) Maximaler Wert zwischen Konturelement und Abszisse (bei Längsbearbeitung) bzw. Ordinate (bei Planbearbeitung). Der Winkel ist abhängig von der programmierten Bearbeitungsart.

(9), (10) Mittelpunktskoordinaten des Konturelements, wenn es ein Kreissatz ist.
(9) = Abszisse, (10) = Ordinate

## Beispiel 2

Erstellen einer Konturtabelle mit

- Namen KTAB
- max. 92 Konturelementen (Kreise, Geraden)
- Betriebsart: Längsdrehen, Außenbearbeitung
- Aufbereitung vorwärts und rückwärts



### NC-Programm:

```
Programmcode                                    Kommentar
N10 DEF REAL KTAB[92,11]                        ; Konturtabelle mit Namen KTAB und max.
                                                92 Konturelementen, Parameterwert 11 ist
                                                eine feste Größe.
N20 DEF CHAR BT="L"                             ; Betriebsart für CONTPRON: Längsdrehen,
                                                Außenbearbeitung
N30 DEF INT HE=0                                ; Anzahl der Hinterschnittelemente=0
N40 DEF INT MODE=1                              ; Aufbereitung vorwärts und rückwärts
N50 DEF INT ERR=0                               ; Fehlerrückmeldung
...
N100 G18 X100 Z100 F1000
N105 CONTPRON(KTAB,BT,HE,MODE)                  ; Konturaufbereitung einschalten.
N110 G1 G90 Z20 X20
N120 X45
N130 Z0
N140 G2 Z-15 X30 K=AC(-15) I=AC(45)
N150 G1 Z-30
N160 X80
N170 Z-40
N180 EXECUTE(ERR)                               ; Füllen der Konturtabelle beenden, Um-
                                                schalten auf normalen Programmbetrieb.
...
```

### Konturtabelle KTAB:

Nach Ende der Konturaufbereitung steht die Kontur in beiden Richtungen zur Verfügung.

| Index | Spalte | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Zeile | (0) | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) | (10) |
| 0 | 6[1)] | 7[2)] | 11 | 100 | 100 | 20 | 20 | 0 | 45 | 0 | 0 |
| 1 | 0[3)] | 2 | 11 | 20 | 20 | 20 | 45 | -3 | 90 | 0 | 0 |
| 2 | 1 | 3 | 11 | 20 | 45 | 0 | 45 | 0 | 0 | 0 | 0 |
| 3 | 2 | 4 | 12 | 0 | 45 | -15 | 30 | 5 | 90 | -15 | 45 |
| 4 | 3 | 5 | 11 | -15 | 30 | -30 | 30 | 0 | 0 | 0 | 0 |
| 5 | 4 | 7 | 11 | -30 | 30 | -30 | 45 | -1111 | 90 | 0 | 0 |
| 6 | 7 | 0[4)] | 11 | -30 | 80 | -40 | 80 | 0 | 0 | 0 | 0 |
| 7 | 5 | 6 | 11 | -30 | 45 | -30 | 80 | 0 | 90 | 0 | 0 |
| 8 | 1[5)] | 2[6)] | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ... | | | | | | | | | | |
| 83 | 84 | 0[7)] | 11 | 20 | 45 | 20 | 80 | 0 | 90 | 0 | 0 |
| 84 | 90 | 83 | 11 | 20 | 20 | 20 | 45 | -1111 | 90 | 0 | 0 |
| 85 | 0[8)] | 86 | 11 | -40 | 80 | -30 | 80 | 0 | 0 | 0 | 0 |
| 86 | 85 | 87 | 11 | -30 | 80 | -30 | 30 | 88 | 90 | 0 | 0 |
| 87 | 86 | 88 | 11 | -30 | 30 | -15 | 30 | 0 | 0 | 0 | 0 |
| 88 | 87 | 89 | 13 | -15 | 30 | 0 | 45 | -90 | 90 | -15 | 45 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 89 | 88 | 90 | 11 | 0 | 45 | 20 | 45 | 0 | 0 | 0 | 0 |
| 90 | 89 | 84 | 11 | 20 | 45 | 20 | 20 | 84 | 90 | 0 | 0 |
| 91 | 83[9)] | 85[10)] | 11 | 20 | 20 | 100 | 100 | 0 | 45 | 0 | 0 |

**Erläuterung der Spalteninhalte und der Anmerkungen zu den Zeilen 0, 1, 6, 8, 83, 85 und 91**

Es gelten die im Beispiel 1 genannten Erläuterungen der Spalteninhalte.

**Immer in Tabellen-Zeile 0:**

1) Vorgänger: Zeile n enthält das Konturende vorwärts

2) Nachfolger: Zeile n ist das Konturtabellenende vorwärts

**Je einmal innerhalb der Konturelemente vorwärts:**

3) Vorgänger: Konturbeginn (vorwärts)

4) Nachfolger: Konturende (vorwärts)

**Immer auf Zeile Konturtabellenende (vorwärts) +1:**

5) Vorgänger: Anzahl der Hinterschnitte vorwärts

6) Nachfolger: Anzahl der Hinterschnitte rückwärts

**Je einmal innerhalb der Konturelemente rückwärts:**

7) Nachfolger: Konturende (rückwärts)

8) Vorgänger: Konturbeginn (rückwärts)

**Immer in letzter Tabellen-Zeile:**

9) Vorgänger: Zeile n ist der Konturtabellenanfang (rückwärts)

10) Nachfolger: Zeile n enthält den Konturanfang (rückwärts)

## Weitere Informationen

**Erlaubte Verfahrbefehle, Koordinatensystem**

Für die Konturprogrammierung sind folgende G-Befehle zulässig:

- G-Gruppe 1: `G0`, `G1`, `G2`, `G3`

Zusätzlich möglich sind:

- Rundung und Fase
- Kreisprogrammierung über `CIP` und `CT`

Die Funktionen Spline, Polynom und Gewinde führen zu Fehlern.

Änderungen des Koordinatensystems durch Einschalten eines Frames sind zwischen `CONTPRON` und `EXECUTE` nicht zulässig. Gleiches gilt für einen Wechsel zwischen `G70` und `G71` bzw. `G700` und `G710`.

Ein Tausch der Geometrieachsen mit `GEOAX` während der Aufbereitung der Konturtabelle führt zu einem Alarm.

#### Hinterschnittelemente

Die Konturbeschreibung der einzelnen Hinterschnittelemente kann wahlweise in einem Unterprogramm oder in einzelnen Sätzen erfolgen.

#### Abspanen unabhängig von der programmierten Konturrichtung

Die Konturaufbereitung mit CONTPRON wurde so erweitert, dass nach ihrem Aufruf die Konturtabelle unabhängig von der programmierten Richtung zur Verfügung steht.

## 3.24.3 Codierte Konturtabelle erstellen (CONTDCON)

Bei der mit CONTDCON eingeschalteten Konturaufbereitung werden die nachfolgend aufgerufenen NC-Sätze in einer 6-spaltigen Konturtabelle speichergünstig codiert abgelegt. Jedem Konturelement entspricht eine Tabellenzeile in der Konturtabelle. Aus Kenntnis der unten angegebenen Codierungsregeln können Sie z. B. für Zyklen aus den Tabellenzeilen DIN-Code-Programme zusammenstellen. In der Tabellenzeile mit der Nummer 0 werden die Daten des Ausgangspunkts gespeichert.

### Syntax

Konturaufbereitung einschalten:
`CONTDCON(<Konturtabelle>,<Bearbeitungsrichtung>)`

Konturaufbereitung ausschalten und in den normalen Abarbeitungsmodus zurückkehren:
`EXECUTE(<FEHLER>)`

Siehe " Konturaufbereitung ausschalten (EXECUTE) (Seite 1040) "

### Bedeutung

| | | | |
|---|---|---|---|
| CONTDCON: | Vordefinierte Prozedur zum Einschalten der Konturaufbereitung zur Erstellung einer codierten Konturtabelle | | |
| <Konturtabelle>: | Name der Konturtabelle | | |
| <Bearbeitungsrichtung>: | Parameter für Bearbeitungsrichtung | | |
| | Typ: | INT | |
| | Wert: | 0 | Konturaufbereitung gemäß der Folge der Kontursätze (Standardwert) |
| | | 1 | unzulässig |

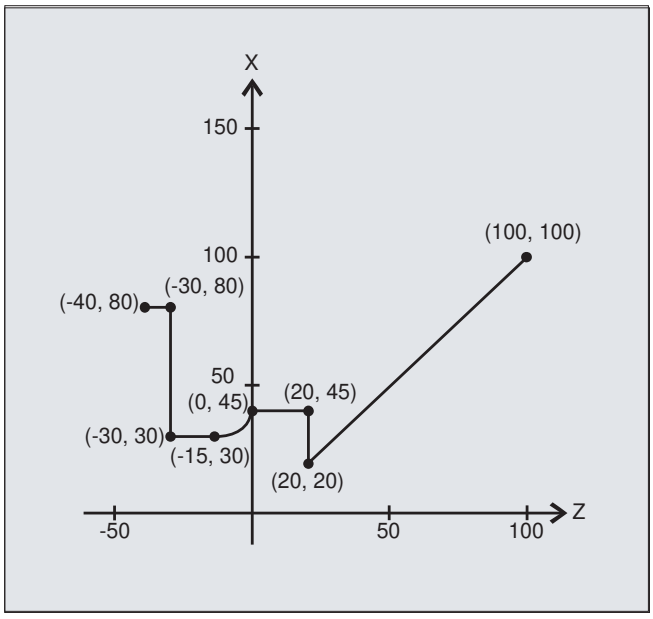
#### Hinweis

Die für CONTDCON zugelassenen G-Befehle im zu tabellierenden Programmstück sind umfangreicher als bei CONTPRON. Darüber hinaus werden Vorschübe und Vorschubtyp pro Konturstück mitgespeichert.

## Beispiel

Erstellen einer Konturtabelle mit:

- Namen "KTAB"
- Konturelementen (Kreise, Geraden)
- Betriebsart: Drehen
- Bearbeitungsrichtung: vorwärts




### NC-Programm:

| **Programmcode** | **Kommentar** |
|---|---|
| N10 DEF REAL KTAB[9,6] | ; Konturtabelle mit Namen KTAB und 9 Tabellenzeilen. Diese erlauben 8 Konturtsätze. Der Parameterwert 6 (Spaltenzahl der Tabelle) ist eine feste Größe. |
| N20 DEF INT MODE = 0 | ; Variable für die Bearbeitungsrichtung. Standardwert 0: nur in programmierter Richtung der Kontur. |
| N30 DEF INT ERROR = 0 | ; Variable für die Fehlerrückmeldung. |
| ... | |
| N100 G18 G64 G90 G94 G710 | |
| N101 G1 Z100 X100 F1000 | |
| N105 CONTDCON (KTAB, MODE) | ; Aufruf Konturaufbereitung (MODE darf weggelassen werden). |
| N110 G1 Z20 X20 F200 | ; Konturbeschreibung. |
| N120 G9 X45 F300 | |
| N130 Z0 F400 | |
| N140 G2 Z-15 X30 K=AC(-15) I=AC(45)F100 | |

```
Programmcode                                    Kommentar
N150 G64 Z-30 F600
N160 X80 F700
N170 Z-40 F800
N180 EXECUTE(ERROR)                             ; Füllen der Konturtabelle beenden,
                                                Umschalten auf normalen Programmbe-
                                                trieb.
...
```

**Konturtabelle KTAB:**

| | Spaltenindex | | | | | |
|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 |
| Zeilenindex | Konturmodus | Endpunkt Abszisse | Endpunkt Ordinate | Mittelpunkt Abszisse | Mittelpunkt Ordinate | Vorschub |
| 0 | 30 | 100 | 100 | 0 | 0 | 7 |
| 1 | 11031 | 20 | 20 | 0 | 0 | 200 |
| 2 | 111031 | 20 | 45 | 0 | 0 | 300 |
| 3 | 11031 | 0 | 45 | 0 | 0 | 400 |
| 4 | 11032 | -15 | 30 | -15 | 45 | 100 |
| 5 | 11031 | -30 | 30 | 0 | 0 | 600 |
| 6 | 11031 | -30 | 80 | 0 | 0 | 700 |
| 7 | 11031 | -40 | 80 | 0 | 0 | 800 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 |

**Erläuterung der Spalteninhalte:**

Zeile 0: Codierungen für den **Startpunkt:**

Spalte 0: $10^0$ (Einerstelle): G0 = 0
$10^1$ (Zehnerstelle): G70 = 0, G71 = 1, G700 = 2, G710 = 3
Spalte 1: Startpunkt Abszisse
Spalte 2: Startpunkt Ordinate
Spalte 3-4: 0
Spalte 5: Zeilenindex des letzten Konturstückes in der Tabelle

Zeilen 1-n: Einträge der **Konturstücke**

Spalte 0: $10^0$ (Einerstelle): G0 = 0, G1 = 1, G2 = 2, G3 = 3
$10^1$ (Zehnerstelle): G70 = 0, G71 = 1, G700 = 2, G710 = 3
$10^2$ (Hunderterstelle): G90 = 0, G91 = 1
$10^3$ (Tausenderstelle): G93 = 0, G94 = 1, G95 = 2, G96 = 3
$10^4$ (Zehntausenderstelle): G60 = 0, G44 = 1, G641 = 2, G642 = 3
$10^5$ (Hunderttausender Stelle): G9 = 1
Spalte 1: Endpunkt Abszisse
Spalte 2: Endpunkt Ordinate
Spalte 3: Mittelpunkt Abszisse bei Kreisinterpolation

Spalte 4: Mittelpunkt Ordinate bei Kreisinterpolation
Spalte 5: Vorschub

## Weitere Informationen

### Erlaubte Verfahrbefehle, Koordinatensystem

Für die Konturprogrammierung sind folgende G-Gruppen und G-Befehle zulässig:

G-Gruppe 1: `G0`, `G1`, `G2`, `G3`
G-Gruppe 10: `G60`, `G64`, `G641`, `G642`
G-Gruppe 11: `G9`
G-Gruppe 13: `G70`, `G71`, `G700`, `G710`
G-Gruppe 14: `G90`, `G91`
G-Gruppe 15: `G93`, `G94`, `G95`, `G96`, `G961`

Zusätzlich möglich sind:

- Rundung und Fase
- Kreisprogrammierung über `CIP` und `CT`

Die Funktionen Spline, Polynom und Gewinde führen zu Fehlern.

Änderungen des Koordinatensystems durch Einschalten eines Frames sind zwischen `CONTDCON` und `EXECUTE` nicht zulässig. Gleiches gilt für einen Wechsel zwischen `G70` und `G71` bzw. `G700` und `G710`.

Ein Tausch der Geometrieachsen mit `GEOAX` während der Aufbereitung der Konturtabelle führt zu einem Alarm.

### Bearbeitungsrichtung

Die mit `CONTDCON` erzeugte Konturtabelle ist zum Abspanen in der programmierten Richtung der Kontur vorgesehen.

## 3.24.4 Schnittpunkt zwischen zwei Konturelementen ermitteln (INTERSEC)

`INTERSEC` ermittelt den Schnittpunkt von zwei normierten Konturelementen aus mit `CONTPRON` erzeugten Konturtabellen.

## Syntax

```
<Status>=INTERSEC(<Konturtabelle_1>[<Konturelement_1>],
<Konturtabelle_2>[<Konturelement_2>],<Schnittpunkt>,<Bearbeitungsart
>)
```

## Bedeutung

| | | | |
|---|---|---|---|
| INTERSEC: | Vordefinierte Funktion zur Ermittlung des Schnittpunkts zweier Konturelemente aus mit CONTPRON erzeugten Konturtabellen | | |
| <Status>: | Variable für den Schnittpunktstatus | | |
| | Typ: | BOOL | |
| | Wert: | TRUE | Schnittpunkt gefunden |
| | | FALSE | kein Schnittpunkt gefunden |
| <Konturtabelle_1>: | Name der ersten Konturtabelle | | |
| <Konturelement_1>: | Nummer des Konturelements der ersten Konturtabelle | | |
| <Konturtabelle_2>: | Name der zweiten Konturtabelle | | |
| <Konturelement_2>: | Nummer des Konturelements der zweiten Konturtabelle | | |
| <Schnittpunkt>: | Schnittpunkt-Koordinaten in der aktiven Ebene (G17 / G18 / G19) | | |
| | Typ: | REAL | |
| <Bearbeitungsart>: | Parameter für die Bearbeitungsart | | |
| | Typ: | INT | |
| | Wert: | 0 | Schnittpunktberechnung in der mit Parameter 2 aktiven Ebene (Standardwert) |
| | | 1 | Schnittpunktberechnung unabhängig der übergebenen Ebene |

### Hinweis

Beachten Sie, dass die Variablen vor ihrer Verwendung definiert sein müssen.

Die Übergabe der Konturen erfordert die Einhaltung der mit CONTPRON definierten Werte:

| Parameter | Bedeutung |
|---|---|
| 2 | Codierung des Kontur-Mode für die Bewegung |
| 3 | Kontur-Anfangpunkt Abszisse |
| 4 | Kontur-Anfangpunkt Ordinate |
| 5 | Kontur-Endpunkt Abszisse |
| 6 | Kontur-Endpunkt Ordinate |
| 9 | Mittelpunkskoordinate für die Abszisse (nur bei Kreis-Kontur) |
| 10 | Mittelpunkskoordinate für die Ordinate (nur bei Kreis-Kontur) |

## Beispiel

Schnittpunkt von Konturelement 3 der Tabelle TABNAME1 und Konturelement 7 der Tabelle TABNAME2 ermitteln. Die Schnittpunkt-Koordinaten in der aktiven Ebene werden in der Variablen ISCOORD (1. Element = Abszisse, 2. Element = Ordinate) abgelegt. Existiert kein Schnittpunkt, erfolgt ein Sprung zu KEINSCH (kein Schnittpunkt gefunden).

```
Programmcode                              Kommentar
DEF REAL TABNAME1[12,11]                  ; Konturtabelle 1
DEF REAL TABNAME2[10,11]                  ; Konturtabelle 2
```

| Programmcode | Kommentar |
|---|---|
| DEF REAL ISCOORD[2] | ; Variable für Schnittpunkt-Koordinaten. |
| DEF BOOL ISPOINT | ; Variable für Schnittpunktstatus. |
| DEF INT MODE | ; Variable für Bearbeitungsart. |
| … | |
| MODE=1 | ; Berechnung unabhängig von der aktiven Ebene. |
| N10 ISPOINT=INTERSEC(TABNAME1[3],TABNAME2[7], ISCOORD,MODE) | ; Aufruf Schnittpunkt der Konturelemente. |
| N20 IF ISPOINT==FALSE GOTOF KEINSCH | ; Sprung zu KEINSCH. |
| … | |

## 3.24.5 Konturelemente einer Tabelle satzweise abfahren (EXECTAB)

Mit EXECTAB können Sie Konturelemente einer Tabelle, die z. B. mit dem CONTPRON erzeugt wurde, satzweise abfahren.

### Syntax

```
EXECTAB(<Konturtabelle>[<Konturelement>])
```

### Bedeutung

| | |
|---|---|
| EXECTAB: | Vordefinierte Proedur zum Abfahren eines Konturelements |
| <Konturtabelle>: | Name der Konturtabelle |
| <Konturelement>: | Nummer des Konturelements |

### Beispiel

Die Konturelemente 0 bis 2 der Tabelle KTAB sollen satzweise abgefahren werden.

| Programmcode | Kommentar |
|---|---|
| N10 EXECTAB(KTAB[0]) | ; Element 0 der Tabelle KTAB verfahren. |
| N20 EXECTAB(KTAB[1]) | ; Element 1 der Tabelle KTAB verfahren. |
| N30 EXECTAB(KTAB[2]) | ; Element 2 der Tabelle KTAB verfahren. |

## 3.24.6 Kreisdaten berechnen (CALCDAT)

Mit CALCDAT können Sie aus drei oder vier bekannten Kreispunkten den Radius und die Kreismittelpunkt-Koordinaten berechnen. Die angegebenen Punkte müssen unterschiedlich sein.

Bei 4 Punkten, die nicht exakt auf dem Kreis liegen, wird für Kreismittelpunkt und Radius ein Mittelwert gebildet.

**Hinweis**

**Rechenvorschrift zur Mittelwertbildung**

Die Kreisbogenberechnung wird 4 x ausgeführt:

1. mit Kreispunkt 1, 2, 3
2. mit Kreispunkt 1, 2, 4
3. mit Kreispunkt 1, 3, 4
4. mit Kreispunkt 2, 3, 4

Die Kreismittelpunkt-Koordinaten Abszissenwert und Ordinatenwert werden berechnet, indem die Abszissenwerte bzw. Ordinatenwerte der vier Kreisbogenberechnungen addiert und durch 4 geteilt werden.

Der Radius wird berechnet, indem die Wurzel aus der Summe der vier Radien der Kreisbogenberechnungen gebildet und das Ergebnis mit 0,5 multipliziert wird.

## Syntax

```
<Status>=CALCDAT(<Kreispunkte>[<Anzahl>,<Art>],<Anzahl>,<Ergebnis>)
```

## Bedeutung

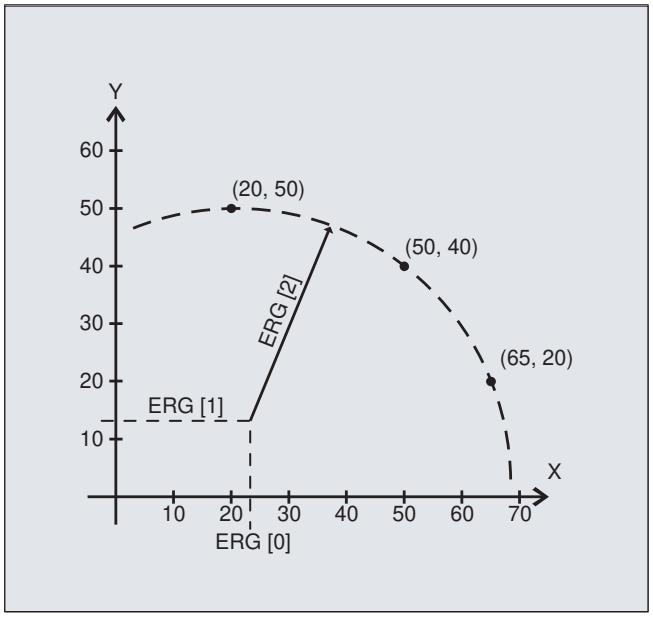
| | | | |
|---|---|---|---|
| CALCDAT: | Vordefinierte Funktion zur Berechnung von Radius und Mittelpunkt-Koordinaten eines Kreises aus 3 oder 4 Punkten | | |
| <Status>: | Variable für den Kreisberechnungsstatus | | |
| | Typ: | BOOL | |
| | Wert: | TRUE | Die angegebenen Punkte liegen auf einem Kreis. |
| | | FALSE | Die angegebenen Punkte liegen **nicht** auf einem Kreis. |
| <Kreispunkte>[]: | Variable zur Angabe der Kreispunkte mit den Parametern: | | |
| | <Anzahl>: | Anzahl der Kreispunkte (3 oder 4) | |
| | <Art>: | Art der Koordinatenangabe, z. B. 2 für 2-Punkt-Koordinaten | |
| <Anzahl>: | Parameter für die Anzahl der zur Berechnung verwendeten Punkte (3 oder 4) | | |
| <Ergebnis>[3]: | Variable für Ergebnis: Angabe von Kreismittelpunkt-Koordinaten und Radius | | |
| | 0 | Kreismittelpunkt-Koordinate: Abszissenwert | |
| | 1 | Kreismittelpunkt-Koordinate: Ordinatenwert | |
| | 2 | Radius | |

**Hinweis**

Beachten Sie, dass die Variablen vor ihrer Verwendung definiert sein müssen.

## Beispiel

Von drei Punkten soll ermittelt werden, ob sie auf einem Kreisabschnitt liegen.




| **Programmcode** | **Kommentar** |
|---|---|
| `N10 DEF REAL PKT[3,2]=(20,50,50,40,65,20)` | `; Variable zur Angabe der Kreispunkte.` |
| `N20 DEF REAL ERG[3]` | `; Variable für Ergebnis.` |
| `N30 DEF BOOL STATUS` | `; Variable für Status.` |
| `N40 STATUS=CALCDAT(PKT,3,ERG)` | `; Aufruf der ermittelten Kreisdaten.` |
| `N50 IF STATUS == FALSE GOTOF ERROR` | `; Sprung zu Fehler.` |

## 3.24.7 Konturaufbereitung ausschalten (EXECUTE)

Mit `EXECUTE` wird die Konturaufbereitung abgeschaltet und gleichzeitig in den normalen Abarbeitungsmodus zurückgeschaltet.

## Syntax

```
EXECUTE(<FEHLER>)
```

## Bedeutung

| | | |
|---|---|---|
| EXECUTE: | Vordefinierte Prozedur zum Beenden der Konturaufbereitung | |
| <FEHLER>: | Variable für Fehlerrückmeldung | |
| | Typ: | INT |
| | Der Wert der Variablen zeigt an, ob die Kontur fehlerfrei aufbereitet werden konnte: | |
| | 0 | Fehler |
| | 1 | kein Fehler |

## Beispiel

**Programmcode**

```
...
N30 CONTPRON(...)
N40 G1 X... Z...
...
N100 EXECUTE(...)
...
```

# 3.25 Zyklen extern programmieren

## 3.25.1 Technologische Zyklen

### 3.25.1.1 Einleitung

#### Inhalt

Dieses Kapitel enthält die Beschreibung der Zyklen für die Technologien Drehen, Fräsen, Schleifen.

#### Aufbau

Die Beschreibung eines Zyklus ist wie folgt aufgebaut:

- **Syntax**
  Zyklusname und Aufrufreihenfolge der Übergabeparameter
- **Parameter**
  Tabelle zur Erläuterung der einzelnen Parameter

#### Parameterbeschreibung

In der Tabelle sind zu einem Parameter folgende Daten angegeben: Name, Beschreibung, Wertebereich und Abhängigkeiten zu anderen Parametern.

Die Spalte zum Verweis auf den Parameter in der Maske dient der Erleichterung zum Wiederauffinden von an der Steuerung programmierte Werten beim Rückübersetzen von extern generierten Zyklenaufrufen.

**Parameter "nur für die Oberfläche"**

In der Tabelle sind Parameter mit "nur für die Oberfläche" gekennzeichnet. Diese sind für die Funktion des Zyklus nicht von Bedeutung. Sie werden nur benötigt, um Zyklenaufrufe vollständig rückübersetzen zu können. Sind sie nicht programmiert, kann der Zyklus trotzdem rückübersetzt werden, die Felder sind dann entsprechend farblich gekennzeichnet und müssen in der Maske ausgefüllt werden.

**Parameter "reserviert"**

Parameter, die mit "reserviert" beschrieben sind, müssen mit Wert 0 oder Leerkomma programmiert werden, damit die Zuordnung der folgenden Aufrufparameter zu den internen Zyklusparametern stimmt. Ausnahme: bei Stringparametern Wert "" oder Leerkomma.

#### Zyklen auf Positionsmuster wiederholen

Bohr- und Fräszyklen können auf Positionsmuster wiederholt werden (modale Aufrufe). Vor dem Zyklus ist dann in derselben Zeile `MCALL` zu schreiben, z. B. `MCALL CYCLE83(...)`.

**Hinweis**

Sind bestimmte Übergabeparameter (z. B. `<_VARI>`, `<_GMODE>`, `<_DMODE>`, `<_AMODE>`) indirekt als Parameter programmiert, wird die Eingabemaske beim Rückübersetzen geöffnet, kann aber nicht abgespeichert werden, weil es zu bestimmtem Auswahlfeldern keine eindeutige Zuordnung gibt.

### 3.25.1.2 Technologie-spezifische Übersicht

In der folgenden Übersichtstabelle sind alle verfügbaren extern programmierbaren technologischen Zyklen aufgelistet und der jeweiligen Technologie zugeordnet:

| Technologie | Technologischer Zyklus |
|---|---|
| Bohren | • CYCLE81 - Bohren, Zentrieren (Seite 1082)<br>• CYCLE82 - Bohren, Plansenken (Seite 1083)<br>• CYCLE85 - Reiben (Seite 1092)<br>• CYCLE86 - Ausdrehen (Seite 1093)<br>• CYCLE83 - Tieflochbohren 1 (Seite 1086)<br>• CYCLE830 - Tieflochbohren 2 (Seite 1120)<br>• CYCLE84 - Gewindebohren ohne Ausgleichsfutter (Seite 1089)<br>• CYCLE840 - Gewindebohren mit Ausgleichsfutter (Seite 1130)<br>• CYCLE78 - Bohrgewinde fräsen (Seite 1078)<br>• CYCLE802 - Beliebige Positionen (Seite 1117)<br>• HOLES1 - Positionsmuster Reihe (Seite 1045)<br>• CYCLE801 - Positionsmuster Gitter oder Rahmen (Seite 1115)<br>• HOLES2 - Positionsmuster Kreis oder Teilkreis (Seite 1045) |
| Drehen | • CYCLE951 - Abspanen (Seite 1140)<br>• CYCLE930 - Einstich (Seite 1135)<br>• CYCLE940 - Freistich Form E und F / Freistich Gewinde (Seite 1137)<br>• CYCLE99 - Gewindedrehen (Seite 1102)<br>• CYCLE98 - Gewindekette (Seite 1098)<br>• CYCLE92 - Abstich (Seite 1094) |
| Konturdrehen | • CYCLE62 - Konturaufruf (Seite 1064)<br>• CYCLE952 - Abspanen / Abspanen Rest / Stechen / Stechen Rest / Stechdrehen / Stechdrehen Rest (Seite 1143) |

| Technologie | Technologischer Zyklus |
|---|---|
| Fräsen | • CYCLE61 - Planfräsen (Seite 1062)<br>• POCKET3 - Rechtecktasche (Seite 1047)<br>• POCKET4 - Kreistasche (Seite 1050)<br>• CYCLE76 - Rechteckzapfen (Seite 1073)<br>• CYCLE77 - Kreiszapfen (Seite 1076)<br>• CYCLE79 - Mehrkant (Seite 1080)<br>• SLOT1 - Längsnut (Seite 1052)<br>• SLOT2 - Kreisnut (Seite 1055)<br>• CYCLE899 - Offene Nut (Seite 1132)<br>• LONGHOLE - Langloch (Seite 1057)<br>• CYCLE70 - Gewindefräsen (Seite 1069)<br>• CYCLE60 - Gravur (Seite 1059) |
| Konturfräsen | • CYCLE62 - Konturaufruf (Seite 1064)<br>• CYCLE72 - Bahnfräsen (Seite 1070)<br>• CYCLE63 - Konturtasche fräsen / Konturtasche Restmaterial / Konturzapfen fräsen / Konturzapfen Restmaterial (Seite 1064)<br>• CYCLE64 - Konturtasche vorbohren (Seite 1067) |
| Schleifen | • CYCLE495 - Profilieren (Seite 1107)<br>• CYCLE435 - Abrichterkoordinatensystem setzen (Seite 1107)<br>• CYCLE4071 - Längsschleifen mit Zustellung am Umkehrpunkt (Seite 1149)<br>• CYCLE4072 - Längsschleifen mit Zustellung am Umkehrpunkt und Abbruchsignal (Seite 1151)<br>• CYCLE4073 - Längschleifen mit kontinuierlicher Zustellung (Seite 1155)<br>• CYCLE4074 - Längschleifen mit kontinuierlicher Zustellung und Abbruchsignal (Seite 1156)<br>• CYCLE4075 - Flachschleifen mit Zustellung am Umkehrpunkt (Seite 1159)<br>• CYCLE4077 - Flachschleifen mit Zustellung am Umkehrpunkt und Abbruchsignal (Seite 1162)<br>• CYCLE4078 - Flachschleifen mit kontinuierlicher Zustellung (Seite 1166)<br>• CYCLE4079 - Flachschleifen mit intermittierender Zustellung (Seite 1168) |
| Sonstige | • CYCLE782 - An Beladung anpassen (Seite 1109)<br>• CYCLE800 - Schwenken Ebene / Schwenken Werkzeug / Ausrichten Werkzeug (Seite 1112)<br>• CYCLE832 - High Speed Settings (Seite 1126) |
| Alle | • GROUP_BEGIN - Anfang Programmblock (Seite 1171)<br>• GROUP_END - Ende Programmblock (Seite 1171)<br>• GROUP_ADDEND - Ende Einfahrzusatz (Seite 1172) |

### 3.25.1.3 HOLES1 - Positionsmuster Reihe

## Syntax

```
HOLES1(<SPCA>, <SPCO>, <STA1>, <FDIS>, <DBH>, <NUM>, <_VARI>,
<_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | X0 | <SPCA> | REAL | Bezugspunkt für Lochreihe in der 1. Achse (abs) | | |
| 2 | Y0 | <SPCO> | REAL | Bezugspunkt für Lochreihe in der 2. Achse (abs) | | |
| 3 | α0 | <STA1> | REAL | Grund-Drehwinkel (Winkel zur 1. Achse) | | |
| 4 | L0 | <FDIS> | REAL | Abstand der 1. Bohrung vom Bezugspunkt | | |
| 5 | L | <DBH> | REAL | Abstand zwischen den Bohrungen | | |
| 6 | N | <NUM> | INT | Anzahl der Bohrungen | | |
| 7 | | <_VARI> | INT | reserviert | | |
| 8 | | <_UMODE> | INT | reserviert | | |
| 9 | | <_HIDE> | STRING [200] | Ausgeblendete Positionen<br>• max. 198 Zeichen<br>• Angabe der fortlaufenden Positionsnummer, z. B. "1,3" (Positionen 1 und 3 werden nicht ausgeführt) | | |
| 10 | | <_NSP> | INT | reserviert | | |
| 11 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/18/19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |

### 3.25.1.4 HOLES2 - Positionsmuster Kreis oder Teilkreis

## Syntax

```
HOLES2(<CPA>, <CPO>, <RAD>, <STA1>, <INDA>, <NUM>, <_VARI>,
<_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | | |
|---|---|---|---|---|---|---|---|
| 1 | X0 | <CPA> | REAL | Mittelpunkt für Lochkreis in der 1. Achse (abs)<br>Bezugspunkt in der 1. Achse | | | (bei XY)<br>(bei XA, YB, ZC) |
| 2 | Y0 | <CPO> | REAL | Mittelpunkt für Lochkreis in der 2. Achse (abs)<br>Bezugspunkt in der 2. Achse | | | (bei XY)<br>(bei XA, YB, ZC) |
| 3 | R | <RAD> | REAL | Radius des Lochkreises | | | (bei XY) |
| 4 | α0 | <STA1> | REAL | Anfangswinkel<br>oder 1. Rundachsposition | | | (bei XY)<br>(bei XA, YB, ZC) |
| 5 | α1 | <INDA> | REAL | Fortschaltwinkel (nur bei Teilkreis) | | | (bei XY, XA, YB, ZC) |
| | | | | | < 0 = | Uhrzeigersinn | |
| | | | | | > 0 = | Gegenuhrzeigersinn | |
| 6 | N | <NUM> | INT | Anzahl der Positionen | | | |
| 7 | | <_VARI> | INT | Bearbeitungsart | | | |
| | | | | EINER: | reserviert | | |
| | | | | ZEHNER: | Positionierungsart | | |
| | | | | | 0 = | Position anfahren - Linear | |
| | | | | | 1 = | Position anfahren - auf Kreisbahn | |
| | | | | HUNDERTER: | reserviert | | |
| | | | | TAUSENDER: | Kreismuster | | |
| | | | | | 0 = | Kompatibilitätsmodus, wenn INDA = 0 dann Vollkreis, INDA <> 0 dann Teilkreis) | |
| | | | | | 1 = | Vollkreis | |
| | | | | | 2 = | Teilkreis | |
| | | | | ZEHNTAUSEN-DER: | Positionsmuster mit Rundachse | | |
| | | | | | 0 = | XY (ohne Rund-achse) | (bei XY) |
| | | | | | 1 = | XA (X-Achse und Rundachse um X) | (nur bei XA) |
| | | | | | 2 = | YB (Y-Achse und Rundachse um Y) | (nur bei YB) |
| | | | | | 3 = | ZC (Z-Achse und Rundachse um C) | (nur bei ZC) |
| | | | | EINEMILLION + HUNDERTTAU-SENDER: | Offset (bei mehreren Rundachsen um die gleiche Achse; wenn Index zu groß, dann 1. Achse) | | |
| | | | | | 00 = | 1. A, B oder C-Achse | |
| | | | | | 01 = | 2. A, B oder C-Achse | |
| | | | | | ... | | |
| | | | | | 10 = | 20. A, B oder C-Achse | |
| 8 | | <_UMODE> | INT | reserviert | | | |
| 9 | | <_HIDE> | STRING [200] | reserviert | | | |
| 10 | | <_NSP> | INT | reserviert | | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 11 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/18/19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |

### 3.25.1.5 POCKET3 - Rechtecktasche

## Syntax

```
POCKET3(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_LENG>, <_WID>, <_CRAD>,
<_PA>, <_PO>, <_STA>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>,
<_CDIR>, <_VARI>, <_MIDA>, <_AP1>, <_AP2>, <_AD>, <_RAD1>, <_DP1>,
<_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | RP | <_RTP> | REAL | Rückzugsebene (abs) | | |
| 2 | Z0 | <_RFP> | REAL | Bezugspunkt der Werkzeugachse (abs) | | |
| 3 | SC | <_SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) | | |
| 4 | Z1 | <_DP> | REAL | Taschentiefe (abs/ink), siehe <_AMODE>) | | |
| 5 | L | <_LENG> | REAL | Taschenlänge (ink, mit Vorzeichen einzugeben) | | |
| 6 | W | <_WID> | REAL | Taschenbreite (ink, mit Vorzeichen einzugeben) | | |
| 7 | R | <_CRAD> | REAL | Eckenradius der Tasche | | |
| 8 | X0 | <_PA> | REAL | Bezugspunkt, 1. Achse (abs) | | |
| 9 | YO | <_PO> | REAL | Bezugspunkt, 2. Achse (abs) | | |
| 10 | α0 | <_STA> | REAL | Drehwinkel, Winkel zwischen Längsachse (L) und 1. Achse | | |
| 11 | DZ | <_MID> | REAL | maximale Tiefenzustellung | | |
| 12 | UXY | <_FAL> | REAL | Schlichtaufmaß Ebene | | |
| 13 | UZ | <_FALD> | REAL | Schlichtaufmaß Tiefe | | |
| 14 | F | <_FFP1> | REAL | Vorschub in der Ebene | | |
| 15 | FZ | <_FFD> | REAL | Zustellvorschub Tiefe | | |
| 16 | | <_CDIR> | INT | Fräsrichtung: | 0 = | Gleichlauf |
| | | | | | 1 = | Gegenlauf |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 17 | | <_VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | | |
| | | | | | 1 = | Schruppen |
| | | | | | 2 = | Schlichten |
| | | | | | 4 = | Schlichten Rand |
| | | | | | 5 = | Anfasen |
| | | | | ZEHNER: | | |
| | | | | | 0 = | vorgebohrt, Zustellung mit G0 |
| | | | | | 1 = | senkrecht, Zustellung mit G1 |
| | | | | | 2 = | helikal |
| | | | | | 3 = | pendeln auf Taschenlängsachse |
| | | | | HUNDERTER: | reserviert | |
| 18 | DXY | <_MIDA> | REAL | maximale Ebenenzustellung, Einheit siehe <_AMODE> | | |
| 19 | L1 | <_AP1> | REAL | Länge der Vorbearbeitung (ink) | | |
| 20 | W1 | <_AP2> | REAL | Breite der Vorbearbeitung (ink) | | |
| 21 | AZ | <_AD> | REAL | Tiefe der Vorbearbeitung (ink) | | |
| 22 | ER | <_RAD1> | REAL | Radius der Helixbahn beim Eintauchen helikal | | |
| | EW | | | maximaler Eintauchwinkel für pendelnd | | |
| 23 | EP | <_DP1> | REAL | Helixsteigung bei Eintauchen helikal | | |
| 24 | | <_UMODE> | INT | reserviert | | |
| 25 | FS | <_FS> | REAL | Fasenbreite (ink) | | |
| 26 | ZFS | <_ZFS> | REAL | Eintauchtiefe (Werkzeugspitze) bei Anfasen (abs/ink), siehe <_AMODE> | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 27 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Auswahl Bearbeitung/nur Startpunktberechnung | |
| | | | | | 0 = | Kompatibilitätsmodus |
| | | | | | 1 = | normale Bearbeitung |
| | | | | TAUSENDER: | Bemaßung über Mitte/Ecke | |
| | | | | | 0 = | Kompatibilitätsmodus |
| | | | | | 1 = | Bemaßung über Mitte |
| | | | | | 2 = | Bemaßung Eckpunkt, Taschenlage +LENG/+WID |
| | | | | | 3 = | Bemaßung Eckpunkt, Taschenlage -LENG/+WID |
| | | | | | 4 = | Bemaßung Eckpunkt, Taschenlage +LENG/-WID |
| | | | | | 5 = | Bemaßung Eckpunkt, Taschenlage -LENG/-WID |
| | | | | ZEHNTAUSENDER: | Komplettbearbeitung/Nachbearbeitung | |
| | | | | | 0 = | Kompatibilitätsmodus (<_AP1>, <_AP2> und <_AD> wie bisher behandeln) |
| | | | | | 1 = | Komplettbearbeitung |
| | | | | | 2 = | Nachbearbeitung |
| 28 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | Vorschubart: G-Gruppe (G94/G95) für Flächen- und Tiefenvorschub | |
| | | | | | 0 = | Kompatibilitätsmodus |
| | | | | | 1 = | G-Befehl wie vor Zyklusaufruf. G94/G95 für Flächen- und Tiefenvorschub gleich |
| | | | | HUNDERTER: | --- | reserviert |
| | | | | TAUSENDER: | --- | reserviert |
| | | | | ZEHNTAUSENDER: | Technologieskalierung innerhalb von Zyklenmasken (Seite 1172) | |
| | | | | | 0 = | Eingabe: komplett |
| | | | | | 1 = | Eingabe: einfach |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 29 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Taschentiefe (Z1) | |
| | | | | | 0 = | absolut (Kompatibilitätsmodus) |
| | | | | | 1 = | inkrementell |
| | | | | ZEHNER: | Einheit für Ebenenzustellung (DXY) | |
| | | | | | 0 = | mm |
| | | | | | 1 = | % vom Werkzeugdurchmesser |
| | | | | HUNDERTER: | Eintauchtiefe bei Anfasen (ZFS) | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |

### 3.25.1.6 POCKET4 - Kreistasche

## Syntax

```
POCKET4(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_CDIAM>, <_PA>, <_PO>,
<_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>, <_VARI>, <_MIDA>,
<_AP1>, <_AD>, <_RAD1>, <_DP1>, <_UMODE>, <_FS>, <_ZFS>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | RP | <_RTP> | REAL | Rückzugsebene (abs) | | |
| 2 | Z0 | <_RFP> | REAL | Bezugspunkt der Werkzeugachse (abs) | | |
| 3 | SC | <_SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) | | |
| 4 | Z1 | <_DP> | REAL | Taschentiefe (abs/ink), siehe <_AMODE> | | |
| 5 | ∅ | <_CDIAM> | REAL | Taschendurchmesser oder Taschenradius, siehe <_DMODE> | | |
| 6 | X0 | <_PA> | REAL | Bezugspunkt 1. Achse (abs) | | |
| 7 | Y0 | <_PO> | REAL | Bezugspunkt 2. Achse (abs) | | |
| 8 | DZ | <_MID> | REAL | maximale Tiefenzustellung, siehe <_VARI> = ebenenweise<br>maximale Helixsteigung, siehe <_VARI> = helikal | | |
| 9 | UXY | <_FAL> | REAL | Schlichtaufmaß Ebene | | |
| 10 | UZ | <_FALD> | REAL | Schlichtaufmaß Tiefe | | |
| 11 | F | <_FFP1> | REAL | Vorschub für Flächenbearbeitung | | |
| 12 | FZ | <_FFD> | REAL | Zustellvorschub Tiefe | | |
| 13 | | <_CDIR> | INT | Fräsrichtung | 0 = | Gleichlauf |
| | | | | | 1 = | Gegenlauf |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 14 | | <_VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | Bearbeitung | |
| | | | | | 1 = | Schruppen |
| | | | | | 2 = | Schlichten |
| | | | | | 4 = | Schlichten Rand |
| | | | | | 5 = | Anfasen |
| | | | | ZEHNER: | Zustellungsart (Schruppen und Schlichten) | |
| | | | | | 0 = | vorgebohrt, Zustellung mit G0 (Tasche ist vorgearbeitet) |
| | | | | | 1 = | senkrecht, Zustellung mit G1 |
| | | | | | 2 = | helikal |
| | | | | HUNDERTER: | reserviert | |
| | | | | TAUSENDER: | | |
| | | | | | 0 = | ebenenweise |
| | | | | | 1 = | helikal |
| 15 | DXY | <_MIDA> | REAL | maximale Ebenenzustellung, siehe <_AMODE>, 0 = 0,8 x WZG-Durchmesser | | |
| 16 | ∅ | <_AP1> | REAL | Durchmesser/Radius der Vorbearbeitung (ink) | | |
| 17 | AZ | <_AD> | REAL | Tiefe der Vorbearbeitung (ink) | | |
| 18 | ER | <_RAD1> | REAL | Radius der Helixbahn beim Eintauchen helikal | | |
| 19 | EP | <_DP1> | REAL | Helixsteigung bei Eintauchen auf Helixbahn | | |
| 20 | | <_UMODE> | INT | reserviert | | |
| 21 | FS | <_FS> | REAL | Fasenbreite (ink) | | |
| 22 | ZFS | <_ZFS> | REAL | Eintauchtiefe (Werkzeugspitze) bei Anfasen (abs/ink), siehe <_AMODE> | | |
| 23 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Bearbeitung/Startpunktberechnung | |
| | | | | | 0 = | Kompatibilitätsmodus |
| | | | | | 1 = | normale Bearbeitung |
| | | | | TAUSENDER: | reserviert | |
| | | | | ZEHNTAUSENDER: | Komplettbearbeitung/Nachbearbeitung | |
| | | | | | 0 = | Kompatibilitätsmodus (<_AP1> und <_AD> wie bisher behandeln) |
| | | | | | 1 = | Komplettbearbeitung |
| | | | | | 2 = | Nachbearbeitung |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 24 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/18/19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | Vorschubart: G-Gruppe (G94/G95) für Flächen- und Tiefenvorschub | |
| | | | | | 0 = | Kompatibilitätsmodus |
| | | | | | 1 = | G-Befehl wie vor Zyklusaufruf. G94/G95 für Flächen- und Tiefenvorschub gleich |
| | | | | HUNDERTER: | | |
| | | | | | 0 = | Kompatibilitätsmodus (<_CDIAM>/<_AP1> alsRadius eingegeben) |
| | | | | | 1 = | <_CDIAM>/<_AP1> als Durchmesser eingegeben |
| | | | | TAUSENDER: | --- | reserviert |
| | | | | ZEHNTAUSENDER: | Technologieskalierung innerhalb von Zyklenmasken (Seite 1172) | |
| | | | | | 0 = | Eingabe: komplett |
| | | | | | 1 = | Eingabe: einfach |
| 25 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Taschentiefe (Z1) | |
| | | | | | 0 = | absolut (Kompatibilitätsmodus) |
| | | | | | 1 = | inkrementell |
| | | | | ZEHNER: | Einheit für Zustellbreite (DXY) | |
| | | | | | 0 = | mm |
| | | | | | 1 = | % vom Werkzeugdurchmesser |
| | | | | HUNDERTER: | Eintauchtiefe bei Anfasen (ZFS) | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |

### 3.25.1.7 SLOT1 - Längsnut

#### Syntax

```
SLOT1 (<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <LENG>, <WID>,
<_CPA>, <_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <_MID>,
<CDIR>, <_FAL>, <VARI>, <_MIDF>, <FFP2>, <SSF>, <_FALD>, <_STA2>,
<_DP1>, <_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | RP | <RTP> | REAL | Rückzugsebene (abs) | | |
| 2 | Z0 | <RFP> | REAL | Bezugspunkt der Werkzeugachse (abs) | | |
| 3 | SC | <SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) | | |
| 4 | Z1 | <_DP> | REAL | Nuttiefe (abs) | | |
| 5 | | <_DPR> | REAL | Nuttiefe (ink), bezogen auf Z0 (ohne Vorzeichen einzugeben) | | |
| 6 | | <NUM> | INT | Anzahl der Nuten = 1 | | |
| 7 | L | <LENG> | REAL | Länge der Nut | | |
| 8 | W | <WID> | REAL | Breite der Nut | | |
| 9 | X0 | <_CPA> | REAL | Bezugspunkt, 1. Achse der Ebene | | |
| 10 | Y0 | <_CPO> | REAL | Bezugspunkt, 2. Achse der Ebene | | |
| 11 | | <RAD> | REAL | reserviert | | |
| 12 | α | <STA1> | REAL | Drehwinkel | | |
| 13 | | <INDA> | REAL | reserviert | | |
| 14 | FZ | <FFD> | REAL | Zustellvorschub Tiefe | | |
| 15 | F | <FFP1> | REAL | Vorschub | | |
| 16 | DZ | <_MID> | REAL | maximale Tiefenzustellung | | |
| 17 | | <CDIR> | INT | Fräsrichtung | 0 = | Gleichlauf |
| | | | | | 1 = | Gegenlauf |
| 18 | UXY | <_FAL> | REAL | Schlichtaufmass Ebene oder Nutrand | | |
| 19 | | <VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | | |
| | | | | | 0 = | reserviert |
| | | | | | 1 = | Schruppen |
| | | | | | 2 = | Schlichten |
| | | | | | 4 = | Schlichten Rand (nur Rand bearbeiten) |
| | | | | | 5 = | Anfasen |
| | | | | ZEHNER: | Anfahren | |
| | | | | | 0 = | vorgebohrt, Zustellung mit G0 (Nut ist vorbearbeitet) |
| | | | | | 1 = | senkrecht, Zustellung mit G1 |
| | | | | | 2 = | helikal |
| | | | | | 3 = | pendelnd |
| | | | | HUNDERTER: | reserviert | |
| 20 | DZF | <_MIDF> | REAL | reserviert | | |
| 21 | FF | <FFP2> | REAL | reserviert | | |
| 22 | SF | <SSF> | REAL | reserviert | | |
| 23 | UZ | <_FALD> | REAL | Schlichtaufmaß Tiefe | | |
| 24 | ER | <_STA2> | REAL | Radius der Helixbahn beim Eintauchen helikal | | |
| | EW | | | maximaler Eintauchwinkel für pendelnd | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 25 | EP | <_DP1> | REAL | Eintauchtiefe pro Umdrehung für Helix | | |
| 26 | | <_UMODE> | INT | reserviert | | |
| 27 | FS | <_FS> | REAL | Fasenbreite (ink) bei Anfasen | | |
| 28 | ZFS | <_ZFS> | REAL | Eintauchtiefe (Werkzeugspitze) bei Anfasen (abs/ink), siehe <_AMODE> | | |
| 29 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Auswahl Bearbeitung oder nur Startpunktberechnung | |
| | | | | | 1 = | normale Bearbeitung |
| | | | | TAUSENDER: | Bemaßung Bezugspunkt, Nutlage | |
| | | | | | 0 = | Mitte |
| | | | | | 1 = | links innen +L |
| | | | | | 2 = | rechts innen -L |
| | | | | | 3 = | linker Rand +L |
| | | | | | 4 = | rechter Rand -L |
| 30 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/18/19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | reserviert | |
| | | | | TAUSENDER: | Kennung SW-Version | |
| | | | | | 1 = | Funktionserweiterung SLOT1 |
| | | | | ZEHNTAUSENDER: | Technologieskalierung innerhalb von Zyklenmasken (Seite 1172) | |
| | | | | | 0 = | Eingabe: komplett |
| | | | | | 1 = | Eingabe: einfach |
| 31 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Endtiefe Z1 (abs/ink) | |
| | | | | | 0 = | Kompatibilität |
| | | | | | 1 = | Z1 (ink) |
| | | | | | 2 = | Z1 (abs) |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Eintauchtiefe bei Anfasen ZFS | |
| | | | | | 0 = | ZFS (abs) |
| | | | | | 1 = | ZFS (ink) |

**Hinweis**

Der Zyklus ist gegenüber früheren SW-Versionen mit neuen Funktionen ausgestattet. Das hat zur Folge, dass bestimmte Parameter in der Eingabemaske nicht mehr angezeigt werden (<NUM>, <RAD>, <INDA>). Mehrere Nuten auf einem Positionsmuster sind mittels "MCALL" und Aufruf des gewünschten Positionsmusters, z B HOLES2, programmierbar.

### 3.25.1.8 SLOT2 - Kreisnut

## Syntax

```
SLOT2(<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <AFSL>, <WID>,
<_CPA>, <_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <_MID>,
<CDIR>, <_FAL>, <VARI>, <_MIDF>, <FFP2>, <SSF>, <_FFCP>, <_UMODE>,
<_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | RP | <RTP> | REAL | Rückzugsebene (abs) | | |
| 2 | Z0 | <RFP> | REAL | Bezugspunkt der Werkzeugachse (abs) | | |
| 3 | SC | <SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) | | |
| 4 | Z1 | <_DP> | REAL | Nuttiefe (abs) | | |
| 5 | | <_DPR> | REAL | Nuttiefe (ink), bezogen auf Z0 (ohne Vorzeichen einzugeben) | | |
| 6 | N | <NUM> | INT | Anzahl der Nuten | | |
| 7 | α1 | <AFSL> | REAL | Öffnungswinkel der Nut | | |
| 8 | W | <WID> | REAL | Breite der Nut | | |
| 9 | X0 | <_CPA> | REAL | Bezugspunkt = Mittelpunkt des Kreises, 1. Achse der Ebene | | |
| 10 | Y0 | <_CPO> | REAL | Bezugspunkt = Mittelpunkt des Kreises, 2. Achse der Ebene | | |
| 11 | R | <RAD> | REAL | Radius des Kreises | | |
| 12 | α0 | <STA1> | REAL | Anfangswinkel | | |
| 13 | α2 | <INDA> | REAL | Fortschaltwinkel | | |
| 14 | FZ | <FFD> | REAL | Zustellvorschub Tiefe | | |
| 15 | F | <FFP1> | REAL | Vorschub | | |
| 16 | DZ | <_MID> | REAL | maximale Tiefenzustellung | | |
| 17 | | <CDIR> | INT | Fräsrichtung | 0 = | Gleichlauf |
| | | | | | 1 = | Gegenlauf |
| 18 | UXY | <_FAL> | REAL | Schlichtaufmaß Ebene oder Nutrand | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 19 | | <VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | | |
| | | | | | 0 = | Komplettbearbeitung |
| | | | | | 1 = | Schruppen |
| | | | | | 2 = | Schlichten |
| | | | | | 3 = | Schlichten Rand |
| | | | | | 5 = | Anfasen |
| | | | | ZEHNER: | | |
| | | | | | 0 = | Zwischenpositionieren mit G0-Gerade |
| | | | | | 1 = | Zwischenpositionieren auf Kreisbahn |
| | | | | HUNDERTER: | reserviert | |
| | | | | TAUSENDER: | | |
| | | | | | 0 = | Kompatibilitätsmode, wenn <INDA> = 0 dann Vollkreis, <INDA> <> 0 dann Teilkreis) |
| | | | | | 1 = | Vollkreis |
| | | | | | 2 = | Teilkreis |
| 20 | DZF | <_MIDF> | REAL | reserviert | | |
| 21 | | <FFP2> | REAL | reserviert | | |
| 22 | | <SSF> | REAL | reserviert | | |
| 23 | FF | <_FFCP> | REAL | reserviert | | |
| 24 | | <_UMODE> | INT | reserviert | | |
| 25 | FS | <_FS> | REAL | Fasenbreite (Ink) | | |
| 26 | ZFS | <_ZFS> | REAL | Eintauchtiefe (Werkzeugspitze) bei Anfasen (abs/ink), siehe <_AMODE> | | |
| 27 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Auswahl Bearbeitung oder nur Startpunktberechnung | |
| | | | | | 0 = | Kompatibilitätsmodus |
| | | | | | 1 = | normale Bearbeitung |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 28 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/18/19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | reserviert | |
| | | | | TAUSENDER: | Kennung SW-Version | |
| | | | | | 1 = | Funktionen SLOT2 ab SW 2.5 |
| | | | | ZEHNTAUSENDER: | Technologieskalierung innerhalb von Zyklenmasken (Seite 1172) | |
| | | | | | 0 = | Eingabe: komplett |
| | | | | | 1 = | Eingabe: einfach |
| 29 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Endtiefe Z1 (abs/ink) | |
| | | | | | 0 = | Kompatibilität |
| | | | | | 1 = | Z1 (ink) |
| | | | | | 2 = | Z1 (abs) |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Eintauchtiefe bei Anfasen ZFS | |
| | | | | | 0 = | ZFS (abs) |
| | | | | | 1 = | ZFS (ink) |

### 3.25.1.9 LONGHOLE - Langloch

## Syntax

```
LONGHOLE(<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <LENG>, <_CPA>,
<_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <MID>, <_VARI>,
<_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung |
|---|---|---|---|---|
| 1 | RP | <RTP> | REAL | Rückzugsebene (abs) |
| 2 | Z0 | <_RFP> | REAL | Bezugspunkt der Werkzeugachse (abs) |
| 3 | SC | <SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) |
| 4 | Z1 | <_DP> | REAL | Langlochtiefe (abs) |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 5 | | <_DPR> | REAL | Langlochtiefe (ink), bezogen auf Z0 (ohne Vorzeichen einzugeben) | | |
| 6 | | <NUM> | INT | Anzahl der Langlöcher = 1 | | |
| 7 | L | <LENG> | REAL | Länge Langloch | | |
| 8 | X0 | <_CPA> | REAL | Bezugspunkt, 1. Achse der Ebene | | |
| 9 | Y0 | <_CPO> | REAL | Bezugspunkt, 2. Achse der Ebene | | |
| 10 | | <RAD> | REAL | reserviert | | |
| 11 | α0 | <STA1> | REAL | Drehwinkel | | |
| 12 | | <INDA> | REAL | reserviert | | |
| 13 | FZ | <FFD> | REAL | Zustellvorschub Tiefe | | |
| 14 | F | <FFP1> | REAL | Vorschub | | |
| 15 | DZ | <MID> | REAL | maximale Tiefenzustellung | | |
| 16 | | <_VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | Zustellart | |
| | | | | | 1 = | senkrecht mit G1 |
| | | | | | 3 = | pendelnd |
| | | | | HUNDERTER: | reserviert | |
| 17 | | <_UMODE> | INT | reserviert | | |
| 18 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Auswahl Bearbeitung oder nur Startpunktberechnung | |
| | | | | | 0 = | Kompatibilitätsmode |
| | | | | | 1 = | normale Bearbeitung |
| | | | | TAUSENDER: | Bemaßung Bezugspunkt, Nutlage | |
| | | | | | 0 = | Mitte |
| | | | | | 1 = | links innen +L |
| | | | | | 2 = | rechts innen -L |
| | | | | | 3 = | linker Rand +L |
| | | | | | 4 = | rechter Rand -L |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 19 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/18/19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | Vorschubart: G-Gruppe (G94/G95) für Flächen- und Tiefenvorschub | |
| | | | | | 0 = | Kompatibilitätsmode |
| | | | | | 1 = | G-Befehl wie vor Zyklusaufruf. G94/G95 für Flächen- und Tiefenvorschub gleich |
| | | | | HUNDERTER: | reserviert | |
| | | | | TAUSENDER: | Kennung SW-Version | |
| | | | | | 1 = | Funktionserweiterung LONGHOLE (Bemaßung Bezugspunkt) |
| 20 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Endtiefe Z1 (abs/ink) | |
| | | | | | 0 = | Kompatibilität |
| | | | | | 1 = | Z1 (ink) |
| | | | | | 2 = | Z1 (abs) |

**Hinweis**

Der Zyklus ist gegenüber früheren SW-Versionen mit neuen Funktionen ausgestattet. Das hat zur Folge, dass bestimmte Parameter in der Eingabemaske nicht mehr angezeigt werden (<NUM>, <RAD>, <INDA>). Mehrere Nuten auf einem Positionsmuster sind mittels "MCALL" und Aufruf des gewünschten Positionsmusters, z. B. HOLES2, programmierbar.

### 3.25.1.10 CYCLE60 - Gravur

## Syntax

```
CYCLE60(<_TEXT>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_PA>,
<_PO>, <_STA>, <_CP1>, <_CP2>, <_WID>, <_DF>, <_FFD>, <_FFP1>,
<_VARI>, <_CODEP>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung |
|---|---|---|---|---|
| 1 | | <_TEXT> | STRING [200] | zu gravierender Text (maximal 100 Zeichen) |
| 2 | RP | <_RTP> | REAL | Rückzugsebene (abs) |
| 3 | Z0 | <_RFP> | REAL | Bezugspunkt der Werkzeugachse (abs) |
| 4 | SC | <_SDIS> | REAL | Sicherheitsabstand (additiv auf Referenzebene, ohne Vorzeichen einzugeben) |
| 5 | Z1 | <_DP> | REAL | Tiefe (abs), siehe <_AMODE> |
| 6 | Z1 | <_DPR> | REAL | Tiefe (ink), siehe <_AMODE> |
| 7 | X0 | <_PA> | REAL | Bezugspunkt, 1. Achse der Ebene (abs) - rechtwinklig, siehe <_VARI> |
| | R | | | Bezugspunkt, Länge (Radius) - polar, siehe <_VARI> |
| 8 | Y0 | <_PO> | REAL | Bezugspunkt, 2. Achse der Ebene (abs) - rechtwinklig, siehe <_VARI> |
| | α0 | | | Bezugspunkt, Winkel bezogen auf die 1. Achse - polar, siehe <_VARI> |
| 9 | α1 | <_STA> | REAL | Textrichtung, Winkel der Textlinie bezogen auf die 1. Achse), siehe <_VARI> |
| 10 | XM | <_CP1> | REAL | Mittelpunkt des Textkreises, 1. Achse der Ebene (abs) - rechtwinklig, siehe <_VARI> |
| | LM | | | Mittelpunkt des Textkreises, Länge (Radius) bezogen auf WNP - polar, siehe <_VARI> |
| 11 | YM | <_CP2> | REAL | Mittelpunkt des Textkreises, 2. Achse der Ebene (abs) - rechtwinklig, siehe <_VARI> |
| | αM | | | Mittelpunkt des Textkreises, Winkel bezogen auf die 1. Achse - polar, siehe <_VARI> |
| 12 | W | <_WID> | REAL | Zeichenhöhe (ohne Vorzeichen einzugeben) |
| 13 | DX1<br>DX2 | <_DF> | REAL | Zeichenabstand / Gesamtbreite, siehe <_VARI> |
| | α2 | | | Öffnungswinkel, siehe <_VARI> |
| 14 | FZ | <_FFD> | REAL | Zustellvorschub Tiefe, siehe <_DMODE> |
| 15 | F | <_FFP1> | REAL | Vorschub für Flächenbearbeitung |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 16 | | <_VARI> | INT | Bearbeitung (Ausrichtung und Bezugspunkt des Gravurtextes) | | |
| | | | | EINER: | Bezugspunkt | |
| | | | | | 0 = | rechtwinklig |
| | | | | | 1 = | polar |
| | | | | ZEHNER: | Textausrichtung | |
| | | | | | 0 = | Text auf einer Linie |
| | | | | | 1 = | Text auf einem Kreisbogen oben |
| | | | | | 2 = | Text auf einem Kreisbogen unten |
| | | | | HUNDERTER: | reserviert | |
| | | | | TAUSENDER: | Bezugspunkt des Textes horizontal | |
| | | | | | 0 = | links |
| | | | | | 1 = | mittig |
| | | | | | 2 = | rechts |
| | | | | ZEHNTAUSENDER: | Bezugspunkt des Textes vertikal | |
| | | | | | 0 = | unten |
| | | | | | 1 = | mittig |
| | | | | | 2 = | oben |
| | | | | HUNDERTTAUSENDER: | Textlänge | |
| | | | | | 0 = | Zeichenabstand |
| | | | | | 1 = | Gesamtbreite des Textes (nur bei Text linear) |
| | | | | | 2 = | Öffnungswinkel (nur bei Text auf Kreisbogen) |
| | | | | EINEMILLION: | Kreismittelpunkt | |
| | | | | | 0 = | rechtwinklig (kartesisch) |
| | | | | | 1 = | polar |
| | | | | ZEHNMILLIONEN: | Spiegelschrift | |
| | | | | | 0 = | Kompatibilität |
| | | | | | 1 = | Spiegelschrift EIN |
| | | | | | 2 = | Spiegelschrift AUS |
| 17 | | <_CODEP> | INT | Nummer der Codepage für die Schrift (z. Z. nur 1252) | | |
| 18 | | <_UMODE> | INT | reserviert | | |
| 19 | | _GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER : | reserviert | |
| | | | | HUNDERTER: | Auswahl Bearbeitung / nur Startpunktberechnung | |
| | | | | | 0 = | Kompatibilitätsmodus |
| | | | | | 1 = | normale Bearbeitung |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 20 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/18/19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER : | Vorschubart: G-Gruppe (G94/G95) für Flächen- und Tiefenvorschub | |
| | | | | | 0 = | Kompatibilitätsmodus |
| | | | | | 1 = | G-Befehl wie vor Zyklusaufruf. G94/G95 für Flächen- und Tiefenvorschub gleich |
| 21 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Endtiefe (<_DP>, <_DPR>) | |
| | | | | | 0 = | Kompatibilität |
| | | | | | 1 = | inkrementell (<_DPR>) |
| | | | | | 2 = | absolut (<_DP>) |

### 3.25.1.11 CYCLE61 - Planfräsen

#### Syntax

```
CYCLE61(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_PA>, <_PO>, <_LENG>,
<_WID>, <_MID>, <_MIDA>, <_FALD>, <_FFP1>, <_VARI>, <_LIM>,
<_DMODE>, <_AMODE>)
```

#### Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung |
|---|---|---|---|---|
| 1 | RP | <_RTP> | REAL | Rückzugsebene (abs) |
| 2 | Z0 | <_RFP> | REAL | Bezugspunkt der Werkzeugachse, Höhe Rohteil (abs) |
| 3 | SC | <_SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) |
| 4 | Z1 | <_DP> | REAL | Höhe Fertigteil (abs/ink), siehe <_AMODE> |
| 5 | X0 | <_PA> | REAL | Eckpunkt 1 in der 1. Achse (abs) |
| 6 | Y0 | <_PO> | REAL | Eckpunkt 1 in der 2. Achse (abs) |
| 7 | X1 | <_LENG> | REAL | Eckpunkt 2 in der 1. Achse (abs/ink), siehe <_AMODE> |
| 8 | Y1 | <_WID> | REAL | Eckpunkt 2 in der 2. Achse (abs/ink), siehe <_AMODE> |
| 9 | DZ | <_MID> | REAL | maximale Tiefenzustellung |
| 10 | DXY | <_MIDA> | REAL | maximale Ebenenzustellung (Einheit, siehe <_AMODE>) |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 11 | UZ | <_FALD> | REAL | Schlichtaufmaß Tiefe | | |
| 12 | F | <_FFP1> | REAL | Bearbeitungsvorschub | | |
| 13 | | <_VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | Bearbeitung | |
| | | | | | 1 = | Schruppen |
| | | | | | 2 = | Schlichten |
| | | | | ZEHNER: | Bearbeitungsrichtung | |
| | | | | | 1 = | parallel zur 1. Achse, eine Richtung |
| | | | | | 2 = | parallel zur 2. Achse, eine Richtung |
| | | | | | 3 = | parallel zur 1. Achse, wechselnde Richtung |
| | | | | | 4 = | parallel zur 2. Achse, wechselnde Richtung |
| 14 | | <_LIM> | INT | Eingrenzungen | | |
| | | | | EINER: | Eingrenzung 1. Achse minus | |
| | | | | | 0 = | nein |
| | | | | | 1 = | ja |
| | | | | ZEHNER: | Eingrenzung 1. Achse plus | |
| | | | | | 0 = | nein |
| | | | | | 1 = | ja |
| | | | | HUNDERTER: | Eingrenzung 2. Achse minus | |
| | | | | | 0 = | nein |
| | | | | | 1 = | ja |
| | | | | TAUSENDER: | Eingrenzung 2. Achse plus | |
| | | | | | 0 = | nein |
| | | | | | 1 = | ja |
| 15 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/18/19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 16 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Endtiefe (<_DP>) | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |
| | | | | ZEHNER: | Einheit für Ebenenzustellung (<_MIDA>) | |
| | | | | | 0 = | mm |
| | | | | | 1 = | % vom Werkzeugdurchmesser |
| | | | | HUNDERTER: | reserviert | |
| | | | | TAUSENDER: | Länge der Fläche | |
| | | | | | 0 = | inkrementell |
| | | | | | 1 = | absolut |
| | | | | ZEHNTAUSENDER: | Breite der Fläche | |
| | | | | | 0 = | inkrementell |
| | | | | | 1 = | absolut |

### 3.25.1.12 CYCLE62 - Konturaufruf

#### Syntax

```
CYCLE62(<_KNAME>,<_TYPE>, <_LAB1>, <_LAB2>)
```

#### Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | PRG/CON | <_KNAME> | STRING [140] | Konturname oder Unterprogrammname, muss nicht programmiert werden bei _TYPE = 2 | | |
| 2 | | <_TYPE> | INT | Bestimmung der Kontureingabe | | |
| | | | | | 0 = | Unterprogramm |
| | | | | | 1 = | Konturname |
| | | | | | 2 = | Labels |
| | | | | | 3 = | Labels in Unterprogramm |
| 3 | LAB1 | <_LAB1> | STRING[32] | Label 1, Konturanfang | | |
| 4 | LAB2 | <_LAB2> | STRING[32] | Label 2, Konturende | | |

### 3.25.1.13 CYCLE63 - Konturtasche fräsen / Konturtasche Restmaterial / Konturzapfen fräsen / Konturzapfen Restmaterial

#### Syntax

```
CYCLE63(<_PRG>, <_VARI>, <_RP>, <_Z0>, <_SC>, <_Z1>, <_F>, <_FZ>,
<_DXY>, <_DZ>, <_UXY>, <_UZ>, <_CDIR>, <_XS>, <_YS>, <_ER>, <_EP>,
```

```
<_EW>, <_FS>, <_ZFS>, <_TR>, <_DR>, <_UMODE>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | PRG | <_PRG> | STRING [100] | Name Ausräumprogramm | | |
| 2 | | <_VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | Technologische Bearbeitung | |
| | | | | | 1 = | Schruppen |
| | | | | | 3 = | Schlichten Boden |
| | | | | | 4 = | Schlichten Rand |
| | | | | | 5 = | Anfasen |
| | | | | ZEHNER: | Zustellungsart | |
| | | | | | 0 = | Eintauchen mittig |
| | | | | | 1 = | Eintauchen helikal |
| | | | | | 2 = | Eintauchen pendelnd |
| | | | | HUNDERTER: | reserviert | |
| | | | | TAUSENDER: | Abhebemodus | |
| | | | | | 0 = | Abheben auf Rückzugsebene |
| | | | | | 1 = | Abheben auf Bezugspunkt + Sicherheitsabstand |
| | | | | ZEHNTAUSENDER: | Startpunkt bei Schruppen und Schlichten Boden | |
| | | | | | 0 = | auto |
| | | | | | 1 = | manuell |
| 3 | RP | <_RP> | REAL | Rückzugsebene (abs) | | |
| 4 | Z0 | <_Z0> | REAL | Bezugspunkt der Werkzeugachse (abs) | | |
| 5 | SC | <_SC> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) | | |
| 6 | Z1 | <_Z1> | REAL | Endtiefe, (siehe <_AMODE> EINER) | | |
| 7 | F | <_F> | REAL | Vorschub in der Ebene Schruppen/Schlichten | | |
| 8 | FZ | <_FZ> | REAL | Zustellvorschub Tiefe | | |
| 9 | DXY | <_DXY> | REAL | Zustellung Ebene - Einheit, (siehe <_AMODE> ZEHNER) | | |
| 10 | DZ | <_DZ> | REAL | Zustellung Tiefe | | |
| 11 | UXY | <_UXY> | REAL | Schlichtaufmaß Ebene | | |
| 12 | UZ | <_UZ> | REAL | Schlichtaufmaß Tiefe | | |
| 13 | | <_CDIR> | INT | Fräsrichtung | 0 = | Gleichlauf |
| | | | | | 1 = | Gegenlauf |
| 14 | XS | <_XS> | REAL | Startpunkt X, absolut | | |
| 15 | YS | <_YS> | REAL | Startpunkt Y, absolut | | |
| 16 | ER | <_ER> | REAL | Eintauchen helikal: Radius | | |
| 17 | EP | <_EP> | REAL | Eintauchen helikal: Steigung | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 18 | EW | <_EW> | REAL | Eintauchen pendelnd: maximaler Eintauchwinkel | | |
| 19 | FS | <_FS> | REAL | Fasenbreite (ink) bei Anfasen | | |
| 20 | ZFS | <_ZFS> | REAL | Eintauchtiefe Werkzeugspitze bei Anfasen, (siehe <_AMODE> HUNDERTER) | | |
| 21 | TR | <_TR> | STRING[32] | Referenzwerkzeugname bei Restmaterialbearbeitung | | |
| 22 | DR | <_DR> | INT | Referenzwerkzeug D-Nummer bei Restmaterialbearbeitung | | |
| 23 | | <_UMODE> | INT | reserviert | | |
| 24 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Auswahl Bearbeitung/nur Startpunktberechnung | |
| | | | | | 0 = | normale Bearbeitung (kein Kompatibilitätsmodus erforderlich) |
| | | | | | 1 = | normale Bearbeitung |
| | | | | | 2 = | reserviert |
| 25 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Technologiemodus | |
| | | | | | 1 = | Tasche |
| | | | | | 2 = | Zapfen |
| | | | | TAUSENDER: | Restmaterial bearbeiten | |
| | | | | | 0 = | nein |
| | | | | | 1 = | ja |
| | | | | ZEHNTAUSENDER: | Technologieskalierung innerhalb von Zyklenmasken (Seite 1172) | |
| | | | | | 0 = | Eingabe: komplett |
| | | | | | 1 = | Eingabe: einfach |
| | | | | HUNDERTTAUSENDER: | Programmname automatisch | |
| | | | | | 0 = | nein |
| | | | | | 1 = | ja |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 26 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Endtiefe (Z1) | |
| | | | | | 0 = | absolut (Kompatibilitätsmodus) |
| | | | | | 1 = | inkrementell |
| | | | | ZEHNER: | Einheit für Ebenenzustellung (DXY) | |
| | | | | | 0 = | mm |
| | | | | | 1 = | % vom Werkzeugdurchmesser |
| | | | | HUNDERTER: | Eintauchtiefe bei Anfasen (ZFS) | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |
| | | | | TAUSENDER: | --- | reserviert |

### 3.25.1.14 CYCLE64 - Konturtasche vorbohren

## Syntax

```
CYCLE64(<_PRG>, <_VARI>, <_RP>, <_Z0>, <_SC>, <_Z1>, <_F>, <_DXY>,
<_UXY>, <_UZ>, <_CDIR>, <_TR>, <_DR>, <_UMODE>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | PRG | <_PRG> | STRING [100] | Name Bohr-/Zentrierprogramm | | |
| 2 | | <_VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | reserviert | |
| | | | | TAUSENDER: | Abhebemodus | |
| | | | | | 0 = | Abheben auf Rückzugsebene |
| | | | | | 1 = | Abheben auf Bezugspunkt + Sicherheitsabstand |
| 3 | RP | <_RP> | REAL | Rückzugsebene (abs) | | |
| 4 | Z0 | <_Z0> | REAL | Bezugspunkt (abs) | | |
| 5 | SC | <_SC> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) | | |
| 6 | Z1 | <_Z1> | REAL | Bohr-/Zentriertiefe, (siehe <_AMODE> EINER) | | |
| 7 | F | <_F> | REAL | Vorschub Bohren/Zentrieren | | |
| 8 | DXY | <_DXY> | REAL | Zustellung Ebene - Einheit, (siehe <_AMODE> ZEHNER) | | |
| 9 | UXY | <_UXY> | REAL | Schlichtaufmaß Ebene | | |
| 10 | UZ | <_UZ> | REAL | Schlichtaufmaß Tiefe | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 11 | | <_CDIR> | INT | Fräsrichtung | 0 = | Gleichlauf |
| | | | | | 1 = | Gegenlauf |
| 12 | TR | <_TR> | STRING[20] | Referenzwerkzeugname | | |
| 13 | DR | <_DR> | INT | Referenzwerkzeug D-Nummer | | |
| 14 | | <_UMODE> | INT | reserviert | | |
| 15 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Auswahl Bearbeitung/nur Startpunktberechnung | |
| | | | | | 0 = | normale Bearbeitung (kein Kompatibilitätsmodus erforderlich) |
| | | | | | 1 = | normale Bearbeitung |
| | | | | | 2 = | reserviert |
| 25 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | Technologiemodus | |
| | | | | | 1 = | Vorbohren |
| | | | | | 2 = | Zentrieren |
| | | | | HUNDERTER: | --- | reserviert |
| | | | | TAUSENDER: | --- | reserviert |
| | | | | ZEHNTAUSENDER: | --- | reserviert |
| | | | | HUNDERTTAUSENDER: | Programmname automatisch | |
| | | | | | 0 = | nein |
| | | | | | 1 = | ja |
| 26 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Bohr-/Zentriertiefe Z1 | |
| | | | | | 0 = | absolut (Kompatibilitätsmodus) |
| | | | | | 1 = | inkrementell |
| | | | | ZEHNER: | Einheit für Ebenenzustellung (DXY) | |
| | | | | | 0 = | mm |
| | | | | | 1 = | % vom Werkzeugdurchmesser |

### 3.25.1.15 CYCLE70 - Gewindefräsen

## Syntax

```
CYCLE70(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DIATH>, <_H1>, <_FAL>,
<_PIT>, <_NT>, <_MID>, <_FFR>, <_TYPTH>, <_PA>, <_PO>, <_NSP>,
<_VARI>, <_PITA>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | RP | <_RTP> | REAL | Rückzugsebene (abs) | | |
| 2 | Z0 | <_RFP> | REAL | Bezugspunkt der Werkzeugachse (abs) | | |
| 3 | SC | <_SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) | | |
| 4 | Z1 | <_DP> | REAL | Gewindelänge (abs, ink), siehe <_AMODE><br>Auslauf am Bohrungsgrund berücksichtigen (minimum halbe Steigung) | | |
| 5 | ∅ | <_DIATH> | REAL | Nenndurchmesser des Gewindes | | |
| 6 | H1 | <_H1> | REAL | Gewindetiefe | | |
| 7 | U | <_FAL> | REAL | Schlichtaufmaß | | |
| 8 | P | <_PIT> | REAL | Gewindesteigung (Auswahl <_PITA>: mm, inch, MODUL, Gänge/Zoll) | | |
| 9 | NT | <_NT> | INT | Anzahl Zähne auf der Schneidplatte<br>Werkzeuglänge immer auf unteren Zahn bezogen! | | |
| 10 | DXY | <_MID> | REAL | Maximale Zustellung pro Schnitt<br><_MID> **>** <_H1>: alles mit einem Schnitt | | |
| 11 | F | <_FFR> | REAL | Fräsvorschub | | |
| 12 | | <_TYPTH> | INT | Gewindetyp | 0 = | Innengewinde |
| | | | | | 1 = | Außengewinde |
| 13 | X0 | <_PA> | REAL | Kreismittelpunkt, 1. Achse (abs) | | |
| 14 | Y0 | <_PO> | REAL | Kreismittelpunkt, 2. Achse (abs) | | |
| 15 | αS | <_NSP> | REAL | Startwinkel (mehrgängige Gewinde) | | |
| 16 | | <_VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | | |
| | | | | | 1 = | Schruppen |
| | | | | | 2 = | Schlichten |
| | | | | ZEHNER: | | |
| | | | | | 1 = | von oben nach unten |
| | | | | | 2 = | von unten nach oben |
| | | | | HUNDERTER: | | |
| | | | | | 0 = | Rechtsgewinde |
| | | | | | 1 = | Linksgewinde |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 17 | | <_PITA> | INT | Bewertung der Gewindesteigung | | |
| | | | | | 0 = | Kompatibilitätsmodus |
| | | | | | 1 = | Steigung in mm |
| | | | | | 2 = | Steigung in Gänge pro Zoll (TPI) |
| | | | | | 3 = | Steigung in inch |
| | | | | | 4 = | Steigung als MODUL |
| 18 | | <_PITM> | STRING[15] | String als Merker für Gewindesteigungseingabe (nur für die Oberfläche) | | |
| 19 | | <_PTAB> | STRING[20] | String für Gewindetabelle ("", "ISO", "BSW", "BSP", "UNC") (nur für die Oberfläche) | | |
| 20 | | <_PTABA> | STRING[20] | String für Auswahl in der Gewindetabelle (z.B. "M 10", "M 12", ...) (nur für die Oberfläche) | | |
| 21 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Bearbeitung/Startpunktberechnung | |
| | | | | | 0 = | Kompatibilitätsmodus |
| | | | | | 1 = | normale Bearbeitung |
| 22 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| 23 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Gewindelänge (<_DP>) | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |

### 3.25.1.16 CYCLE72 - Bahnfräsen

## Syntax

```
CYCLE72(<_KNAME>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_MID>, <_FAL>,
<_FALD>, <_FFP1>, <_FFD>, <_VARI>, <_RL>, <_AS1>, <_LP1>, <_FF3>,
<_AS2>, <_LP2>,<_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | | <_KNAME> | STRING [141] | Name des Kontur-Unterprogrammes | | |
| 2 | RP | <_RTP> | REAL | Rückzugsebene (abs) | | |
| 3 | Z0 | <_RFP> | REAL | Bezugspunkt der Werkzeugachse (abs) | | |
| 4 | SC | <_SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) | | |
| 5 | Z1 | <_DP> | REAL | Endpunkt, Endtiefe(abs/ink), siehe <_AMODE> | | |
| 6 | DZ | <_MID> | REAL | maximale Tiefenzustellung (ink, ohne Vorzeichen einzugeben) | | |
| 7 | UXY | <_FAL> | REAL | Schlichtaufmaß Ebene (ink), Aufmaß an der Randkontur | | |
| 8 | UZ | <_FALD> | REAL | Schlichtaufmaß Tiefe (ink), Aufmaß am Grund ohne Vorzeichen einzugeben) | | |
| 9 | FX | <_FFP1> | REAL | Vorschub an der Kontur | | |
| 10 | FZ | <_FFD> | REAL | Vorschub für Tiefenzustellung (oder räumliche Zustellung) | | |
| 11 | | <_VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | Bearbeitung | |
| | | | | | 1 = | Schruppen |
| | | | | | 2 = | Schlichten |
| | | | | | 5 = | Anfasen |
| | | | | ZEHNER: | | |
| | | | | | 0 = | Zwischenwege mit G0 |
| | | | | | 1 = | Zwischenwege mit G1 |
| | | | | HUNDERTER: | Rückzug am Konturende | |
| | | | | | 0 = | Rückzug am Konturende auf Bezugspunkt |
| | | | | | 1 = | Rückzug am Konturende auf Bezugspunkt + <_SDIS> |
| | | | | | 2 = | Rückzug am Konturende um <_SDIS> |
| | | | | | 3 = | kein Rückzug am Konturende, nächster Startpunkt wird mit Konturvorschub angefahren |
| | | | | TAUSENDER: | reserviert | |
| | | | | ZEHNTAUSENDER: | Kontur bearbeiten | |
| | | | | | 0 = | Kontur vorwärts bearbeiten |
| | | | | | 1 = | Kontur rückwärts bearbeiten<br>Einschränkungen bei rückwärts:<br>• max. 170 Konturelemente (inclusive Fasen oder Verrundungen)<br>• es werden nur Werte in der Ebene (X/Y) und F ausgewertet |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 12 | | <_RL> | INT | Bearbeitungsrichtung | | |
| | | | | | 40 = | mittig der Kontur (G40, An- und Abfahren: Gerade oder senkrecht) |
| | | | | | 41 = | links der Kontur (G41, An- und Abfahren: Gerade oder Kreis) |
| | | | | | 42 = | rechts der Kontur (G42, An- und Abfahren: Gerade oder Kreis) |
| 13 | | <_AS1> | INT | Kontur-Anfahrbewegung | | |
| | | | | EINER: | | |
| | | | | | 1 = | Gerade |
| | | | | | 2 = | Viertelkreis |
| | | | | | 3 = | Halbkreis |
| | | | | | 4 = | senkrecht An- und Abfahren |
| | | | | ZEHNER: | | |
| | | | | | 0 = | letzte Bewegung, in der Ebene |
| | | | | | 1 = | letzte Bewegung, räumlich |
| 14 | L1 | <_LP1> | REAL | Anfahrweg, oder Anfahrradius (ink, ohne Vorzeichen einzugeben) | | |
| 15 | FZ | <_FF3> | REAL | Vorschub für Zwischenwege (G94/G95 wie an Kontur) | | |
| 16 | | <_AS2> | INT | Kontur-Abfahrbewegung (nicht bei senkrecht An-/Abfahren) | | |
| | | | | EINER: | | |
| | | | | | 1 = | Gerade |
| | | | | | 2 = | Viertelkreis |
| | | | | | 3 = | Halbkreis |
| | | | | ZEHNER: | | |
| | | | | | 0 = | letzte Bewegung, in der Ebene |
| | | | | | 1 = | letzte Bewegung, räumlich |
| 17 | L2 | <_LP2> | REAL | Abfahrweg, oder Abfahrradius (ink, ohne Vorzeichen einzugeben) | | |
| 18 | | <_UMODE> | INT | reserviert | | |
| 19 | FS | <_FS> | REAL | Fasenbreite (Ink) | | |
| 20 | ZFS | <_ZFS> | REAL | Eintauchtiefe (Werkzeugspitze) bei Anfasen (abs/ink ), siehe <_AMODE> | | |
| 21 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Auswahl Bearbeitung/nur Startpunktberechnung | |
| | | | | | 0 = | Kompatibilitätsmodus |
| | | | | | 1 = | normale Bearbeitung |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 22 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/18/19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | Vorschubart: G-Gruppe (G94/G95) für Flächen- und Tiefenvorschub | |
| | | | | | 0 = | Kompatibilitätsmodus |
| | | | | | 1 = | G-Befehl wie vor Zyklusaufruf. G94/G95 für Flächen- und Tiefenvorschub gleich |
| | | | | TAUSENDER: | | |
| | | | | | 0 = | Kompatibilitätsmode: Konturname steht in <_KNAME> |
| | | | | | 1 = | Konturname wird im CYCLE62 programmiert und in _SC_CONT_NAME übergeben |
| 23 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Endpunkt Z1 (<_DP>) | |
| | | | | | 0 = | absolut (Kompatibilitätsmodus) |
| | | | | | 1 = | inkrementell |
| | | | | ZEHNER: | Einheit für Ebenenzustellung | |
| | | | | | 0 = | mm, inch |
| | | | | | 1 = | reserviert |
| | | | | HUNDERTER: | Eintauchtiefe bei Anfasen (<_ZFS>) | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |

**Hinweis**

Sind folgende Übergabeparameter indirekt (als Parameter) programmiert, wird die Eingabemaske nicht rückübersetzt:

```
<_VARI>, <_RL>, <_AS1>, <_AS2>, <_UMODE>, <_GMODE>, <_DMODE>,
<_AMODE>
```

### 3.25.1.17 CYCLE76 - Rechteckzapfen

## Syntax

```
CYCLE76(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_LENG>, <_WID>,
<_CRAD>, <_PA>, <_PO>, <_STA>, <_MID>, <_FAL>, <_FALD>, <_FFP1>,
```

```
<_FFD>, <_CDIR>, <_VARI>, <_AP1>, <_AP2>, <_FS>, <_ZFS>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | RP | <_RTP> | REAL | Rückzugsebene (abs) | | |
| 2 | Z0 | <_RFP> | REAL | Bezugspunkt der Werkzeugachse (abs) | | |
| 3 | SC | <_SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) | | |
| 4 | Z1 | <_DP> | REAL | Zapfentiefe(abs) | | |
| 5 | | <_DPR> | REAL | Zapfentiefe (ink), bezogen auf Z0 (ohne Vorzeichen einzugeben) | | |
| 6 | L | <_LENG> | REAL | Zapfenlänge, siehe <_GMODE> (ohne Vorzeichen einzugeben) | | |
| 7 | W | <_WID> | REAL | Zapfenbreite, siehe <_GMODE> (ohne Vorzeichen einzugeben) | | |
| 8 | R | <_CRAD> | REAL | Eckenradius des Zapfen (ohne Vorzeichen einzugeben) | | |
| 9 | X0 | <_PA> | REAL | Bezugspunkt Zapfen, 1. Achse der Ebene (abs) | | |
| 10 | Y0 | <_PO> | REAL | Bezugspunkt Zapfen, 2. Achse der Ebene (abs) | | |
| 11 | α0 | <_STA> | REAL | Drehwinkel, Winkel zwischen Längsachse (L) und 1. Achse der Ebene | | |
| 12 | DZ | <_MID> | REAL | maximale Tiefenzustellung (ink, ohne Vorzeichen einzugeben) | | |
| 13 | UXY | <_FAL> | REAL | Schlichtaufmaß Ebene (ink), Aufmaß an der Randkontur | | |
| 14 | UZ | <_FALD> | REAL | Schlichtaufmaß Tiefe (ink), Aufmaß am Grund (ohne Vorzeichen einzugeben) | | |
| 15 | FX | <_FFP1> | REAL | Vorschub an der Kontur | | |
| 16 | FZ | <_FFD> | REAL | Zustellvorschub Tiefe | | |
| 17 | | <_CDIR> | INT | Fräsrichtung (ohne Vorzeichen einzugeben) | | |
| | | | | EINER: | | |
| | | | | | 0 = | Gleichlauf |
| | | | | | 1 = | Gegenlauf |
| 18 | | <_VARI> | INT | Bearbeitung | | |
| | | | | EINER: | | |
| | | | | | 1 = | Schruppen |
| | | | | | 2 = | Schlichten |
| | | | | | 5 = | Anfasen |
| 19 | L1 | <_AP1> | REAL | Länge des Rohteilzapfens | | |
| 20 | W1 | <_AP2> | REAL | Breite des Rohteilzapfens | | |
| 21 | FS | <_FS> | REAL | Fasenbreite (ink) | | |
| 22 | ZFS | <_ZFS> | REAL | Eintauchtiefe (Werkzeugspitze) bei Anfasen (abs, ink), siehe <_AMODE> | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 23 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Auswahl Bearbeitung oder nur Startpunktberechnung | |
| | | | | | 0 = | Kompatibilitätsmodus |
| | | | | | 1 = | normale Bearbeitung |
| | | | | TAUSENDER: | Bemaßung des Zapfens über Mitte oder Ecke | |
| | | | | | 0 = | Kompatibilitätsmodus |
| | | | | | 1 = | Bemaßung über Mitte |
| | | | | | 2 = | Bemaßung Eckpunkt, Zapfen +L +W |
| | | | | | 3 = | Bemaßung Eckpunkt, Zapfen -L +W |
| | | | | | 4 = | Bemaßung Eckpunkt, Zapfen +L -W |
| | | | | | 5 = | Bemaßung Eckpunkt, Zapfen -L -W |
| | | | | ZEHNTAUSENDER: | Komplettbearbeitung oder Nachbearbeitung | |
| | | | | | 0 = | Kompatibilitätsmodus |
| | | | | | 1 = | Komplettbearbeitung |
| | | | | | 2 = | Nachbearbeitung |
| 24 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/18/19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | --- | reserviert |
| | | | | HUNDERTER: | --- | reserviert |
| | | | | TAUSENDER: | --- | reserviert |
| | | | | ZEHNTAUSENDER: | Technologieskalierung innerhalb von Zyklenmasken (Seite 1172) | |
| | | | | | 0 = | Eingabe: komplett |
| | | | | | 1 = | Eingabe: einfach |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 25 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Endtiefe Z1 (DP) | |
| | | | | | 0 = | Kompatibilität |
| | | | | | 1 = | inkrementell |
| | | | | | 2 = | absolut |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Eintauchtiefe bei Anfasen (ZFS) | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |

### 3.25.1.18 CYCLE77 - Kreiszapfen

## Syntax

```
CYCLE77(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_CDIAM>, <_PA>,
<_PO>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>, <_VARI>,
<_AP1>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | RP | <_RTP> | REAL | Rückzugsebene (abs) | | |
| 2 | Z0 | <_RFP> | REAL | Bezugspunkt der Werkzeugachse (abs) | | |
| 3 | SC | <_SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) | | |
| 4 | Z1 | <_DP> | REAL | Zapfentiefe(abs) | | |
| 5 | | <_DPR> | REAL | Zapfentiefe (ink), bezogen auf Z0 (ohne Vorzeichen einzugeben) | | |
| 6 | ∅ | <_CDIAM> | REAL | Durchmesser des Zapfens (ohne Vorzeichen einzugeben) | | |
| 7 | X0 | <_PA> | REAL | Bezugspunkt Zapfen, 1. Achse der Ebene (abs) | | |
| 8 | Y0 | <_PO> | REAL | Bezugspunkt Zapfen, 2. Achse der Ebene (abs) | | |
| 9 | DZ | <_MID> | REAL | maximale Tiefenzustellung (ink, ohne Vorzeichen einzugeben) | | |
| 10 | UXY | <_FAL> | REAL | Schlichtaufmaß Ebene (ink), Aufmaß an der Randkontur | | |
| 11 | UZ | <_FALD> | REAL | Schlichtaufmaß Tiefe (ink), Aufmaß am Grund (ohne Vorzeichen einzugeben) | | |
| 12 | FX | <_FFP1> | REAL | Vorschub an der Kontur | | |
| 13 | FZ | <_FFD> | REAL | Zustellvorschub Tiefe | | |
| 14 | | <_CDIR> | INT | Fräsrichtung (ohne Vorzeichen einzugeben) | | |
| | | | | EINER: | | |
| | | | | | 0 = | Gleichlauf |
| | | | | | 1 = | Gegenlauf |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 15 | | <_VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | Bearbeitung | |
| | | | | | 1 = | Schruppen bis auf Schlichtaufmaß |
| | | | | | 2 = | Schlichten (Aufmaß X/Y/Z = 0) |
| | | | | | 5 = | Anfasen |
| 16 | ∅1 | <_AP1> | REAL | Durchmesser des Rohteilzapfens | | |
| 17 | FS | <_FS> | REAL | Fasenbreite (ink) | | |
| 18 | ZFS | <_ZFS> | REAL | Eintauchtiefe (Werkzeugspitze) bei Anfasen (abs/ink) siehe <_AMODE> | | |
| 19 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Auswahl Bearbeitung/nur Startpunktberechnung | |
| | | | | | 0 = | Kompatibilitätsmodus |
| | | | | | 1 = | normale Bearbeitung |
| | | | | TAUSENDER: | reserviert | |
| | | | | ZEHNTAUSENDER: | Komplettbearbeitung/Nachbearbeitung | |
| | | | | | 0 = | Kompatibilitätsmodus (<_AP1> wie bisher behandeln) |
| | | | | | 1 = | Komplettbearbeitung |
| | | | | | 2 = | Nachbearbeitung |
| 20 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/18/19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | --- | reserviert |
| | | | | HUNDERTER: | --- | reserviert |
| | | | | TAUSENDER: | --- | reserviert |
| | | | | ZEHNTAUSENDER: | Technologieskalierung innerhalb von Zyklenmasken (Seite 1172) | |
| | | | | | 0 = | Eingabe: komplett |
| | | | | | 1 = | Eingabe: einfach |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 21 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Endtiefe Z1 (DP) | |
| | | | | | 0 = | absolut (Kompatibilitätsmodus) |
| | | | | | 1 = | inkrementell |
| | | | | | 2 = | absolut |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Eintauchtiefe bei Anfasen (ZFS) | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |

### 3.25.1.19 CYCLE78 - Bohrgewinde fräsen

## Syntax

```
CYCLE78(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_ADPR>, <_FDPR>, <_LDPR>,
<_DIAM>, <_PIT>, <_PITA>, <_DAM>, <_MDEP>, <_VARI>, <_CDIR>, <_GE>,
<_FFD>, <_FRDP>, <_FFR>, <_FFP2>, <_FFA>, <_PITM>, <_PTAB>,
<_PTABA>, <_GMODE>, <_DMODE>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | RP | <_RTP> | REAL | Rückzugsebene (abs) | | |
| 2 | Z0 | <_RFP> | REAL | Bezugspunkt der Werkzeugachse (abs) | | |
| 3 | SC | <_SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) | | |
| 4 | Z1 | <_DP> | REAL | Endbohrtiefe (abs/ink), siehe <_AMODE> | | |
| 5 | | <_ADPR> | REAL | Anbohrtiefe mit reduziertem Bohrvorschub (ink), mit <_VARI> ZEHNTAUSENDER wirksam | | |
| 6 | D | <_FDPR> | REAL | maximale Tiefenzustellung (ink)<br>D ≥ Z1 ⇒ eine Zustellung auf Endbohrtiefe<br>D < Z1 ⇒ Tiefbohrzyklus mit mehreren Zustellungen und Entspanen | | |
| 7 | ZR | <_LDPR> | REAL | Restbohrtiefe beim Durchbohren (ink), mit Vorschub FR | | |
| 8 | ∅ | <_DIAM> | REAL | Nenndurchmesser des Gewindes | | |
| 9 | P | <_PIT> | REAL | Gewindesteigung als Zahlenwert | | |
| 10 | | <_PITA> | INT | Bewertung der Gewindesteigung P | | |
| | | | | | 1 = | Steigung in mm/U |
| | | | | | 2 = | Steigung in Gänge/Zoll |
| | | | | | 3 = | Steigung in inch/U |
| | | | | | 4 = | Steigung als MODUL |
| 11 | DF | <_DAM> | REAL | Betrag/Prozentsatz für jede weitere Zustellung (Degression), siehe <_AMODE> | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 12 | V1 | <_MDEP> | REAL | minimale Zustellung (ink), nur bei Degression wirksam | | |
| 13 | | <_VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | Entspanen vor Gewindefräsen | |
| | | | | | 0 = | kein Entspanen vor Gewindefräsen (wirkt nur auf Endbohrtiefe) |
| | | | | | 1 = | Entspanen vor Gewindefräsen (wirkt nur auf Endbohrtiefe) |
| | | | | HUNDERTER: | Rechts-/Linksgewinde | |
| | | | | | 0 = | Rechtsgewinde |
| | | | | | 1 = | Linksgewinde |
| | | | | TAUSENDER: | Restbohrtiefe mit Bohrvorschub | |
| | | | | | 0 = | keine Restbohrtiefe mit Bohrvorschub FR |
| | | | | | 1 = | Restbohrtiefe mit Bohrvorschub FR |
| | | | | ZEHNTAUSENDER: | Anbohren mit verringertem Vorschub | |
| | | | | | 0 = | kein Anbohren mit verringertem Vorschub |
| | | | | | 1 = | Anbohren mit verringertem Vorschub<br>Anbohrvorschub = 0,3 F1, wenn F1 < 0,15 mm/U<br>Anbohrvorschub = 0,1 mm/U, wenn F1 ≥0,15 mm/U |
| 14 | | <_CDIR> | INT | Fräsrichtung | 0 = | Gleichlauf |
| | | | | | 1 = | Gegenlauf |
| | | | | | 4 = | Gegenlauf + Gleichlauf (Kombination Schruppen + Schlichten) |
| 15 | Z2 | <_GE> | REAL | Rückzugsbetrag vor Gewindefräsen (ink) | | |
| 16 | F1 | <_FFD> | REAL | Bohrvorschub (mm/min bzw. in/min oder mm/U) | | |
| 17 | FR | <_FRDP> | REAL | Bohrvorschub für Restbohrtiefe (mm/min oder mm/U) | | |
| 18 | F2 | <_FFR> | REAL | Vorschub für Gewindefräsen (mm/min oder mm/Zahn) | | |
| 19 | FS | <_FFP2> | REAL | Schlichtvorschub für <_CDIR> = 4 (mm/min oder mm/Zahn) | | |
| 20 | | <_FFA> | INT | Bewertung Vorschübe | | |
| | | | | EINER: | Bohrvorschub F1 | |
| | | | | ZEHNER: | Bohrvorschub für Restbohrtiefe FR | |
| | | | | HUNDERTER: | Vorschub für Gewindefräsen F2 | |
| | | | | TAUSENDER: | Schlichtvorschub FS | |
| 21 | | <_PITM> | STRING[15] | String als Merker für Gewindesteigungseingabe (nur für die Oberfläche)[1)] | | |
| 22 | | <_PTAB> | STRING[20] | String für Gewindetabelle ("", "ISO", "BSW", "BSP", "UNC") (nur für die Oberfläche)[1)] | | |
| 23 | | <_PTABA> | STRING[20] | String für Auswahl in der Gewindetabelle (z.B. "M 10", "M 12", ...) (nur für die Oberfläche)[1)] | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 24 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte), reserviert | | |
| 25 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/18/19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| 26 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Bohrtiefe = Endbohrtiefe Z1 abs/ink | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |
| | | | | ZEHNER: | Betrag/Prozentsatz DF für jede weitere Zustellung (Degression) | |
| | | | | | 0 = | Betrag |
| | | | | | 1 = | Prozentsatz (0.001 bis 100 %) |

**Hinweis**

[1)] Die Parameter 21, 22 und 23 werden nur bei der Gewindeauswahl in den Gewindetabellen der Eingabemaske verwendet. Ein Zugriff auf die Gewindetabellen über Zyklusdefinition bei Zykluslaufzeit ist nicht möglich.

### 3.25.1.20 CYCLE79 - Mehrkant

## Syntax

```
CYCLE79(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_NUM>, <_SWL>, <_PA>,
<_PO>, <_STA>, <_RC>, <_AP1>, <_MIDA>, <_MID>, <_FAL>, <_FALD>,
<_FFP1>, <_CDIR>, <_VARI>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung |
|---|---|---|---|---|
| 1 | RP | <_RTP> | REAL | Rückzugsebene (abs) |
| 2 | Z0 | <_RFP> | REAL | Bezugspunkt der Werkzeugachse (abs) |
| 3 | SC | <_SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) |
| 4 | Z1 | <_DP> | REAL | Mehrkanttiefe (abs/ink), siehe <_AMODE> |
| 5 | N | <_NUM> | INT | Anzahl der Kanten (1...n) |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 6 | SW/L | <_SWL> | REAL | Schlüsselweite oder Kantenlänge (je nach <_VARI>)<br>("SW" bei Schlüsselweite, "L" bei Kantenlänge)<br>Schlüsselweite nur bei gerader Anzahl Kanten, und Einkant | | |
| 7 | X0 | <_PA> | REAL | Zapfenbezugspunkt, 1. Achse (abs) | | |
| 8 | Y0 | <_PO> | REAL | Zapfenbezugspunkt, 2. Achse (abs) | | |
| 9 | α0 | <_STA> | REAL | Drehwinkel Kantenmitte gegen 1. Achse (X-Achse) | | |
| 10 | R1/FS1 | <_RC> | REAL | Eckenverrundung bei <_NUM> > 2 (Radius/Fase, siehe <_AMODE>)<br>(ink, ohne Vorzeichen einzugeben)<br>("R1" bei Radius, "FS1" bei Fase) | | |
| 11 | ∅ | <_AP1> | REAL | Rohdurchmesser des Zapfens | | |
| 12 | DXY | <_MIDA> | REAL | maximale Zustellbreite (Einheit, siehe <_AMODE>) | | |
| 13 | DZ | <_MID> | REAL | maximale Tiefenzustellung | | |
| 14 | UXY | <_FAL> | REAL | Schlichtaufmaß Ebene | | |
| 15 | UZ | <_FALD> | REAL | Schlichtaufmaß Tiefe | | |
| 16 | F | <_FFP1> | REAL | Bearbeitungsvorschub | | |
| 17 | | <_CDIR> | INT | Fräsrichtung | 0 = | Gleichlauf |
| | | | | | 1 = | Gegenlauf |
| 18 | | <_VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | Bearbeitung | |
| | | | | | 1 = | Schruppen |
| | | | | | 2 = | Schlichten |
| | | | | | 3 = | Schlichten Rand |
| | | | | | 5 = | Anfasen |
| | | | | ZEHNER: | Schlüsselweite oder Kantenlänge | |
| | | | | | 0 = | Schlüsselweite |
| | | | | | 1 = | Kantenlänge |
| 19 | FS | <_FS> | REAL | Fasenbreite (ink) | | |
| 20 | ZFS | <_ZFS> | REAL | Eintauchtiefe (Werkzeugspitze) bei Anfasen (abs/ink), siehe <_AMODE> | | |
| 21 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Auswahl Bearbeitung oder nur Startpunktberechnung | |
| | | | | | 1 = | normale Bearbeitung |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 22 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/18/19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | --- | reserviert |
| | | | | HUNDERTER: | --- | reserviert |
| | | | | TAUSENDER: | --- | reserviert |
| | | | | ZEHNTAUSENDER: | Technologieskalierung innerhalb von Zyklenmasken (Seite 1172) | |
| | | | | | 0 = | Eingabe: komplett |
| | | | | | 1 = | Eingabe: einfach |
| 23 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Endtiefe (<_DP>) | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |
| | | | | ZEHNER: | Einheit für Ebenenzustellung (<_MIDA>) | |
| | | | | | 0 = | mm |
| | | | | | 1 = | % vom Werkzeugdurchmesser |
| | | | | HUNDERTER: | Eintauchtiefe bei Anfasen (<_ZFS>) | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |
| | | | | TAUSENDER: | Eckenverrundung (<_RC>) | |
| | | | | | 0 = | Radius |
| | | | | | 1 = | Fase |

### 3.25.1.21 CYCLE81 - Bohren, Zentrieren

## Syntax

```
CYCLE81(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung |
|---|---|---|---|---|
| 1 | RP | <RTP> | REAL | Rückzugsebene (abs) |
| 2 | Z0 | <RFP> | REAL | Bezugspunkt (abs) |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 3 | SC | <SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) | | |
| 4 | Z1/∅ | <DP> | REAL | Bohrtiefe (abs) / Durchmesser der Zentrierung (abs), siehe <_GMODE> | | |
| 5 | Z1 | <DPR> | REAL | Bohrtiefe (ink) | | |
| 6 | DT | <DTB> | REAL | Verweilzeit auf Endbohrtiefe, siehe <_AMODE> | | |
| 7 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | Zentrierung bezogen auf die Tiefe / den Durchmesser | |
| | | | | | 0 = | Kompatibilität, Tiefe |
| | | | | | 1 = | Durchmesser |
| 8 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| 9 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Bohrtiefe Z1 (abs/ink) | |
| | | | | | 0 = | Kompatibilität, aus Programmierung DP/DPR |
| | | | | | 1 = | inkrementell |
| | | | | | 2 = | absolut |
| | | | | ZEHNER: | Verweilzeit auf Endbohrtiefe DT in Sekunden/Umdrehungen | |
| | | | | | 0 = | Kompatibilität, aus Vorzeichen von DTB (> 0 Sekunden oder < 0 Umdrehungen) |
| | | | | | 1 = | in Sekunden |
| | | | | | 2 = | in Umdrehungen |

### 3.25.1.22 CYCLE82 - Bohren, Plansenken

## Syntax

```
CYCLE82(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>,
<_DMODE>, <_AMODE>, <_VARI>, <S_ZA>, <S_FA>, <S_ZD>, <S_FD>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | RP | <RTP> | REAL | Rückzugsebene (abs) | | |
| 2 | Z0 | <RFP> | REAL | Bezugspunkt (abs) | | |
| 3 | SC | <SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) | | |
| 4 | Z1 | <DP> | REAL | Bohrtiefe (abs), siehe <_AMODE> | | |
| 5 | Z1 | <DPR> | REAL | Bohrtiefe (ink), siehe <_AMODE> | | |
| 6 | DT | <DTB> | REAL | Verweilzeit auf Endbohrtiefe, siehe <_AMODE> | | |
| 7 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | Bohrtiefe bezogen auf Spitze/Schaft | |
| | | | | | 0 = | Kompatibilität, Spitze |
| | | | | | 1 = | Schaft |
| 8 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | reserviert | |
| | | | | TAUSENDER: | reserviert | |
| | | | | ZEHNTAUSENDER: | Technologieskalierung innerhalb von Zyklenmasken (Seite 1172) | |
| | | | | | 0 = | Eingabe: komplett |
| | | | | | 1 = | Eingabe: einfach |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 9 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Bohrtiefe Z1 (abs/ink) | |
| | | | | | 0 = | Kompatibilität aus Programmierung DP/DPR |
| | | | | | 1 = | inkrementell |
| | | | | | 2 = | absolut |
| | | | | ZEHNER: | Verweilzeit DT auf Endbohrtiefe in Sekunden/Umdrehungen | |
| | | | | | 0 = | Kompatibilität, aus Vorzeichen von DT (> 0 Sekunden / < 0 Umdrehungen) |
| | | | | | 1 = | in Sekunden |
| | | | | | 2 = | in Umdrehungen |
| | | | | HUNDERTER: | Anbohrtiefe ZA abs/ink | |
| | | | | | 0 = | inkrementell |
| | | | | | 1 = | absolut |
| | | | | TAUSENDER: | Bewertung Anbohrvorschub | |
| | | | | | 0 = | in % vom Bohrvorschub |
| | | | | | 1 = | F/min |
| | | | | | 2 = | F/U |
| | | | | ZEHNTAUSENDER: | Restbohrtiefe ZD abs/ink | |
| | | | | | 0 = | inkrementell |
| | | | | | 1 = | absolut |
| | | | | HUNDERTTAUSENDER: | Bewertung Restbohrvorschub | |
| | | | | | 0 = | in % vom Bohrvorschub |
| | | | | | 1 = | F/min |
| | | | | | 2 = | F/U |
| 10 | | <_VARI> | INT | Bearbeitungsart Anbohren/Durchbohren | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | reserviert | |
| | | | | TAUSENDER: | Durchbohren | |
| | | | | | 0 = | Durchbohren "Nein" |
| | | | | | 1 = | Durchbohren "Ja" |
| | | | | ZEHNTAUSENDER: | Anbohren | |
| | | | | | 0 = | Anbohren "Nein" |
| | | | | | 1 = | Anbohren "Ja" |
| 11 | ZA | <S_ZA> | REAL | Anbohrtiefe inkrementell bezogen auf Bezugspunkt oder absolut (siehe <_AMODE> HUNDERTER) | | |
| 12 | FA | <S_FA> | REAL | Anbohrvorschub als Wert oder in % (in Verbindung mit <_AMODE> TAUSENDER) | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung |
|---|---|---|---|---|
| 13 | ZD | <S_ZD> | REAL | Restbohrtiefe inkrementell bezogen auf Endbohrtiefe oder absolut (in Verbindung mit <_AMODE> ZEHNTAUSENDER) |
| 14 | FD | <S_FD> | REAL | Restbohrvorschub als Wert oder in % (in Verbindung mit <_AMODE> HUNDERTTAUSENDER) |

### 3.25.1.23 CYCLE83 - Tieflochbohren 1

## Syntax

```
CYCLE83(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <FDEP>, <FDPR>, <_DAM>,
<DTB>, <DTS>, <FRF>, <VARI>, <_AXN>, <_MDEP>, <_VRT>, <_DTD>,
<_DIS1>, <_GMODE>, <_DMODE>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | RP | <RTP> | REAL | Rückzugsebene (abs) | | |
| 2 | Z0 | <RFP> | REAL | Bezugspunkt (abs) | | |
| 3 | SC | <SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) | | |
| 4 | Z1 | <DP> | REAL | Endbohrtiefe (abs), siehe <_AMODE> | | |
| 5 | Z1 | <DPR> | REAL | Endbohrtiefe (ink), siehe <_AMODE> | | |
| 6 | D | <FDEP> | REAL | 1. Bohrtiefe (abs), siehe <_AMODE> | | |
| 7 | D | <FDPR> | REAL | 1. Bohrtiefe (ink), siehe <_AMODE> | | |
| 8 | DF | <_DAM> | REAL | Degressionsbetrag/Prozentsatz für jede weitere Zustellung, siehe <_AMODE> | | |
| 9 | DTB | <DTB> | REAL | Verweilzeit auf Bohrtiefe, siehe <_AMODE> | | |
| 10 | DTS | <DTS> | REAL | Verweilzeit am Anfangspunkt (nur bei Entspanen), siehe <_AMODE> | | |
| 11 | FD1 | <FRF> | REAL | Prozentsatz für den Vorschub bei der ersten Zustellung, siehe <_AMODE> | | |
| 12 | | <VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | Spänebrechen / Entspanen | |
| | | | | | 0 = | Spänebrechen |
| | | | | | 1 = | Entspanen |
| 13 | | <_AXN> | INT | Werkzeugachse | | |
| | | | | | 0 = | 3. Geometrieachse |
| | | | | | 1 = | 1. Geometrieachse |
| | | | | | 2 = | 2. Geometrieachse |
| | | | | | > 2 | 3. Geometrieachse |
| 14 | V1 | <_MDEP> | REAL | minimale Zustellung (nur bei Prozentsatz für Degression) | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 15 | V2 | <_VRT> | REAL | Rückzugsbetrag nach jeder Bearbeitung (nur bei Spänebrechen) | | |
| | | | | | > 0 | variabler Rückzugsbetrag |
| | | | | | 0 = | Standardwert 1 mm |
| 16 | DT | <_DTD> | REAL | Verweilzeit auf Endbohrtiefe, siehe <_AMODE> | | |
| 17 | V3 | <_DIS1> | REAL | Vorhalteabstand (nur bei Entspanen), siehe <_AMODE> | | |
| 18 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | Bohrtiefe bezogen auf Spitze/Schaft | |
| | | | | | 0 = | Spitze |
| | | | | | 1 = | Schaft |
| 19 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | --- | reserviert |
| | | | | HUNDERTER: | --- | reserviert |
| | | | | TAUSENDER: | --- | reserviert |
| | | | | ZEHNTAUSENDER: | Technologieskalierung innerhalb von Zyklenmasken (Seite 1172) | |
| | | | | | 0 = | Eingabe: komplett |
| | | | | | 1 = | Eingabe: einfach |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 20 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Bohrtiefe = Endbohrtiefe Z1 (abs/ink) | |
| | | | | | 0 = | Kompatibilität, aus Programmierung <DP>/<DPR> |
| | | | | | 1 = | inkrementell |
| | | | | | 2 = | absolut |
| | | | | ZEHNER: | Verweilzeit auf Bohrtiefe DTB in Sekunden/Umdrehungen | |
| | | | | | 0 = | Kompatibilität aus Vorzeichen von DTB (> 0 Sekunden oder < 0 Umdrehungen) |
| | | | | | 1 = | in Sekunden |
| | | | | | 2 = | in Umdrehungen |
| | | | | HUNDERTER: | Verweilzeit am Anfangspunkt von DTS in Sekunden/Umdrehungen | |
| | | | | | 0 = | Kompatibilität aus Vorzeichen von DTS (> 0 Sekunden oder < 0 Umdrehungen) |
| | | | | | 1 = | in Sekunden |
| | | | | | 2 = | in Umdrehungen |
| | | | | TAUSENDER: | Verweilzeit auf Endbohrtiefe DTD in Sekunden/Umdrehungen | |
| | | | | | 0 = | Kompatibilität aus Vorzeichen von DTD (> 0 Sekunden oder < 0 Umdrehungen) |
| | | | | | 1 = | in Sekunden |
| | | | | | 2 = | in Umdrehungen |
| | | | | ZEHNTAUSENDER: | 1. Bohrtiefe D (abs/ink) | |
| | | | | | 0 = | Kompatibilität, aus Programmierung <FDEPF>/<DPR> |
| | | | | | 1 = | inkrementell |
| | | | | | 2 = | absolut |
| | | | | HUNDERTTAUSENDER: | Degressionsbetrag/Prozentsatz <_DAM> für jede weitere Zustellung | |
| | | | | | 0 = | Kompatibilität, aus Vorzeichen von <_DAM> (> 0 Degressionsbetrag oder < 0 Faktor 0,001 bis 1,0) |
| | | | | | 1 = | Degressionsbetrag |
| | | | | | 2 = | Prozentsatz (0,001 bis 100 %) |
| | | | | EINEMILLION: | Vorhalteabstand V3 automatisch/manuell | |
| | | | | | 0 = | Kompatibilität aus Vorzeichen von <_DIS1> (= 0 automatisch oder > 0 manuell) |
| | | | | | 1 = | automatisch (wird im Zyklus berechnet) |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| | | | | | 2 = | manuell (programmierter Wert) |
| | | | | ZEHNMILLIONEN: | Vorschubfaktor für erste Zustellung <FRF> als Faktor/Prozentsatz | |
| | | | | | 0 = | Kompatibilität, als Faktor (0,001 bis 1,0, FRF = 0 bedeutet 100 %) |
| | | | | | 1 = | Prozentsatz (0,001 bis 999,999 %) |

### 3.25.1.24 CYCLE84 - Gewindebohren ohne Ausgleichsfutter

## Syntax

```
CYCLE84(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDAC>, <MPIT>,
<PIT>, <POSS>, <SST>, <SST1>, <_AXN>, <_PITA>, <_TECHNO>, <_VARI>,
<_DAM>, <_VRT>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | RP | <RTP> | REAL | Rückzugsebene (abs) | | |
| 2 | Z0 | <RFP> | REAL | Bezugspunkt (abs) | | |
| 3 | SC | <SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) | | |
| 4 | Z1 | <DP> | REAL | Bohrtiefe = Endbohrtiefe (abs), siehe <_AMODE> | | |
| 5 | Z1 | <DPR> | REAL | Bohrtiefe = Endbohrtiefe (ink), siehe <_AMODE> | | |
| 6 | DT | <DTB> | REAL | Verweilzeit auf Bohrtiefe in Sekunden | | |
| 7 | SDE | <SDAC> | INT | Drehrichtung nach Zyklusende | | |
| 8 | | <MPIT> | REAL | Gewindegröße nur für "ISO metrisch" (Steigung wird intern während der Laufzeit berechnet) | | |
| 9 | P | <PIT> | REAL | Gewindesteigung als Wert, Maßeinheit siehe <_PITA> | | |
| 10 | αS[1)] | <POSS> | REAL | Spindelposition für orientierten Spindelstop | | |
| 11 | S | <SST> | REAL | Spindeldrehzahl für Gewindebohren | | |
| 12 | SR | <SST1> | REAL | Spindeldrehzahl für Rückzug | | |
| 13 | | <_AXN> | INT | Bohrachse | 0 = | 3. Geometrieachse |
| | | | | | 1 = | 1. Geometrieachse |
| | | | | | 2 = | 2. Geometrieachse |
| | | | | | ≥ 3 = | 3. Geometrieachse |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | | |
|---|---|---|---|---|---|---|---|
| 14 | | <_PITA> | INT | Maßeinheit der Gewindesteigung (Auswertung <PIT> und <MPIT>) | | | |
| | | | | | 0 = | Steigung in mm | - Auswertung <MPIT>/<PIT> |
| | | | | | 1 = | Steigung in mm | - Auswertung <PIT> |
| | | | | | 2 = | Steigung in TPI | - Auswertung <PIT> (Gewindegänge pro inch) |
| | | | | | 3 = | Steigung in inch | - Auswertung <PIT> |
| | | | | | 4 = | MODUL | - Auswertung <PIT> |
| 15 | | <_TECHNO> | INT | Technologie[1)] | | | |
| | | | | EINER: | Genauhaltverhalten | | |
| | | | | | 0 = | Genauhaltverhalten wie vor Zyklusaufruf aktiv | |
| | | | | | 1 = | Genauhalt G601 | |
| | | | | | 2 = | Genauhalt G602 | |
| | | | | | 3 = | Genauhalt G603 | |
| | | | | ZEHNER: | Vorsteuerung | | |
| | | | | | 0 = | mit/ohne Vorsteuerung wie vor Zyklusaufruf aktiv | |
| | | | | | 1 = | mit Vorsteuerung FFWON | |
| | | | | | 2 = | ohne Vorsteuerung FFWOF | |
| | | | | HUNDERTER: | Beschleunigung | | |
| | | | | | 0 = | SOFT/BRISK/DRIVE wie vor Zyklusaufruf aktiv | |
| | | | | | 1 = | mit Ruckbegrenzung SOFT | |
| | | | | | 2 = | ohne Ruckbegrenzung BRISK | |
| | | | | | 3 = | reduzierte Beschleunigung DRIVE | |
| | | | | TAUSENDER: | MCALL Spindelbetrieb | | |
| | | | | | 0 = | bei MCALL Spindelbetrieb wieder aktivieren | |
| | | | | | 1 = | bei MCALL in Lageregelung bleiben | |
| 16 | | <_VARI> | INT | Bearbeitungsart | | | |
| | | | | EINER: | 0 = | 1 Schnitt | |
| | | | | | 1 = | Spänebrechen (Tieflochgewindebohren) | |
| | | | | | 2 = | Entspanen (Tieflochgewindebohren) | |
| | | | | TAUSENDER: | ISO/SIEMENS Modus für Eingabemaske nicht relevant | | |
| | | | | | 0 = | Aufruf aus ISO-Kompatibilität | |
| | | | | | 1 = | Aufruf aus SIEMENS-Kontext | |
| 17 | D | <_DAM> | REAL | maximale Tiefenzustellung (nur bei Entspanen/Spänebrechen) | | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 18 | V2 | <_VRT> | REAL | Rückzugsbetrag nach jeder Bearbeitung (nur bei Spänebrechen), siehe <_AMODE> | | |
| 19 | | <_PITM> | STRING[15] | String als Merker für Gewindesteigungseingabe[2)] | | |
| 20 | | <_PTAB> | STRING[5] | String für Gewindetabelle ("", "ISO", "BSW", "BSP", "UNC")[2)] | | |
| 21 | | <_PTABA> | STRING[20] | String für Auswahl in der Gewindetabelle (z. B. "M 10", "M 12", ...)[2)] | | |
| 22 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | reserviert | |
| 23 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | reserviert | |
| | | | | TAUSENDER: | Kompatibilitätsmode (nur für Eingabemaske zu Rückübersetzen), wenn MD 52216 Bit0 = 1[1)] | |
| | | | | | 0 = | Technologieparameter werden angezeigt (Kompatibilität): TECHNO-Parameter wirken |
| | | | | | 1 = | Technologieparameter werden nicht angezeigt: Technologie "wie vor Zyklusaufruf programmiert" wirkt |
| | | | | ZEHNTAUSENDER: | Technologieskalierung innerhalb von Zyklenmasken (Seite 1172) | |
| | | | | | 0 = | Eingabe: komplett |
| | | | | | 1 = | Eingabe: einfach |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 24 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Bohrtiefe = Endbohrtiefe Z1 (abs/ink) | |
| | | | | | 0 = | Kompatibilität, aus Programmierung <DP>/<DPR> |
| | | | | | 1 = | inkrementell |
| | | | | | 2 = | absolut |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | reserviert | |
| | | | | TAUSENDER: | Gewindedrehrichtung rechts/links | |
| | | | | | 0 = | Kompatibilität, aus Vorzeichen PIT/MPIT |
| | | | | | 1 = | rechts |
| | | | | | 2 = | links |
| | | | | ZEHNTAUSENDER: | reserviert | |
| | | | | HUNDERTTAUSENDER: | reserviert | |
| | | | | EINEMILLION: | Rückzugsbetrag nach jeder Bearbeitung V2 manuell/automatisch | |
| | | | | | 0 = | Kompatibilität, aus Programmierung <_VRT> (> 0 variabler Wert oder ≤ 0 Standardwert 1 mm/ 0,0394 inch) |
| | | | | | 1 = | automatisch (Standardwert 1 mm/ 0,0394 inch) |
| | | | | | 2 = | manuell (wie unter V2 programmiert) |

[1] Felder Technologie können ausgeblendet sein in Abhängigkeit von Settingdatum SD52216 $MCS_FUNCTION_MASK_DRILL

[2] Die Parameter 19, 20, und 21 werden nur bei der Gewindeauswahl in den Gewindetabellen der Eingabemaske verwendet. Ein Zugriff auf die Gewindetabellen über Zyklusdefinition bei Zykluslaufzeit ist nicht möglich.

### 3.25.1.25 CYCLE85 - Reiben

## Syntax

```
CYCLE85(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <FFR>, <RFF>,
<_GMODE>, <_DMODE>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung |
|---|---|---|---|---|
| 1 | RP | <RTP> | REAL | Rückzugsebene (abs) |
| 2 | Z0 | <RFP> | REAL | Bezugspunkt (abs) |
| 3 | SC | <SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 4 | Z1 | <DP> | REAL | Bohrtiefe (abs), siehe <_AMODE> | | |
| 5 | Z1 | <DPR> | REAL | Bohrtiefe (ink), siehe <_AMODE> | | |
| 6 | DT | <DTB> | REAL | Verweilzeit auf Endbohrtiefe, siehe <_AMODE> | | |
| 7 | F | <FFR> | REAL | Vorschub | | |
| 8 | FR | <RFF> | REAL | Vorschub beim Rückzug | | |
| 9 | | <_GMODE> | INT | reserviert | | |
| 10 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| 11 | | <_AMODE> | INT | Alternativmodus (Bohren) | | |
| | | | | EINER: | Bohrtiefe Z1 (abs/ink) | |
| | | | | | 0 = | Kompatibilität, aus Programmierung DP/DPR |
| | | | | | 1 = | inkrementell |
| | | | | | 2 = | absolut |
| | | | | ZEHNER: | Verweilzeit DT auf Endbohrtiefe in Sekunden/Umdrehungen | |
| | | | | | 0 = | Kompatibilität, aus Vorzeichen von DT (> 0 Sekunden oder < 0 Umdrehungen) |
| | | | | | 1 = | in Sekunden |
| | | | | | 2 = | in Umdrehungen |

### 3.25.1.26 CYCLE86 - Ausdrehen

## Syntax

```
CYCLE86(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDIR>, <RPA>,
<RPO>, <RPAP>, <POSS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung |
|---|---|---|---|---|
| 1 | RP | <RTP> | REAL | Rückzugsebene (abs) |
| 2 | Z0 | <RFP> | REAL | Bezugspunkt (abs) |
| 3 | SC | <SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 4 | Z1 | <DP> | REAL | Bohrtiefe (abs), siehe <_AMODE> | | |
| 5 | Z1 | <DPR> | REAL | Bohrtiefe (ink), siehe <_AMODE> | | |
| 6 | DT | <DTB> | REAL | Verweilzeit auf Endbohrtiefe, siehe <_AMODE> | | |
| 7 | DIR | <SDIR> | INT | Spindeldrehrichtung | 3 = | M3 |
| | | | | | 4 = | M4 |
| 8 | DX | <RPA> | REAL | Abhebebetrag in X-Richtung | | |
| 9 | DY | <RPO> | REAL | Abhebebetrag in Y-Richtung | | |
| 10 | DZ | <RPAP> | REAL | Abhebebetrag in Z-Richtung | | |
| 11 | SPOS | <POSS> | REAL | Spindelposition zum Abheben (für orientierten Spindelhalt, in Grad) | | |
| 12 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | Abhebemodus | |
| | | | | | 0 = | abheben, Kompatibilität |
| | | | | | 1 = | nicht abheben |
| 13 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| 14 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Bohrtiefe Z1 (abs/ink) | |
| | | | | | 0 = | Kompatibilität, aus Programmierung <DP>/<DPR> |
| | | | | | 1 = | inkrementell |
| | | | | | 2 = | absolut |
| | | | | ZEHNER: | Verweilzeit auf Endbohrtiefe DT in Sekunden/Umdrehungen | |
| | | | | | 0 = | Kompatibilität, aus Vorzeichen von DT (> 0 Sekunden oder < 0 Umdrehungen) |
| | | | | | 1 = | in Sekunden |
| | | | | | 2 = | in Umdrehungen |

### 3.25.1.27 CYCLE92 - Abstich

## Syntax

```
CYCLE92(<_SPD>, <_SPL>, <_DIAG1>, <_DIAG2>, <_RC>, <_SDIS>, <_SV1>,
<_SV2>, <_SDAC>, <_FF1>, <_FF2>, <_SS2>, <_DIAGM>, <_VARI>, <_DN>,
<_DMODE>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | X0 | <_SPD> | REAL | Bezugspunkt (abs, immer Durchmesser) | | |
| 2 | Y0 | <_SPL> | REAL | Bezugspunkt (abs) | | |
| 3 | X1 | <_DIAG1> | REAL | Tiefe für Drehzahlreduzierung, siehe <_AMODE> (EINER) | | |
| 4 | X2 | <_DIAG2> | REAL | Endtiefe, siehe <_AMODE> (ZEHNER) | | |
| 5 | R/FS | <_RC> | REAL | Verrundusradius oder Fasebreite, siehe <_AMODE> (TAUSENDER) | | |
| 6 | SC | <_SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) | | |
| 7 | S | <_SV1> | REAL | konstante Spindeldrehzahl, siehe <_AMODE> (ZEHNTAUSENDER) | | |
| | V | | | konstante Schnittgeschwindigkeit | | |
| 8 | SV | <_SV2> | REAL | Maximaldrehzahl bei konstanter Schnittgeschwindigkeit | | |
| 9 | DIR | <_SDAC> | INT | Spindeldrehrichtung | 3 = | für M3 |
| | | | | | 4 = | für M4 |
| 10 | F | <_FF1> | REAL | Vorschub bis Tiefe für Drehzahlreduzierung | | |
| 11 | FR | <_FF2> | REAL | reduzierter Vorschub bis Endtiefe | | |
| 12 | SR | <_SS2> | REAL | reduzierte Drehzahl bis Endtiefe | | |
| 13 | XM | <_DIAGM> | REAL | Tiefe Teilefänger ausfahren (abs, immer Durchmesser) | | |
| 14 | | <_VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | Rückzug | |
| | | | | | 0 = | Rückzug auf <_SPD> + <_SDIS> |
| | | | | | 1 = | kein Rückzug am Ende |
| | | | | ZEHNER: | Teilefänger | |
| | | | | | 0 = | Nein, kein M-Befehl ausführen |
| | | | | | 1 = | Ja, Aufruf von CUST_TECHCYC(101)- Schublade ausfahren, CUST_TECHCYC(102)- Schublade schließen |
| 15 | | <_DN> | INT | D-Nummer für 2. Schneide, wenn nicht programmiert ⇒ D+1 | | |
| 20 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 21 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Tiefe für Drehzahlreduzierung (<_DIAG1>) | |
| | | | | | 0 = | absolut, Wert der Planachse im Durchmesser |
| | | | | | 1 = | inkrementell, Wert der Planachse im Radius |
| | | | | ZEHNER: | Endtiefe (<_DIAG2>) | |
| | | | | | 0 = | absolut, Wert der Planachse im Durchmesser |
| | | | | | 1 = | inkrementell, Wert der Planachse im Radius |
| | | | | HUNDERTER: | reserviert | |
| | | | | TAUSENDER: | Radius/Fase (<_RC>) | |
| | | | | | 0 = | Radius |
| | | | | | 1 = | Fase |
| | | | | ZEHNTAUSENDER: | Spindeldrehzahl/Schnittgeschwindigkeit (<_SV1>) | |
| | | | | | 0 = | konstante Spindeldrehzahl |
| | | | | | 1 = | konstante Schnittgeschwindigkeit |

### 3.25.1.28 CYCLE95 - Konturabspanen

## Syntax

```
CYCLE95(<NPP>, <MID>, <FALZ>, <FALX>, <FAL>, <FF1>, <FF2>, <FF3>,
<_VARI>, <DT>, <DAM>, <_VRT>, <_GMODE>, <_DMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung |
|---|---|---|---|---|
| 1 | CON | <NPP> | STRING [140] | Konturname |
| 2 | D | <MID> | REAL | maximale Tiefenzustellung beim Schruppen, siehe <_GMODE> |
| 3 | UZ | <FALZ> | REAL | Schlichtaufmaß in Z |
| 4 | UX | <FALX> | REAL | Schlichtaufmaß in X |
| 5 | U | <FAL> | REAL | Schlichtaufmaß konturparallel (wirkt in beiden Achsen) |
| 6 | F | <FF1> | REAL | Vorschub für Schruppen |
| 7 | FY | <FF2> | REAL | Eintauchvorschub Hinterschnitte |
| 8 | FS | <FF3> | REAL | Schlichtvorschub |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 9 | | <_VARI> | INT | Bearbeitungsart | | |
| | | | | EINER und ZEHNER: | | |
| | | | | | 1 = | Schruppen, längs, Außen |
| | | | | | 2 = | Schruppen, plan, Außen |
| | | | | | 3 = | Schruppen, längs, Innen |
| | | | | | 4 = | Schruppen, plan, Innen |
| | | | | | 5 = | Schlichten, längs, Außen |
| | | | | | 6 = | Schlichten, plan, Außen |
| | | | | | 7 = | Schlichten, längs, Innen |
| | | | | | 8 = | Schlichten, plan, Innen |
| | | | | | 9 = | Komplettbearbeitung, längs, Außen |
| | | | | | 10 = | Komplettbearbeitung, plan, Außen |
| | | | | | 11 = | Komplettbearbeitung, längs, Innen |
| | | | | | 12 = | Komplettbearbeitung, plan, Innen |
| | | | | HUNDERTER: | | |
| | | | | | 0 = | mit Nachziehen an der Kontur, ohne Restecken |
| | | | | | 1 = | ohne Nachziehen an der Kontur |
| | | | | | 2 = | Nachziehen nur bis zum vorherigen Schnittpunkt, es können Restecken entstehen |
| 10 | DT | <DT> | REAL | Verweilzeit bei Vorschubunterbrechung | | |
| 11 | DI | <DAM> | REAL | Abstand Vorschubunterbrechungen | | |
| 12 | VRT | <_VRT> | REAL | Abhebweg von der Kontur | | |
| | | | | | 0 = | intern wird ein Abhebweg von 1 mm unabhängig vom Maßsystem inch/metrisch verwendet |
| | | | | | > 0 = | Abhebweg |
| 13 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | Bewertung der Zustelltiefe | |
| | | | | | 0 = | Zustelltiefe wird entsprechend G-Gruppe DIAMON/DIAMOF verrechnet |
| | | | | | 1 = | Zustelltiefe wirkt als Radiuswert (unabhängig vom DIAMON/ DIAMOF) |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 14 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | TAUSENDER: | | |
| | | | | | 0 = | Kompatibilitätsmodus: Konturname steht in NPP |
| | | | | | 1 = | Konturname wird im CYCLE62 programmiert und in _SC_CONT_NAME übergeben |

### 3.25.1.29 CYCLE98 - Gewindekette

## Syntax

```
CYCLE98(<_PO1>, <_DM1>, <_PO2>, <_DM2>, <_PO3>, <_DM3>, <_PO4>,
<_DM4>, <APP>, <ROP>, <TDEP>, <FAL>, <_IANG>, <NSP>, <NRC>, <NID>,
<_PP1>, <_PP2>, <_PP3>, <_VARI>, <_NUMTH>, <_VRT>, <_MID>, <_GDEP>,
<_IFLANK>, <_PITA>, <_PITM1>, <_PITM2>, <_PITM3>, <_DMODE>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung |
|---|---|---|---|---|
| 1 | Z0 | <_PO1> | REAL | Bezugspunkt in Z (abs) |
| 2 | X0 | <_DM1> | REAL | Bezugspunkt in X (abs), im Durchmesser |
| 3 | Z1 | <_PO2> | REAL | Zwischenpunkt 1 in Z, (abs/ink), siehe <_AMODE> (EINER) |
| 4 | X1 | <_DM2> | REAL | Zwischenpunkt 1 in X (abs/ink), siehe <_AMODE> (ZEHNER) oder |
| | X1α | | | Gewindeschräge 1 (-90° bis 90°)<br>abs immer Durchmesser, ink immer Radius |
| 5 | Z2 | <_PO3> | REAL | Zwischenpunkt 2 in Z, (abs/ink), siehe <_AMODE> (HUNDERTER) |
| 6 | X2 | <_DM3> | REAL | Zwischenpunkt 2 in X (abs/ink), siehe <_AMODE> (TAUSENDER) oder |
| | X2α | | | Gewindeschräge 2 (-90° bis 90°)<br>abs immer Durchmesser, ink immer Radius |
| 7 | Z3 | <_PO4> | REAL | Endpunkt in Z (abs/ink), siehe <_AMODE> (ZEHNTAUSENDER) |
| 8 | X3 | <_DM4> | REAL | Endpunkt in X, (abs/ink), siehe <_AMODE> (HUNDERTTAUSENDER) oder |
| | X3α | | | Gewindeschräge 3 (-90° bis 90°)<br>abs immer Durchmesser, ink immer Radius |
| 9 | LW | <APP> | REAL | Gewindevorlauf (ink, ohne Vorzeichen einzugeben) |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 10 | LR | <ROP> | REAL | Gewindeauslauf (ink, ohne Vorzeichen einzugeben) | | |
| 11 | H1 | <TDEP> | REAL | Gewindetiefe (ink, ohne Vorzeichen einzugeben) | | |
| 12 | U | <FAL> | REAL | Schlichtaufmaß in X und Z | | |
| 13 | DP | <_IANG> | REAL | Zustellschräge als Abstand oder Winkel, siehe <_AMODE> (EINEMILLION) | | |
| | αP | | | Die Zustellschräge wirkt entsprechend der Einstellung Parameter <_VARI> (HUNDERTER). | | |
| | | | | | Definition für <_VARI_HUNDERTER = 0 - Kompatibilitätsmode: | |
| | | | | | > 0 = | Flankenzustellung an einer Flanke |
| | | | | | 0 = | Zustellung senkrecht im Gewinde |
| | | | | | < 0 = | Flankenzustellung mit alternierenden Flanken |
| | | | | | Definition für _VARI_HUNDERTER<>0: | |
| | | | | | > 0 = | Zustellung an der positiven Flanke |
| | | | | | 0 = | mittige Zustellung |
| | | | | | < 0 = | Zustellung an der negativen Flanke |
| 14 | α0 | <NSP> | REAL | Startwinkelversatz für den 1. Gewindegang | | |
| 15 | | <NRC> | INT | Anzahl der Schruppschnitte, siehe <_VARI> (ZEHNTAUSENDER) | | |
| 16 | NN | <NID> | INT | Anzahl Leerschnitte | | |
| 17 | P0 | <_PP1> | REAL | Gewindesteigung 1. Gewindeabschnitt, siehe <_PITA> | | |
| 18 | P1 | <_PP2> | REAL | Gewindesteigung 2. Gewindeabschnitt, siehe <_PITA> | | |
| 19 | P2 | <_PP3> | REAL | Gewindesteigung 3. Gewindeabschnitt, siehe <_PITA> | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 20 | | <_VARI> | INT | Bearbeitung | | |
| | | | | EINER: | Technologie | |
| | | | | | 1 = | Außengewinde mit linearer Zustellung |
| | | | | | 2 = | Innengewinde mit linearer Zustellung |
| | | | | | 3 = | Außengewinde mit degressiver Zustellung, Spanquerschnitt bleibt konstant |
| | | | | | 4 = | Innengewinde mit degressiver Zustellung, Spanquerschnitt bleibt konstant |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Zustellungsart | |
| | | | | | 0 = | Kompatibilitätsmode für <_IANG> |
| | | | | | 1 = | einseitige Zustellung |
| | | | | | 2 = | wechselnde Zustellung |
| | | | | TAUSENDER: | reserviert | |
| | | | | ZEHNTAUSENDER: | Alternative Tiefenzustellung | |
| | | | | | 0 = | Kompatibilität, Vorgabe Anzahl der Schruppschnitte (<_NRC>) |
| | | | | | 1 = | Vorgabe Wert für 1. Zustellung (<_MID>) |
| | | | | HUNDERTTAUSENDER: | Bearbeitungsart | |
| | | | | | 0 = | Kompatibilität (Schruppen und Schlichten) |
| | | | | | 1 = | Schruppen |
| | | | | | 2 = | Schlichten |
| | | | | | 3 = | Schruppen und Schlichten |
| | | | | EINEMILLION: | Bearbeitungsreihenfolge bei mehrgängigen Gewinde | |
| | | | | | 0 = | Reihenfolge der Gänge aufsteigend |
| | | | | | 1 = | Reihenfolge der Gänge gegenüber |
| 21 | N | <_NUMTH> | INT | Anzahl Gewindegänge | | |
| 22 | | <_VRT> | REAL | Rücklaufabstand (ink) | | |
| | | | | | 0 = | intern wird ein Abhebweg von 1 mm unabhängig vom Maßsystem inch/metrisch verwendet |
| | | | | | > 0 = | Abhebweg |
| 23 | D1 | <_MID> | REAL | erste Zustellung, siehe <_VARI> (ZEHNTAUSENDER) | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 24 | DA | <_GDEP> | REAL | Gangwechseltiefe (wirkt nur bei "mehrgängig") | | |
| | | | | | 0 = | keine Gangwechseltiefe berücksichtigen |
| | | | | | > 0 = | Gangwechseltiefe berücksichtigen |
| 25 | | <_IFLANK> | REAL | Zustellschräge als Breite (nur für die Oberfläche) | | |
| 26 | | <_PITA> | INT | Bewertung der Gewindesteigung | | |
| | | | | | 0 = | Kompatibilitätsmode für Gewindesteigung, Auswertung <_PP1> bis <_PP3> wie bisher entsprechend aktivem System metrisch/inch |
| | | | | | 1 = | Steigung in mm |
| | | | | | 2 = | Steigung in TPI (Gewindegänge pro inch) |
| | | | | | 3 = | Steigung in inch |
| | | | | | 4 = | MODUL |
| 27 | | <_PITM1> | STRING[15] | String als Merker für Gewindesteigungseingabe (nur für die Oberfläche) | | |
| 28 | | <_PITM2> | STRING[15] | String als Merker für Gewindesteigungseingabe (nur für die Oberfläche) | | |
| 29 | | <_PITM3> | STRING[15] | String als Merker für Gewindesteigungseingabe (nur für die Oberfläche) | | |
| 30 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | --- | reserviert |
| | | | | HUNDERTER: | --- | reserviert |
| | | | | TAUSENDER: | --- | reserviert |
| | | | | ZEHNTAUSENDER: | Technologieskalierung innerhalb von Zyklenmasken (Seite 1172) | |
| | | | | | 0 = | Eingabe: komplett |
| | | | | | 1 = | Eingabe: einfach |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 31 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | 1. Zwischenpunkt in Z (Z1) | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |
| | | | | ZEHNER: | 1. Zwischenpunkt in X (X1) | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |
| | | | | | 2 = | α |
| | | | | HUNDERTER: | 2. Zwischenpunkt in Z (Z2) | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |
| | | | | TAUSENDER: | 2. Zwischenpunkt in X (X2) | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |
| | | | | | 2 = | α |
| | | | | ZEHNTAUSENDER: | Endpunkt in Z (Z3) | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |
| | | | | HUNDERTTAUSENDER: | Endpunkt in X (X3) | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |
| | | | | | 2 = | α |
| | | | | EINEMILLION: | Auswahl Zustellschräge als Winkel oder Breite | |
| | | | | | 0 = | Zustellwinkel <_IANG> |
| | | | | | 1 = | Zustellschräge <_IFLANK> |
| | | | | ZEHNMILLIONEN: | eingängig/mehrgängig | |
| | | | | | 0 = | Kompatibilitätsmodus (Startwinkel <_NSP> wird ausgewertet) |
| | | | | | 1 = | eingängig (mit Startwinkelversatz <_NSP>) |
| | | | | | 2 = | mehrgängig |

### 3.25.1.30 CYCLE99 - Gewindedrehen

## Syntax

```
CYCLE99(<_SPL>, <_SPD>, <_FPL>, <_FPD>, <_APP>, <_ROP>, <_TDEP>,
<_FAL>, <_IANG>, <_NSP>, <_NRC>, <_NID>, <_PIT>, <_VARI>, <_NUMTH>,
<_SDIS>, <_MID>, <_GDEP>, <_PIT1>, <_FDEP>, <_GST>, <_GUD>,
<_IFLANK>, <_PITA>, <_PITM>, <_PTAB>, <_PTABA>, <_DMODE>, <_AMODE>,
<_S_XRS>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | Z0 | <_SPL> | REAL | Bezugspunkt (abs) | | |
| 2 | X0 | <_SPD> | REAL | Bezugspunkt (abs, immer Durchmesser) | | |
| 3 | Z1 | <_FPL> | REAL | Endpunkt in Verbindung mit <_AMODE> (EINER) | | |
| 4 | X1 | <_FPD> | REAL | Endpunkt, in Verbindung mit <_AMODE> (ZEHNER) | | |
| 5 | LW/LW2 | <_APP> | REAL | Gewindevorlauf in Verbindung mit <_AMODE> (HUNDERTER) oder<br>Gewindeeinlauf = Gewindeauslauf in Verbindung mit <_AMODE> (HUNDERTER) | | |
| 6 | LR | <_ROP> | REAL | Gewindeauslauf | | |
| 7 | H1 | <_TDEP> | REAL | Gewindetiefe | | |
| 8 | U | <_FAL> | REAL | Schlichtaufmaß in X und Z | | |
| 9 | DP | <_IANG> | REAL | Zustellschräge als Abstand oder Winkel, in Verbindung mit <_AMODE> (TAUSENDER) | | |
| | αP | | | | > 0 = | Zustellung an der positiven Flanke |
| | | | | | < 0 = | Zustellung an der negativen Flanke |
| | | | | | 0 = | mittige Zustellung |
| 10 | α0 | <_NSP> | REAL | Startwinkelversatz (wirkt nur bei "eingängig") | | |
| 11 | ND | <_NRC> | INT | Anzahl Schruppschnitte, in Verbindung mit <_VARI> (ZEHNTAUSENDER) | | |
| 12 | NN | <_NID> | INT | Anzahl Leerschnitte | | |
| 13 | P | <_PIT> | REAL | Gewindesteigung als Wert, in Verbindung mit <_PITA> | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 14 | | <_VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | Technologie | |
| | | | | | 1 = | Außengewinde mit linearer Zustellung |
| | | | | | 2 = | Innengewinde mit linearer Zustellung |
| | | | | | 3 = | Außengewinde mit degressiver Zustellung, Spanquerschnitt bleibt konstant |
| | | | | | 4 = | Innengewinde mit degressiver Zustellung, Spanquerschnitt bleibt konstant |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Zustellungsart | |
| | | | | | 1 = | einseitige Zustellung |
| | | | | | 2 = | wechselnde Zustellung |
| | | | | TAUSENDER: | reserviert | |
| | | | | ZEHNTAUSENDER: | Alternative Tiefenzustellung | |
| | | | | | 0 = | Vorgabe Anzahl der Schruppschnitte (<_NRC>) |
| | | | | | 1 = | Vorgabe Wert für 1. Zustellung (<_MID>) |
| | | | | HUNDERTTAUSENDER: | Bearbeitungsart | |
| | | | | | 1 = | Schruppen |
| | | | | | 2 = | Schlichten |
| | | | | | 3 = | Schruppen und Schlichten |
| | | | | EINEMILLION: | Bearbeitungsreihenfolge bei mehrgängigen Gewinde | |
| | | | | | 0 = | Reihenfolge der Gänge aufsteigend |
| | | | | | 1 = | Reihenfolge der Gänge gegenüber |
| 15 | N | <_NUMTH> | INT | Anzahl Gewindegänge | | |
| 16 | VR | <_SDIS> | REAL | Rücklaufabstand, ink | | |
| 17 | D1 | <_MID> | REAL | Erste Zustelltiefe, in Verbindung mit <_VARI> (ZEHNTAUSENDER) | | |
| 18 | DA | <_GDEP> | REAL | Gangwechseltiefe (wirkt nur bei "mehrgängig" | | |
| | | | | | 0 = | keine Gangwechseltiefe berücksichtigen |
| | | | | | > 0 = | Gangwechseltiefe berücksichtigen |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 19 | G | <_PIT1> | REAL | Änderung der Steigung pro Umdrehung | | |
| | | | | | 0 = | Gewindesteigung ist konstant (G33) |
| | | | | | > 0 = | Gewindesteigung wird größer (G34) |
| | | | | | < 0 = | Gewindesteigung wird kleiner (G35) |
| 20 | | <_FDEP> | REAL | Eintauchtiefe (ohne Vorzeichen einzugeben) | | |
| 21 | N1 | <_GST> | INT | Startgang N1 = 1...N, in Verbindung mit <_AMODE> (HUNDERTTAUSENDER) | | |
| 22 | | <_GUD> | INT | reserviert | | |
| 23 | | <_IFLANK> | REAL | Zustellschräge als Breite (nur für die Oberfläche) | | |
| 24 | | <_PITA> | INT | Maßeinheit der Gewindesteigung (Auswertung PIT und/oder MPIT) | | |
| | | | | | 0 = | Steigung in mm - Auswertung MPIT/PIT |
| | | | | | 1 = | Steigung in mm - Auswertung PIT |
| | | | | | 2 = | Steigung in TPI - Auswertung PIT (Gewindegänge pro inch) |
| | | | | | 3 = | Steigung in inch - Auswertung PIT |
| | | | | | 4 = | MODUL - Auswertung PIT |
| 25 | | <_PITM> | STRING[15] | String als Merker für Gewindesteigungseingabe (nur für die Oberfläche)[1)] | | |
| 26 | | <_PTAB> | STRING[20] | String für Gewindetabelle (nur für die Oberfläche)[1)] | | |
| 27 | | <_PTABA> | STRING[20] | String für Auswahl in der Gewindetabelle (nur für die Oberfläche)[1)] | | |
| 28 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | Gewindeart | |
| | | | | | 0 = | Längsgewinde |
| | | | | | 1 = | Plangewinde |
| | | | | | 2 = | Kegelgewinde |
| | | | | HUNDERTER: | --- | reserviert |
| | | | | TAUSENDER: | --- | reserviert |
| | | | | ZEHNTAUSENDER: | Technologieskalierung innerhalb von Zyklenmasken (Seite 1172) | |
| | | | | | 0 = | Eingabe: komplett |
| | | | | | 1 = | Eingabe: einfach |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 29 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Gewindelänge in Z | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |
| | | | | ZEHNER: | Gewindelänge in X | |
| | | | | | 0 = | absolut, Wert der Planachse im Durchmesser |
| | | | | | 1 = | inkrementell, Wert der Planachse im Radius |
| | | | | | 2 = | α |
| | | | | HUNDERTER: | Vorlauf-/Einlaufwegbewertung <_APP> | |
| | | | | | 0 = | Gewindevorlauf <_APP> |
| | | | | | 1 = | Gewindeeinlauf = Gewindeauslauf <_APP> = -<_ROP> |
| | | | | | 2 = | Gewindeeinlaufweg vorgeben <_APP> = -<_APP> |
| | | | | TAUSENDER: | Auswahl Zustellschräge als Winkel oder Breite | |
| | | | | | 0 = | Zustellwinkel <_IANG> |
| | | | | | 1 = | Zustellschräge <_IFLANK> |
| | | | | ZEHNTAUSENDER: | eingängig/mehrgängig | |
| | | | | | 0 = | eingängig (mit Startwinkelversatz <_NSP>) |
| | | | | | 1 = | mehrgängig |
| | | | | HUNDERTTAUSENDER: | Startgang <_GST> | |
| | | | | | 0 = | komplette Bearbeitung |
| | | | | | 1 = | ab diesem Gang die Bearbeitung starten |
| | | | | | 2 = | nur diesen Gang bearbeiten |
| | | | | EINEMILLION: | Durchhangskompensation bei Längsgewinde | |
| | | | | | 0 = | Segmenthöhe balliges Gewinde XS |
| | | | | | 1 = | Radius balliges Gewinde RS |
| 30 | XS / RS | <_S_XRS> | REAL | Duchhangkompensation bei Längsgewinde in Verbindung mit <_AMODE>: EINEMILLION | | |

**Hinweis**

[1] Die Parameter <_PITM>, <_PTAB> und <_PTABA> werden nur bei der Gewindeauswahl in den Gewindetabellen der Eingabemaske verwendet.
Ein Zugriff auf die Gewindetabellen über Zyklusdefinition bei Zykluslaufzeit ist nicht möglich.

### 3.25.1.31 CYCLE435 - Abrichterkoordinatensystem setzen

## Syntax

```
CYCLE435(<_T>, <_DD>, <S_TA>, <S_DA>, <S_AD>, <S_AL>, <S_PVD>,
<S_PVL>, <S_PD>, <S_PL>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | | <_T> | STRING[32] | Werkzeugname der Schleifscheibe | | |
| 2 | | <_DD> | INT | Schneidennummer der Schleifscheibe | | |
| 3 | | <S_TA> | STRING[32] | Bezugspunkt Abrichter - Name des Abrichters | | |
| 4 | | <S_DA> | INT | Schneidennummer des Abrichters | | |
| 5 | | <S_AD> | REAL | Abrichtbetrag Durchmesser | | |
| 6 | | <S_AL> | REAL | Abrichtbetrag plan | | |
| 7 | | <S_PVD> | REAL | Profilierverschiebung Durchmesser | | |
| 8 | | <S_PVL> | REAL | Profilierverschiebung plan | | |
| 9 | | <S_PD> | REAL | Profilieraufmaß Durchmesser | | |
| 10 | | <S_PL> | REAL | Profilieraufmaß plan | | |
| 11 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | aktives Werkzeug bei Zyklenende | |
| | | | | | 0 = | Abrichter aktiv |
| | | | | | 1 = | Scheibe aktiv |

### 3.25.1.32 CYCLE495 - Profilieren

## Syntax

```
CYCLE495(<_T>, <_DD>, <_SC>, <_F>, <_VARI>, <_D>, <_DX>, <_DZ>,
<S_PA>, <S_N>, <_DMODE>, <_AMODE>, <S_FW>, <S_HW>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung |
|---|---|---|---|---|
| 1 | | <_T> | STRING[20] | Werkzeugname der Schleifscheibe |
| 2 | | <_DD> | INT | Schneidennummer der Schleifscheibe |
| 3 | | <_SC> | REAL | Abhebebetrag zum Umfahren von Hindernissen, inkrementell |
| 4 | | <_F> | REAL | Vorschub Profilieren |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 5 | | <_VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | Profilierungsart | |
| | | | | | 1 = | achsparallel |
| | | | | | 2 = | konturparallel |
| | | | | ZEHNER: | Bearbeitungsrichtung | |
| | | | | | 0 = | ziehend<br>möglich mit den Schneidenlagen 1 bis 4 |
| | | | | | 1 = | stoßend<br>möglich mit den Schneidenlagen 1 bis 4 |
| | | | | | 2 = | wechselnd<br>möglich mit den Schneidenlagen 1 bis 8 |
| | | | | | 3 = | Anfang → Ende<br>möglich mit den Schneidenlagen 1 bis 8 |
| | | | | | 4 = | Ende → Anfang<br>möglich mit den Schneidenlagen 1 bis 8 |
| | | | | HUNDERTER: | Zustellrichtung | |
| | | | | | 1 = | Zustellung X- bei G18 bzw. Y- bei G19 |
| | | | | | 2 = | Zustellung X+ bei G18 bzw. Y+ bei G19 |
| | | | | | 3 = | Zustellung Z- bei G18 und bei G19 |
| | | | | | 4 = | Zustellung Z+ bei G18 und bei G19 |
| 6 | | <_D> | REAL | Abrichtbetrag bei der Profilierart achsparallel | | |
| 7 | | <_DX> | REAL | Abrichtbetrag X bei G18 bzw. Y bei G19 bei der Profilierart konturparallel | | |
| 8 | | <_DZ> | REAL | Abrichtbetrag Z bei G18 und G19 bei der Profilierart konturparallel | | |
| 9 | | <S_PA> | REAL | Profilieraufmaß | | |
| 10 | | <S_N> | INT | Anzahl der Hübe im Profilierprogramm | | |
| 11 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 12 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Auswahl Profilieren Neu/Fortsetzen | |
| | | | | | 1 = | Neu |
| | | | | | 2 = | Fortsetzen |
| | | | | ZEHNER: | Auswahl Profilieraufmaß | |
| | | | | | 0 = | von der Rohkontur bis zum tiefsten Punkt der Kontur |
| | | | | | 1 = | von der Rohkontur bis zum höchsten Punkt der Kontur |
| 13 | | <S_FW> | REAL | Freiwinkel des Abrichters | | |
| 14 | | <S_HW> | REAL | Halterwinkel des Abrichters | | |

### 3.25.1.33 CYCLE782 - An Beladung anpassen

**Hinweis**

Um CYCLE782 nutzen zu können, ist eine Lizenz für die Option "Intelligente Lastanpassung" erforderlich.

## Syntax

```
CYCLE782(<S_MODE>, <S_TESTAXIS>, <S_VALUE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | | <S_MODE> | INT | Bearbeitungsmodus | | |
| | | | | EINER: | An-/Abwahl | |
| | | | | | 0 = | Adaption deaktivieren |
| | | | | | 1 = | Adaption aktivieren |
| | | | | | 2 = | Grundträgheit der Achse ermittlen und aktivieren (Inbetriebnahmefunktion) |
| | | | | ZEHNER: | Messvariante (nur bei <S_MODE> HUNDERTER-STELLE = 0) | |
| | | | | | 0 = | Normalmodus |
| | | | | | 1 = | Genau-Modus (z. B. mit Reibungsermittlung, etc.) |
| | | | | HUNDERTER | Messung oder Vorgabe eines Trägheitswerts | |
| | | | | | 0 = | Messung<br>In diesem Modus werden Verfahrbewegungen zur Ermittlung der Beladung durchgeführt. |
| | | | | | 1 = | Vorgabe eines Trägheitswerts<br>In diesem Modus werden keine Verfahrbewegungen zur Ermittlung der Beladung durchgeführt. Stattdessen wird ein vorher z. B. durch die Automatische Servo Optimierung (AST) ermittelter Trägheitswert übergeben. |
| | | | | TAUSENDER | Messergebnisanzeige | |
| | | | | | 0 = | Messergebnisbild AUS |
| | | | | | 1 = | Messergebnisbild EIN |
| | | | | ZEHNTAUSENDER | Dauer der Messergebnisanzeige | |
| | | | | | 1 = | Abwahl der Anzeige automatisch nach 8 s |
| | | | | | 3 = | Quittieren mit NC-Start |
| | | | | HUNDERTTAUSENDER | Auswahl Beladung [1] | |
| | | | | | 0 = | Beladung insgesamt ermitteln: Achse leer + Werkstück |
| | | | | | 1 = | Beladung separat ermitteln: Werkstück |
| 2 | | <S_TESTAXIS> | AXIS | Kanalachse:<br>• für die Adaptionen aktiviert werden sollen.<br>• für die die Trägheit ermittelt werden soll (nur bei <S_MODE> HUNDERTERSTELLE = 0) | | |
| 3 | | <S_VALUE> | REAL | Trägheitswert (nur bei <S_MODE> HUNDERTERSTELLE = 1) | | |

[1] Wert wird aus Maschinendatum 52212 $MCS_FUNCTION_MASK_TECH Bit 18 automatisch entnommen.

**Hinweis**

Die Inbetriebnahmefunktion "Grundträgheit der Achse ermitteln und aktivieren" (<S_MODE> EINERSTELLE = 2) wird nicht durch die Bedienoberfläche unterstützt. Hierfür muss der Zyklenaufruf selber programmiert werden. Das Ergebnis der Messung wird in das Maschinendatum 53350 $MAS_ILC_BASE_VALUE geschrieben.

Der Wert des Parameters "Auswahl Beladung" (<S_MODE> HUNDERTTAUSENDERSTELLE) wird von der Bedienoberfläche aus dem Maschinendatum 52212 $MCS_FUNCTION_MASK_TECH Bit 18 gelesen und in den Zyklenaufruf geschrieben.

Die Beladung kann sowohl für Linear- als auch für Rundachsen ermittelt oder vorgegeben werden. Für Linearachsen wird immer eine Masse und für Rundachse immer eine Trägheit im Messergebnisbild angezeigt bzw. in der Bedienoberfläche vorgegeben.

Unabhängig vom Achstyp wird in das Maschinendatum 53350 $MAS_ILC_BASE_VALUE und in die Adaption immer eine Trägheit geschrieben.

## Beispiele

### Beispiel 1:

Am Anfang eines NC-Programms sollen die Beladung der Achse MX1 ermittelt und damit Adaptionen aktualisiert werden. Das Ergebnis soll angezeigt und die Anzeige mit NC-Start quittiert werden.

**Programmcode**

```
...
CYCLE782(31011,MX1)
...
```

### Beispiel 2:

Der z. B. über AST ermittelte Trägheitswert soll im Verlauf des Programms in die Variable _MY_VALUE übernommen, der Beladungszyklus mit diesem Wert aufgerufen und damit Adaptionen für die Achse MX2 eingeschaltet werden. Es soll kein Ergebnis angezeigt werden.

**Programmcode**

```
DEF REAL _MY_VALUE
_MY_VALUE=...
...
CYCLE782(101,MX2,_MY_VALUE)
...
```

### Beispiel 3:

Adaptionen für die Achse MX2 sollen ausgeschaltet werden.

**Programmcode**

```
...
CYCLE782(0,MX2)
```

```
Programmcode
...
```

### 3.25.1.34 CYCLE800 - Schwenken Ebene / Schwenken Werkzeug / Ausrichten Werkzeug

## Syntax

```
CYCLE800(<_FR>, <_TC>, <_ST>, <_MODE>, <_X0>, <_Y0>, <_Z0>, <_A>,
<_B>, <_C>, <_X1>, <_Y1>, <_Z1>, <_DIR>, <_FR_I>, <_DMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | | <_FR> | INT | Freifahrmodus: | 0 = | kein Freifahren |
| | | | | | 1 = | Freifahren Maschinenachse Z |
| | | | | | 2 = | Freifahren Maschinenachse Z und danach XY |
| | | | | | 3 = | reserviert |
| | | | | | 4 = | Freifahren in Werkzeugrichtung maximal |
| | | | | | 5 = | Freifahren in Werkzeugrichtung inkrementell |
| 2 | | <_TC> | STRING[32] | Name Schwenkdatensatz: | "" | (kein Name) wenn nur 1 Schwenkdatensatz vorhanden |
| | | | | | "0" | Abwahl Schwenkdatensatz (Löschen der Schwenkframes) |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 3 | | <_ST> | INT | Status Transformationen | | |
| | | | | EINER: | | |
| | | | | | 0 = | Neu , Schwenkebene wird gelöscht und mit den aktuellen Parametern neu berechnet |
| | | | | | 1 = | Additiv , Schwenkebene setzt additiv auf aktive Schwenkebene auf |
| | | | | ZEHNER: | Werkzeugspitze nachführen ja/nein (nur aktiv, wenn in IBN SCHWENKEN Funktion eingerichtet ist) | |
| | | | | | 0 = | Werkzeugspitze nicht nachführen |
| | | | | | 1 = | Werkzeugspitze nachführen (TRAORI) |
| | | | | HUNDERTER: | Werkzeug anstellen / ausrichten (Funktion wird in Eingabemaske SCHWENKEN Werkzeug angezeigt) | |
| | | | | | 0 = | Werkzeug nicht anstellen |
| | | | | | 1 = | Werkzeug (vorzugweise Radienfräser) anstellen |
| | | | | | 2 = | **Dreh**werkzeug ausrichten (wenn B-Achskinematik für Drehtechnologie in IBN Schwenken eingerichtet ist) |
| | | | | | 3 = | **Fräs**werkzeug ausrichten (wenn B-Achskinematik für Drehtechnologie in IBN Schwenken eingerichtet ist) |
| | | | | TAUSENDER: | interner Parameter Schwenken in JOG | |
| | | | | ZEHNTAUSENDER: | siehe Parameter Richtung <_DIR> | |
| | | | | | 0 = | Schwenken "ja" |
| | | | | | 1 = | Schwenken "nein" Richtung "Minus" [3] |
| | | | | | 2 = | Schwenken "nein" Richtung "Plus" [3] |
| | | | | HUNDERTTAUSENDER: | siehe Parameter Richtung <_DIR> | |
| | | | | | 0 = | Kompatibilität |
| | | | | | 1 = | Richtungsauswahl "Minus" optimiert (nur für Bedienoberfläche) [4] |
| | | | | | 2 = | Richtungsauswahl "Plus" optimiert (nur für Bedienoberfläche) [4] |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 4 | | <_MODE> [5] | INT | Schwenkmodus: Auswertung der Schwenkwinkel und der Schwenkreihenfolge (bitcodiert!) | | |
| | | | | Bit: 7 6 | 0 0: | Schwenkwinkel achsweise -> siehe Parameter <_A>, <_B>, <_C> |
| | | | | | 0 1: | Raumwinkel -> siehe Parameter <_A>, <_B> [1] |
| | | | | | 1 0: | Projektionswinkel -> siehe Parameter <_A>, <_B>, <_C> [1] |
| | | | | | 1 1: | Schwenkmodus Rundachsen direkt -> siehe Parameter <_A>, <_B> [1] |
| | | | | Bit: 5 4 3 2 1 0<br>(bei Raumwinkeln ohne Bedeutung!) | x x x x 0 1 | 1.Drehung _A um X |
| | | | | | x x x x 1 0 | 1.Drehung _A um Y |
| | | | | | x x x x 1 1 | 1.Drehung _A um Z |
| | | | | | x x 0 1 x x | 2.Drehung _B um X |
| | | | | | x x 1 0 x x | 2.Drehung _B um Y |
| | | | | | x x 1 1 x x | 2.Drehung _B um Z |
| | | | | | 0 1 x x x x | 3.Drehung _C um X |
| | | | | | 1 0 x x x x | 3.Drehung _C um Y |
| | | | | | 1 1 x x x x | 3.Drehung _C um Z |
| 5 | X0 | <_X0> | REAL | Bezugspunkt X vor der Drehung | | |
| 6 | Y0 | <_Y0> | REAL | Bezugspunkt Y vor der Drehung | | |
| 7 | Z0 | <_Z0> | REAL | Bezugspunkt Z vor der Drehung | | |
| 8 | X(A) | <_A> | REAL | 1. Drehung laut Einstellung in Parameter <_MODE> | | |
| 9 | Y(B) | <_B> | REAL | 2. Drehung laut Einstellung in Parameter <_MODE> | | |
| 10 | Z(C) | <_C> | REAL | 3. Drehung laut Einstellung in Parameter <_MODE> | | |
| 11 | X1 | <_X1> | REAL | Bezugspunkt X nach der Drehung | | |
| 12 | Y1 | <_Y1> | REAL | Bezugspunkt Y nach der Drehung | | |
| 13 | Z1 | <_Z1> | REAL | Bezugspunkt Z nach der Drehung | | |
| 14 | - oder + | <_DIR> | INT | Verfahrbewegung der Rundachsen auslösen (default = -1!) | | |
| | | | | | -1 = | auf kleineren Wert der Rundachse 1 oder 2 positionieren[2] |
| | | | | | +1 = | auf größeren Wert der Rundachse 1 oder 2 positionieren[2] |
| | | | | | 0 = | Schwenken nein (nur Schwenkframe berechnen) [1] [3] |
| 15 | FR | <_FR_I> | REAL | Wert (ink) Freifahren in Werkzeugrichtung inkrementell | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 16 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | Darstellung des Beta-Wertes bei Ausrichten Werkzeug | |
| | | | | | 0 = | Wert |
| | | | | | 1 = | Pfeil |

**Hinweis**

Sind folgende Übergabeparameter indirekt (als Parameter) programmiert, wird die Eingabemaske nicht rückübersetzt: <_FR>, <_ST>, <_TC>, <_MODE>, <_DIR>

[1] Auswahl möglich, wenn in IBN SCHWENKEN Funktion eingerichtet ist.

[2] Auswahl möglich, wenn in IBN SCHWENKEN Richtungsbezug auf Rundachse 1 oder 2 eingestellt ist.

kein Auswahlfeld bei Richtungsbezug nein

[3] Auswahl Schwenken "nein" kann ausgeblendet sein SD 55221 Bit 0

Schwenken "nein" Richtung "Minus" entspricht <_DIR> = 0 und _ST ZEHNTAUSENDER = 1

Schwenken "nein" Richtung "Plus" entspricht <_DIR> = 0 und _ST ZEHNTAUSENDER = 2

[4] Die Richtungsauswahl der Rundachse 1 oder 2 erfolgt auch, wenn sich die Rundachse mit dem Richtungsbezug in Polstellung (Positionswert gleich Null) befindet.

[5] Beispiel Codierung: Drehung achsweise, Drehreihenfolge ZYX

Binär: 00011011; Dezimal: 27

Die Achsbezeichner XYZ entsprechen den Geometrieachsen des NC-Kanals. Drehungen um die Achsen XYZ dürfen einzeln ausgeführt werden. Z.B. Drehreihenfolge um ZXZ ist in einem Aufruf des CYCLE800 nicht erlaubt

### 3.25.1.35 CYCLE801 - Positionsmuster Gitter oder Rahmen

## Syntax

```
CYCLE801(<_SPCA>, <_SPCO>, <_STA>, <_DIS1>, <_DIS2>, <_NUM1>,
<_NUM2>, <_VARI>, <_UMODE>, <_ANG1>, <_ANG2>, <_HIDE>, <_NSP>,
<_DMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | X0 | <_SPCA> | REAL | Bezugspunkt für Positionsmuster (Gitter/Rahmen) in der 1. Achse (abs) | | |
| 2 | Y0 | <_SPCO> | REAL | Bezugspunkt für Positionsmuster (Gitter/Rahmen) in der 2. Achse (abs) | | |
| 3 | α0 | <_STA> | REAL | Grund-Drehwinkel (Winkel zur 1. Achse) | < 0 = | Drehung im Uhrzeigersinn |
| | | | | | > 0 = | Drehung im Gegenuhrzeigersinn |
| 4 | L1 | <_DIS1> | REAL | Abstand der Spalten (Positionsabstand 1. Achse, ohne Vorzeichen eingeben) | | |
| 5 | L2 | <_DIS2> | REAL | Abstand der Zeilen(Positionsabstand 2. Achse, ohne Vorzeichen eingeben) | | |
| 6 | N1 | <_NUM1> | INT | Anzahl der Spalte | | |
| 7 | N2 | <_NUM2> | INT | Anzahl der Zeilen | | |
| 8 | | <_VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | Positionsmuster | |
| | | | | | 0 = | Gitter |
| | | | | | 1 = | Rahmen |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | reserviert | |
| 9 | | <_UMODE> | INT | reserviert | | |
| 10 | αX | <_ANG1> | REAL | Scherwinkel zur 1. Achse (Schrägstellung der Zeilen bezogen auf die 1. Achse) | | |
| | | | | | < 0 = | Messung im Uhrzeigersinn (0 bis -90 Grad) |
| | | | | | > 0 = | Messung im Gegenuhrzeigersinn (0 bis 90 Grad) |
| 11 | αY | <_ANG2> | REAL | Scherwinkel zur 2. Achse (Schrägstellung der Spalten bezogen auf die 2. Achse) | | |
| | | | | | < 0 = | Messung im Uhrzeigersinn (0 bis -90 Grad) |
| | | | | | > 0 = | Messung im Gegenuhrzeigersinn (0 bis 90 Grad) |
| 12 | | <_HIDE> | STRING [200] | Ausgeblendete Positionen<br>• max. 198 Zeichen<br>• Angabe der fortlaufenden Positionsnummer, z. B. "1,3" (Positionen 1 und 3 werden nicht ausgeführt) | | |
| 13 | | <_NSP> | INT | reserviert | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 14 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |

### 3.25.1.36 CYCLE802 - Beliebige Positionen

## Syntax

```
CYCLE802(<_XA>, <_YA>, <_X0>, <_Y0>, <_X1>, <_Y1>, <_X2>, <_Y2>,
<_X3>, <_Y3>, <_X4>, <_Y4>, <_X5>, <_Y5>, <_X6>, <_Y6>, <_X7>, <_Y7>,
<_X8>, <_Y8>, <_VARI>, <_UMODE>, <_DMODE>, <S_ABA>, <S_AB0>,
<S_AB1>, <S_AB2>, <S_AB3>, <S_AB4>, <S_AB5>, <S_AB6>, <S_AB7>,
<S_AB8>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | | <_XA> | INT | Alternative für alle X-Positionen (9-stelliger Dezimalwert)<br>Stellenzahl: 876543210 (Die Stelle entspricht der Bohrposition Xn) | | |
| | | | | Stellenwert: | 1 = | absolut (1. programmierte Position immer absolut) |
| | | | | | 2 = | inkrementell |
| 2 | | <_YA> | INT | Alternative für alle Y-Positionen (9-stelliger Dezimalwert)<br>Stellenzahl: 876543210 (Die Stelle entspricht der Bohrposition Yn) | | |
| | | | | Stellenwert: | 1 = | absolut (1. programmierte Position immer absolut) |
| | | | | | 2 = | inkrementell |
| 3 | X0 | <_X0> | REAL | 1. Position X | | |
| 4 | Y0 | <_Y0> | REAL | 1. Position Y | | |
| 5 | X1 | <_X1> | REAL | 2. Position X | | |
| 6 | Y1 | <_Y1> | REAL | 2. Position Y | | |
| 7 | X2 | <_X2> | REAL | 3. Position X | | |
| 8 | Y2 | <_Y2> | REAL | 3. Position Y | | |
| 9 | X3 | <_X3> | REAL | 4. Position X | | |
| 10 | Y3 | <_Y3> | REAL | 4. Position Y | | |
| 11 | X4 | <_X4> | REAL | 5. Position X | | |
| 12 | Y4 | <_Y4> | REAL | 5. Position Y | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung |
|---|---|---|---|---|
| 13 | X5 | <_X5> | REAL | 6. Position X |
| 14 | Y5 | <_Y5> | REAL | 6. Position Y |
| 15 | X6 | <_X6> | REAL | 7. Position X |
| 16 | Y6 | <_Y6> | REAL | 7. Position Y |
| 17 | X7 | <_X7> | REAL | 8. Position X |
| 18 | Y7 | <_Y7> | REAL | 8. Position Y |
| 19 | X8 | <_X8> | REAL | 9. Position X |
| 20 | Y8 | <_Y8> | REAL | 9. Position Y |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 21 | | <_VARI> | INT | Bearbeitung | | |
| | | | | HUNDERTER: | (nur für Aufruf aus Jobshop) (vorerst nur 0 und 2 ausgewertet) | |
| | | | | | 0 = | kein Spindel klemmen |
| | | | | | 1 = | Spindel klemmen nur bei senkrechtem Eintauchen mit G00 oder G01 |
| | | | | | 2 = | Spindel klemmen während der gesamten Bearbeitung |
| | | | | TAUSENDER: | reserviert | |
| | | | | ZEHNTAUSENDER: | Positionsmuster mit/ohne Rundachse – Achskombination (mit <_VARI> HUNDERTTAUSENDER) | |
| | | | | | 0 = | XY (nur XY ohne Rundachse, Kompatibilität) |
| | | | | | 1 = | X,Y oder Z und Rundachse: XA, YB, ZC (1 Rundachse mit Geo-Achse, um die die Rundachse dreht) |
| | | | | | 2 = | XY und Rundachse: XYA, XYB, XYC (1 Rundachse mit 1. und 2. Geo-Achse, ohne TRACYL) |
| | | | | HUNDERTTAUSENDER: | Rundachse | |
| | | | | | 0 = | ohne Rundachse (nur XY, Kompatibilität) |
| | | | | | 1 = | A-Achse (Rundachse um X) |
| | | | | | 2 = | B-Achse (Rundachse um Y) |
| | | | | | 3 = | C-Achse (Rundachse um Z) |
| | | | | ZEHNMILLIONEN + EINEMILLION: | Positionsmuster mit Rundachse – Offset (bei mehreren Rundachsen um die gleiche Achse; wenn Index zu groß, dann 1. Achse) | |
| | | | | | 00 = | 1. A, B oder C-Achse oder bei Kompatibilität |
| | | | | | 01 = | 2. A, B oder C-Achse |
| | | | | | ... | |
| | | | | | 19 = | 20. A, B oder C-Achse |
| 22 | | <_UMODE> | INT | Auswahl der zu klemmenden Spindel: (nur für Aufruf aus Jobshop) (Aufruf Anwenderzyklus CUST_TECHCYC) | | |
| | | | | | 3 = | Klemmen/Lösen Hauptspindel |
| | | | | | 23 = | Klemmen/Lösen Gegenspindel |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 23 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| 24 | | <S_ABA> | INT | Alternative für alle AB-Positionen (9-stelliger Dezimalwert)<br>Stellenzahl: 876543210 (Die Stelle entspricht der Position ABn) | | |
| | | | | Stellenwert: | 1 = | absolut (1. programmierte Position immer absolut) |
| | | | | | 2 = | inkrementell |
| 25 | A0 | <S_AB0> | REAL | 1. Rundachsposition bei Positionsmuster mit Rundachse (in Verbindung mit <_VARI>) | | |
| 26 | A1 | <S_AB1> | REAL | 2. Rundachsposition bei Positionsmuster mit Rundachse | | |
| 27 | A2 | <S_AB2> | REAL | 3. Rundachsposition bei Positionsmuster mit Rundachse | | |
| 28 | A3 | <S_AB3> | REAL | 4. Rundachsposition bei Positionsmuster mit Rundachse | | |
| 29 | A4 | <S_AB4> | REAL | 5. Rundachsposition bei Positionsmuster mit Rundachse | | |
| 30 | A5 | <S_AB5> | REAL | 6. Rundachsposition bei Positionsmuster mit Rundachse | | |
| 31 | A6 | <S_AB6> | REAL | 7. Rundachsposition bei Positionsmuster mit Rundachse | | |
| 32 | A7 | <S_AB7> | REAL | 8. Rundachsposition bei Positionsmuster mit Rundachse | | |
| 33 | A8 | <S_AB8> | REAL | 9. Rundachsposition bei Positionsmuster mit Rundachse | | |

**Hinweis**

Nicht benötigte Positionen der Parameter X1/Y1/A1 bis X8/Y8/A8 können weggelassen werden. Die Alternativwerte für <_XA>, <_YA> und <S_ABA> sind jedoch vollständig für alle 9 Positionen anzugeben.

Bei Positionsmuster XA, YB oder ZC (eine Geometrieachse und Rundachse) ist die Achse der Bearbeitungsebene, die über das Positionsmuster nicht verfahren wird (bei G17 und XA ist das Y) vor Zyklusaufruf zu positionieren.

### 3.25.1.37 CYCLE830 - Tieflochbohren 2

## Syntax

```
CYCLE830(<RTP>, <RFP>, <SDIS>, <_DP>, <FDEP>, <_DAM>, <DTB>, <DTS>,
<FRF>, <VARI>, <_MDEP>, <_VRT>, <_DTD>, <_DIS1>, <S_FP>, <S_SDAC2>,
<S_SV2>, <S_FB>, <_SDAC>, <_SV1>, <S_SPOS>, <S_ZA>, <S_FA>, <S_ZP>,
<S_FS>, <S_ZS1>, <S_ZS2>, <S_N>, <S_ZD>, <S_FD>, <S_FR>, <S_SDAC3>,
<S_SV3>, <S_CON>, <S_COFF>, <_GMODE>, <_DMODE>, <_AMODE>,
<S_AMODE2>, <S_AMODE3>, <S_ZPV>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | RP | <RTP> | REAL | Rückzugsebene (abs) | | |
| 2 | Z0 | <RFP> | REAL | Bezugspunkt (abs) | | |
| 3 | SC | <SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen) | | |
| 4 | Z1 | <_DP> | REAL | Endbohrtiefe abs/ink (siehe <_AMODE> EINER) | | |
| 5 | D | <FDEP> | REAL | 1. Bohrtiefe zum Spänebrechen/Entspanen absolut oder inkrementell bezogen auf den Bezugspunkt bei Anbohren/ohne Anbohren oder bezogen auf Pilotbohrungstiefe (siehe <_AMODE> ZEHNTAUSENDER) | | |
| 6 | DF | <_DAM> | REAL | Betrag/Prozentsatz für jede weitere Zustellung Degressionsbetrag oder -Prozentsatz (siehe <_AMODE> HUNDERTTAUSENDER) | | |
| 7 | DTB | <DTB> | REAL | Verweilzeit auf jeder Bohrtiefe (siehe <_AMODE> ZEHNER) | | |
| 8 | DTS | <DTS> | REAL | Verweilzeit beim Entspanen am Anfangspunkt (siehe <_AMODE> HUNDERTER) | | |
| 9 | FD1 | <FRF> | REAL | Prozentsatz für den Vorschub bei der ersten Zustellung (siehe <_AMODE> ZEHNMILLIONEN) | | |
| 10 | | <VARI> | INT | Bearbeitung | | |
| | | | | EINER: | Spänebrechen/Entspanen | |
| | | | | | 0 = | in einem Schnitt |
| | | | | | 1 = | Spänebrechen |
| | | | | | 2 = | Entspanen |
| | | | | | 3 = | Spänebrechen und Entspanen |
| | | | | ZEHNER: | Rückzug bei Entspanen | |
| | | | | | 0 = | auf Pilotbohrtiefe |
| | | | | | 1 = | auf Sicherheitsabstand |
| | | | | HUNDERTER: | weicher Anschnitt | |
| | | | | | 0 = | nein |
| | | | | | 1 = | ja |
| | | | | TAUSENDER: | Durchbohren | |
| | | | | | 0 = | nein |
| | | | | | 1 = | ja |
| | | | | ZEHNTAUSENDER: | Anbohren/Pilotbohrung | |
| | | | | | 0 = | ohne Anbohren |
| | | | | | 1 = | mit Anbohren |
| | | | | | 2 = | mit Pilotbohrung |
| | | | | HUNDERTTAUSENDER: | Rückzug | |
| | | | | | 0 = | auf Pilotbohrtiefe |
| | | | | | 1 = | auf Rückzugsebene |
| 11 | V1 | <_MDEP> | REAL | minimale Zustellung inkrementell (nur bei Prozentsatz für Degression) | | |
| 12 | V2 | <_VRT> | REAL | Rückzugsbetrag nach jeder Bearbeitung inkrementell (nur bei Spänebrechen) | | |
| | | | | | 0 = | Standardwert 1 mm |
| | | | | | > 0 = | variabler Rückzugsbetrag |
| 13 | DT | <_DTD> | REAL | Verweilzeit auf Endbohrtiefe (siehe <_AMODE> TAUSENDER) | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 14 | V3 | <_DIS1> | REAL | Vorhalteabstand inkrementell nur bei Entspanen (siehe <_AMODE> EINEMILLION) | | |
| 15 | FP | <S_FP> | REAL | Vorschub zum Einfahren in die Pilotbohrung als Wert oder in % (in Verbindung mit <S_AMODE2> HUNDERTER) | | |
| 16 | | <S_SDAC2> | INT | Spindeldrehrichtung beim Anfahren | | |
| | | | | | 3 = | M3 |
| | | | | | 4 = | M4 |
| | | | | | 5 = | M5 (Default) |
| 17 | SP | <S_SV2> | REAL | Anfahren mit | konstanter Spindeldrehzahl (siehe <S_AMODE2> ZEHNMILLIONEN) | |
| | V4 | | | | konstanter Schnittgeschwindigkeit | |
| | | | | Spindeldrehzahl in % von der Bohrdrehzahl | | |
| 18 | F | <S_FB> | REAL | Bohrvorschub (siehe <S_AMODE2> EINER) | | |
| 19 | | <_SDAC> | REAL | Spindeldrehrichtung beim Bohren | | |
| | | | | | 3 = | M3 |
| | | | | | 4 = | M4 |
| 20 | S | <_SV1> | REAL | Bohren mit | konstanter Spindeldrehzahl (siehe <S_AMODE2> EINEMILLION) | |
| | V5 | | | | konstanter Schnittgeschwindigkeit | |
| 21 | SPOS | <S_SPOS> | REAL | Spindelposition, nur wenn Anfahren mit M5 | | |
| 22 | ZA | <S_ZA> | REAL | Anbohrtiefe inkrementell bezogen auf Bezugspunkt oder absolut (siehe <S_AMODE3> EINER) | | |
| 23 | FA | <S_FA> | REAL | Anbohrvorschub als Wert oder in % (in Verbindung mit <S_AMODE2> ZEHNER) | | |
| 24 | ZP | <S_ZP> | REAL | Pilotbohrung inkrementell bezogen auf Bezugspunkt, absolut oder Faktor des Bohrungsdurchmessers (siehe <S_AMODE3> ZEHNER) | | |
| 25 | FS | <S_FS> | REAL | Vorschub Anschnitt als Wert oder in % (in Verbindung mit <S_AMODE2> TAUSENDER) | | |
| 26 | ZS1 | <S_ZS1> | REAL | Tiefe jedes Anschnitts mit konstantem Vorschub (ink) | | |
| 27 | ZS2 | <S_ZS2> | REAL | Tiefe jedes Anschnitts für Vorschuberhöhung (ink) | | |
| 28 | N | <S_N> | INT | Anzahl der Spänebrechhübe vor jedem Entspanen | | |
| 29 | ZD | <S_ZD> | REAL | Restbohrtiefe inkrementell bezogen auf Endbohrtiefe oder absolut (in Verbindung mit <S_AMODE3> HUNDERTER) | | |
| 30 | FD | <S_FD> | REAL | Restbohrvorschub als Wert oder in % (in Verbindung mit <S_AMODE2> ZEHNTAUSENDER) | | |
| 31 | FR | <S_FR> | REAL | Vorschub Rückzug (in Verbindung mit <S_AMODE2> HUNDERTTAUSENDER) | | |
| 32 | | <S_SDAC3> | INT | Spindeldrehrichtung beim Rückzug | | |
| | | | | | 3 = | M3 |
| | | | | | 4 = | M4 |
| | | | | | 5 = | M5 |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 33 | SR | <S_SV3> | REAL | Rückzug mit | konstanter Spindeldrehzahl (siehe <S_AMODE2> HUNDERTMILLIONEN) | |
| | V6 | | | | konstanter Schnittgeschwindigkeit | |
| | | | | Spindeldrehzahl in % von der Bohrdrehzahl | | |
| 34 | Kühlmittel ein | <S_CON> | STRING[10] | Kühlmittel ein, M-Befehl oder Unterprogrammaufruf | | |
| 35 | Kühlmittel aus | <S_COFF> | STRING[10] | Kühlmittel aus, M-Befehl oder Unterprogrammaufruf | | |
| 36 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | Bohrtiefe bezogen auf Spitze/Schaft | |
| | | | | | 0 = | Spitze |
| | | | | | 1 = | Schaft |
| 37 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | reserviert | |
| | | | | TAUSENDER: | reserviert | |
| | | | | ZEHNTAUSENDER: | Technologieskalierung innerhalb von Zyklenmasken (Seite 1172) | |
| | | | | | 0 = | Eingabe: komplett |
| | | | | | 1 = | Eingabe: einfach |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 38 | | <_AMODE> | INT | Alternativmodus 1 | | |
| | | | | EINER: | Bohrtiefe = Endbohrtiefe Z1 abs/ink | |
| | | | | | 0 = | inkrementell |
| | | | | | 1 = | absolut |
| | | | | ZEHNER: | Verweilzeit auf jeder Bohrtiefe DTB in Sekunden/Umdrehungen | |
| | | | | | 0 = | in Sekunden |
| | | | | | 1 = | in Umdrehungen |
| | | | | HUNDERTER: | Verweilzeit zum Entspanen DTS in Sekunden/Umdrehungen | |
| | | | | | 0 = | in Sekunden |
| | | | | | 1 = | in Umdrehungen |
| | | | | TAUSENDER: | Verweilzeit auf Endbohrtiefe DT in Sekunden/Umdrehungen | |
| | | | | | 0 = | in Sekunden |
| | | | | | 1 = | in Umdrehungen |
| | | | | ZEHNTAUSENDER: | 1. Bohrtiefe D abs/ink | |
| | | | | | 0 = | inkrementell |
| | | | | | 1 = | absolut |
| | | | | HUNDERTTAUSENDER: | Betrag/Prozentsatz DF für jede weitere Zustellung (Degression) | |
| | | | | | 0 = | Betrag |
| | | | | | 1 = | Prozentsatz (0,001 bis 100 %) |
| | | | | EINEMILLION: | Vorhalteabstand V3 automatisch/manuell | |
| | | | | | 0 = | automatisch (wird im Zyklus berechnet) |
| | | | | | 1 = | manuell (programmierter Wert) |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 39 | | <S_AMODE2> | INT | Alternativmodus 2 | | |
| | | | | EINER: | EINER: Bohrvorschub F | |
| | | | | | 0 = | F/min |
| | | | | | 1 = | F/U |
| | | | | ZEHNER: | Bewertung Anbohrvorschub FA | |
| | | | | | 0 = | in % vom Bohrvorschub |
| | | | | | 1 = | F/min |
| | | | | | 2 = | F/U |
| | | | | HUNDERTER: | Bewertung Vorschub zum Einfahren in Pilotbohrung FP | |
| | | | | | 0 = | in % vom Bohrvorschub |
| | | | | | 1 = | F/min |
| | | | | | 2 = | F/U |
| | | | | TAUSENDER: | Bewertung Vorschub Anschnitt FS | |
| | | | | | 0 = | in % vom Bohrvorschub |
| | | | | | 1 = | F/min |
| | | | | | 2 = | F/U |
| | | | | ZEHNTAUSENDER: | Bewertung Vorschub Durchbohren FD | |
| | | | | | 0 = | in % vom Bohrvorschub |
| | | | | | 1 = | F/min |
| | | | | | 2 = | F/U |
| | | | | HUNDERTTAUSENDER: | Vorschub Rückzug FR | |
| | | | | | 0 = | F/min |
| | | | | | 1 = | Eilgang |
| | | | | EINEMILLION: | Bohren - Spindeldrehzahl/Schnittgeschwindigkeit (S/V5) | |
| | | | | | 0 = | konstante Spindeldrehzahl |
| | | | | | 1 = | konstante Schnittgeschwindigkeit |
| | | | | ZEHNMILLIONEN: | Anfahren mit Spindeldrehzahl/Schnittgeschwindigkeit (SP/V4) | |
| | | | | | 0 = | konstante Spindeldrehzahl |
| | | | | | 1 = | konstante Schnittgeschwindigkeit |
| | | | | | 2 = | Spindeldrehzahl in % von der Bohrdrehzahl |
| | | | | HUNDERTMILLIONEN: | Rückzug - Spindeldrehzahl/Schnittgeschwindigkeit (SR/V6) | |
| | | | | | 0 = | konstante Spindeldrehzahl |
| | | | | | 1 = | konstante Schnittgeschwindigkeit |
| | | | | | 2 = | Spindeldrehzahl in % von der Bohrdrehzahl |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 40 | | <S_AMODE3> | INT | Alternativmodus 3 | | |
| | | | | EINER: | Anbohrtiefe ZA abs/ink | |
| | | | | | 0 = | inkrementell |
| | | | | | 1 = | absolut |
| | | | | ZEHNER: | Tiefe der Pilotbohrung ZP | |
| | | | | | 0 = | inkrementell |
| | | | | | 1 = | absolut |
| | | | | | 2 = | Faktor des Bohrungsdurchmessers |
| | | | | HUNDERTER: | Restbohrtiefe ZD ink/abs | |
| | | | | | 0 = | inkrementell |
| | | | | | 1 = | absolut |
| 41 | ZPV | <S_ZPV> | REAL | Vorhalteabstand inkrementell auf Pilotbohrungstiefe | | |

### 3.25.1.38 CYCLE832 - High Speed Settings

## Syntax

```
CYCLE832(<S_TOL>, <S_TOLM>, <S_OTOL>)
```

**Hinweis**

Der CYCLE832 entlastet den Maschinenhersteller nicht von notwendigen Optimierungsaufgaben bei der Inbetriebnahme der Maschine. Dies betrifft die Optimierung der an der Bearbeitung beteiligten Achsen und die Einstellungen der NCU (Vorsteuerung, Ruckbegrenzung, usw.).

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | Toleranz | <S_TOL> | REAL | Konturtoleranz<br>Die Konturtoleranz entspricht der Achstoleranz der Geometrieachsen. | | |
| 2 | | <S_TOLM> | INT | Bearbeitungsart (Technologie) | | |
| | | | | EINER: | | |
| | | | | | 0 = | Abwahl |
| | | | | | 1 = | Schlichten (Finish) |
| | | | | | 2 = | Vorschlichten (Semifinish) |
| | | | | | 3 = | Schruppen (Rough) |
| | | | | | 4 = | Feinschlichten (Precision) |
| | | | | ZEHNER: | | |
| | | | | | 0 = | Kompatibilität[1] bzw. keine Orientierungstoleranz |
| | | | | | 1 = | Orientierungstoleranz im Parameter <S_OTOL> |
| | | | | HUNDERTER<br>...<br>HUNDERT-TAUSENDER | belegt<br>wegen<br>Kompatibilität | |
| | | | | EINEMILLION: | | |
| | | | | | 0 = | Kompatibilität. Es wird automatisch die beste zur Verfügung stehende Formenbaufunktion verwendet:<br>• Option Top Surface nicht aktiv: ⇒ Advanced Surface<br>• Option Top Surface aktiv: ⇒ Top Surface mit Glättung |
| | | | | | 1 = | Top Surface ohne Glättung |
| | | | | | 2 = | Top Surface mit Glättung |
| 3 | ORI-Toleranz | <S_OTOL> | REAL | Orientierungstoleranz bzw. Versionskennzeichen CYCLE832<br>Toleranzparameter für die Orientierung des Werkzeugs.<br>Wird benötigt bei der Abarbeitung eines Hochgeschwindigkeitsbearbeitungsprogramms auf Maschinen mit dynamischer Orientierungstransformation (z. B. der 5-Achs-Bearbeitung).<br>Der Parameter <S_OTOL> **muss** programmiert werden. Dies gilt auch für Anwendungen auf 3-Achs-Maschinen bei Programmen ohne Orientierung des Werkzeugs (<S_OTOL> = 1). | | |

[1] Orientierungstoleranz abgeleitet von den Zyklen-Settingdaten SD55451 ... SD55454 (Orientierungstoleranz für Dynamikmodus ...) bzw. SD55445 ... SD55449 (Konturtoleranz für Dynamikmodus ...) multipliziert mit dem Faktor aus SD55441 ... SD55444.
**Weitere Informationen:** Inbetriebnahmehandbuch SINUMERIK Operate

### Klartexteingabe

Um die Lesbarkeit des Zyklenaufrufs zu verbessern, kann der Parameter <S_TOLM> (Bearbeitungsart) auch im Klartext eingegeben werden. Klartexte sind sprachunabhängig. Folgende Eingaben sind zulässig:

| | | | |
|---|---|---|---|
| _OFF | für | 0 | Abwahl |
| _FINISH | für | 1 | Schlichten |
| _SEMIFIN | für | 2 | Vorschlichten |
| _ROUGH | für | 3 | Schruppen |
| _PRECISION | für | 4 | Feinschlichten |
| _ORI_FINISH | für | 11 | Schlichten mit Eingabe einer Orientierungstoleranz |
| _ORI_SEMIFIN | für | 12 | Vorschlichten mit Eingabe einer Orientierungstoleranz |
| _ORI_ROUGH | für | 13 | Schruppen mit Eingabe einer Orientierungstoleranz |
| _ORI_PRECISION | für | 14 | Feinschlichten mit Eingabe einer Orientierungstoleranz |
| _TOP_SURFACE_SMOOTH_OFF | für | 1000000 | Top Surface ohne Glättung |
| _TOP_SURFACE_SMOOTH_ON | für | 2000000 | Top Surface mit Glättung |

Für die Klartexteingabe bei Top Surface werden die Klartexte wie im folgenden Beispiel kombiniert:

```
CYCLE832(0.1, _TOP_SURFACE_SMOOTH_OFF+_ORI_FINISH, 1)
```

### Hinweis

Die Klartexte sind an die Funktionsnamen der G-Gruppe 59 (Dynamikmodus für Bahninterpolation) angelehnt. Mit diesen Klartexten werden 3-Achs-Maschinen und Maschinen mit Mehrachsorientierungstransformation (TRAORI) in der Anwendung klar getrennt.

### Abwahl von CYCLE832

Bei Abwahl von CYCLE832 muss der Parameter <S_TOL> mit Null übergeben werden.

Beispiel: `CYCLE832(0,0,1)`

Die Syntax `CYCLE832()` ist für die Abwahl von CYCLE832 ebenfalls erlaubt.

## Beispiele

### Beispiel 1: CYCLE832 auf 3-Achs-Maschine ohne Orientierungstransformation

a) Zyklenaufruf mit Klartexteingabe

| Programmcode | Kommentar |
|---|---|
| G710 | ; Maßsystem ist metrisch. |

| **Programmcode** | **Kommentar** |
|---|---|
| CYCLE832(0.004,_FINISH,1) | ; Aufruf CYCLE832 mit: Konturtoleranz = 0,004 mm, Bearbeitungsart: Schlichten |
| ... | ; Abarbeitung eines Hochgeschwindigkeitsbearbeitungsprogramms |

b) Zyklenaufruf ohne Klartexteingabe

| **Programmcode** | **Kommentar** |
|---|---|
| G710 | ; s. o. |
| CYCLE832(0.004,1,1) | ; s. o. |
| ... | ; s. o. |

### Beispiel 2: CYCLE832 auf 5-Achs-Maschine mit Orientierungstransformation

a) Zyklenaufruf und Abwahl mit Klartexteingabe

| **Programmcode** | **Kommentar** |
|---|---|
| G710 | ; Maßsystem ist metrisch. |
| TRAORI | ; Orientierungstransformation aktivieren. |
| CYCLE832(0.3,_ORI_ROUGH,0.8) | ; Aufruf CYCLE832 mit: Konturtoleranz = 0,3 mm, Bearbeitungsart: Schruppen mit Eingabe einer Orientierungstoleranz, Orientierungstoleranz = 0,8 Grad |
| ... | ; Abarbeitung eines Hochgeschwindigkeitsbearbeitungsprogramms |
| CYCLE832(0,_OFF,1) | ; Konturtoleranz = 0, Bearbeitungsart: Abwahl CYCLE832, Orientierungstoleranz = 0 Grad |

b) Zyklenaufruf und Abwahl ohne Klartexteingabe

| **Programmcode** | **Kommentar** |
|---|---|
| G710 | ; s. o. |
| TRAORI | ; s. o. |
| CYCLE832(0.3,13,0.8) | ; s. o. |
| ... | ; s. o. |
| CYCLE832(0,0,1) | ; s. o. |

### 3.25.1.39 CYCLE840 - Gewindebohren mit Ausgleichsfutter

## Syntax

```
CYCLE840(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDR>, <SDAC>,
<ENC>, <MPIT>, <PIT>, <_AXN>, <_PITA>, <_TECHNO>, <_PITM>, <_PTAB>,
<_PTABA>, <_GMODE>, <_DMODE>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | | |
|---|---|---|---|---|---|---|---|
| 1 | RP | <RTP> | REAL | Rückzugsebene (abs) | | | |
| 2 | Z0 | <RFP> | REAL | Bezugspunkt (abs) | | | |
| 3 | SC | <SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) | | | |
| 4 | Z1 | <DP> | REAL | Bohrtiefe (abs), siehe <_AMODE> | | | |
| 5 | Z1 | <DPR> | REAL | Bohrtiefe (ink) , siehe <_AMODE> | | | |
| 6 | DT | <DTB> | REAL | Verweilzeit auf Bohrtiefe/auf Sicherheitsabstand nach Rückzug in Sekunden, siehe <ENC> | | | |
| 7 | | <SDR> | INT | Drehrichtung für den Rückzug | | | |
| 8 | SDE | <SDAC> | INT | Drehrichtung nach Zyklusende | | | |
| 9 | | <ENC> | INT | Gewindebohren mit Spindelgeber (G33)/Gewindebohren ohne Spindelgeber(G63) | | | |
| | | | | | 0 = | mit Spindelgeber | - Steigung aus <MPIT>/<PIT> - ohne DT |
| | | | | | 20 = | mit Spindelgeber | - Steigung aus <MPIT>/<PIT> - mit DT nach Rückzug auf Sicherheitsabstand |
| | | | | | 11 = | ohne Spindelgeber | - Steigung aus <MPIT>/<PIT> - mit DT auf Bohrtiefe |
| | | | | | 1 = | ohne Spindelgeber | - Steigung aus programmierten Vorschub - mit DT auf Bohrtiefe (Vorschub = Drehzahl · Steigung) |
| 10 | | <MPIT> | REAL | Gewindegröße nur für "ISO metrisch" (Steigung wird intern während der Laufzeit berechnet)<br>Wertebereich: 3 bis 48 (für M3 bis M48), alternativ zu <PIT> | | | |
| 11 | | <PIT> | REAL | Gewindesteigung als Wert, Maßeinheit siehe <_PITA><br>Wertebereich: > 0, alternativ zu MPIT | | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | | |
|---|---|---|---|---|---|---|---|
| 12 | | <_AXN> | INT | Bohrachse | 0 = | 3. Geometrieachse | |
| | | | | | 1 = | 1. Geometrieachse | |
| | | | | | 2 = | 2. Geometrieachse | |
| | | | | | ≥ 3 = | 3. Geometrieachse | |
| 13 | | <_PITA> | INT | Maßeinheit der Gewindesteigung (Auswertung <PIT> und <MPIT>) | | | |
| | | | | | 0 = | Steigung in mm | - Auswertung <MPIT>/<PIT> |
| | | | | | 1 = | Steigung in mm | - Auswertung <PIT> |
| | | | | | 2 = | Steigung in TPI | - Auswertung <PIT> (Gewindegänge pro inch) |
| | | | | | 3 = | Steigung in inch | - Auswertung <PIT> |
| | | | | | 4 = | MODUL | - Auswertung <PIT> |
| 14 | | <_TECHNO> | INT | Technologie[1] | | | |
| | | | | EINER: | Genauhaltverhalten | | |
| | | | | | 0 = | Genauhaltverhalten wie vor Zyklusaufruf aktiv | |
| | | | | | 1 = | Genauhalt G601 | |
| | | | | | 2 = | Genauhalt G602 | |
| | | | | | 3 = | Genauhalt G603 | |
| | | | | ZEHNER: | Vorsteuerung | | |
| | | | | | 0 = | mit/ohne Vorsteuerung wie vor Zyklusaufruf aktiv | |
| | | | | | 1 = | mit Vorsteuerung FFWON | |
| | | | | | 2 = | ohne Vorsteuerung FFWOF | |
| 15 | | <_PITM> | STRING[15] | String als Merker für Gewindesteigungseingabe[2] | | | |
| 16 | | <_PTAB> | STRING[5] | String für Gewindetabelle ("", "ISO", "BSW", "BSP", "UNC")[2] | | | |
| 17 | | <_PTABA> | STRING[20] | String für Auswahl in der Gewindetabelle (z.B. "M 10", "M 12", ...)[2] | | | |
| 18 | | <_GMODE> | INT | reserviert | | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 19 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | reserviert | |
| | | | | TAUSENDER: | Kompatibilitätsmode (nur für Eingabemaske zu Rückübersetzen), wenn MD 52216 Bit0 = 1[1] | |
| | | | | | 0 = | Technologieparameter werden angezeigt (Kompatibilität): TECHNO-Parameter wirken |
| | | | | | 1 = | Technologieparameter werden nicht angezeigt: Technologie "wie vor Zyklusaufruf programmiert" wirkt |
| | | | | ZEHNTAUSENDER: | Technologieskalierung innerhalb von Zyklenmasken (Seite 1172) | |
| | | | | | 0 = | Eingabe: komplett |
| | | | | | 1 = | Eingabe: einfach |
| 20 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Bohrtiefe Z1 (abs/ink) | |
| | | | | | 0 = | Kompatibilität, aus Programmierung <DP>/<DPR> |
| | | | | | 1 = | inkrementell |
| | | | | | 2 = | absolut |

[1] Felder Technologie können ausgeblendet sein in Abhängigkeit von Settingdatum SD52216 MCS_FUNCTION_MASK_DRILL

[2] Die Parameter 15, 16, und 17 werden nur bei der Gewindeauswahl in den Gewindetabellen der Eingabemaske verwendet. Ein Zugriff auf die Gewindetabellen über Zyklusdefinition bei Zykluslaufzeit ist nicht möglich!

### 3.25.1.40 CYCLE899 - Offene Nut

## Syntax

```
CYCLE899(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_LENG>, <_WID>, <_PA>,
<_PO>, <_STA>, <_MID>, <_MIDA>, <_FAL>, <_FALD>, <_FFP1>, <_CDIR>,
<_VARI>, <_GMODE>, <_DMODE>, <_AMODE>, <_UMODE>, <_FS>, <_ZFS>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | RP | <_RTP> | REAL | Rückzugsebene (abs) | | |
| 2 | Z0 | <_RFP> | REAL | Bezugspunkt der Werkzeugachse(abs) | | |
| 3 | SC | <_SDIS> | REAL | Sicherheitsabstand (additiv auf Bezugspunkt, ohne Vorzeichen einzugeben) | | |
| 4 | Z1 | <_DP> | REAL | Nuttiefe(abs/ink), siehe <_AMODE> | | |
| 5 | L | <_LENG> | REAL | Länge der Nut (ink) | | |
| 6 | W | <_WID> | REAL | Breite der Nut (ink) | | |
| 7 | X0 | <_PA> | REAL | Bezugspunkt/Startposition 1. Achse (abs) | | |
| 8 | Y0 | <_PO> | REAL | Bezugspunkt/Startposition 2. Achse (abs) | | |
| 9 | α0 | <_STA> | REAL | Drehwinkel zur 1.Achse | | |
| 10 | DZ | <_MID> | REAL | maximale Zustelltiefe (ink), nur für Wirbelfräsen | | |
| 11 | DXY | <_MIDA> | REAL | maximale Ebenenzustellung, siehe <_AMODE> | | |
| 12 | UXY | <_FAL> | REAL | Schlichtaufmaß Ebene | | |
| 13 | UZ | <_FALD> | REAL | Schlichtaufmaß Tiefe | | |
| 14 | F | <_FFP1> | REAL | Vorschub | | |
| 15 | | <_CDIR> | INT | Fräsrichtung | | |
| | | | | EINER: | | |
| | | | | | 0 = | Gleichlauf |
| | | | | | 1 = | Gegenlauf |
| | | | | | 4 = | Wechselweise |
| 16 | | <_VARI> | INT | Bearbeitung | | |
| | | | | EINER: | | |
| | | | | | 1 = | Schruppen |
| | | | | | 2 = | Schlichten |
| | | | | | 3 = | Schlichten Boden |
| | | | | | 4 = | Schlichten Rand |
| | | | | | 5 = | Vorschlichten |
| | | | | | 6 = | Anfasen |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | reserviert | |
| | | | | TAUSENDER: | | |
| | | | | | 1 = | Wirbelfräsen |
| | | | | | 2 = | Tauchfräsen |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 17 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Auswahl Bearbeitung/nur Startpunktberechnung | |
| | | | | | 1 = | normale Bearbeitung |
| | | | | TAUSENDER: | Bemaßung über Mitte / Kante | |
| | | | | | 0 = | Bemaßung über Mitte |
| | | | | | 1 = | Bemaßung über Kante "links" ("-" Richtung der 1.Achse) |
| | | | | | 2 = | Bemaßung über Kante "rechts" ("+" Richtung der 1.Achse) |
| 18 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | --- | reserviert |
| | | | | HUNDERTER: | --- | reserviert |
| | | | | TAUSENDER: | --- | reserviert |
| | | | | ZEHNTAUSENDER: | Technologieskalierung innerhalb von Zyklenmasken (Seite 1172) | |
| | | | | | 0 = | Eingabe: komplett |
| | | | | | 1 = | Eingabe: einfach |
| 19 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Nuttiefe Z1 | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |
| | | | | ZEHNER: | Einheit für Ebenenzustellung (<_MIDA>) DXY | |
| | | | | | 0 = | mm |
| | | | | | 1 = | % vom Werkzeugdurchmesser |
| | | | | HUNDERTER: | Eintauchtiefe bei Anfasen ZFS | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |
| 20 | | <_UMODE> | INT | reserviert | | |
| 21 | FS | <_FS> | REAL | Fasenbreite (ink) | | |
| 22 | ZFS | <_ZFS> | REAL | Eintauchtiefe (Werkzeugspitze) bei Anfasen (abs/ink), siehe <_AMODE>) | | |

### 3.25.1.41 CYCLE930 - Einstich

## Syntax

```
CYCLE930(<_SPD>, <_SPL>, <_WIDG>, <_WIDG2>, <_DIAG>, <_DIAG2>,
<_STA>, <_ANG1>, <_ANG2>, <_RCO1>, <_RCI1>, <_RCI2>, <_RCO2>,
<_FAL>, <_IDEP1>, <_SDIS>, <_VARI>, <_DN>, <_NUM>, <_DBH>, <_FF1>,
<_NR>, <_FALX>, <_FALZ>, <_DMODE>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | X0 | <_SPD> | REAL | Bezugspunkt in der Planachse (immer Durchmesser) | | |
| 2 | Z0 | <_SPL> | REAL | Bezugspunkt in der Längsachse | | |
| 3 | B1 | <_WIDG> | REAL | Einstichbreite unten | | |
| 4 | B2 | <_WIDG2> | REAL | Einstichbreite oben (nur für die Oberfläche) | | |
| 5 | T1 | <_DIAG> | REAL | Einstichtiefe am Bezugspunkt,<br>bei abs und Längsbearbeitung = Durchmesser, sonst ink | | |
| 6 | T2 | <_DIAG2> | REAL | Einstichtiefe gegenüber dem Bezugspunkt (nur für die Oberfläche),<br>bei abs und Längsbearbeitung = Durchmesser, sonst ink | | |
| 7 | α0 | <_STA> | REAL | Winkel der Schräge (-180 ≤ <_STA> ≤ 180) | | |
| 8 | α1 | <_ANG1> | REAL | Flankenwinkel 1 (0 ≤ <_ANG1> < 90) an der durch den Bezugspunkt bestimmten Seite des Einstichs | | |
| 9 | α2 | <_ANG2> | REAL | Flankenwinkel 2 (0 ≤ <_ANG2> < 90) gegenüber dem Bezugspunkt | | |
| 10 | R1/FS1 | <_RCO1> | REAL | Verrundungsradius oder Fasenbreite 1, außen am Bezugspunkt | | |
| 11 | R2/FS2 | <_RCI1> | REAL | Verrundungsradius oder Fasenbreite 2, innen am Bezugspunkt | | |
| 12 | R3/FS3 | <_RCI2> | REAL | Verrundungsradius oder Fasenbreite 3, innen gegenüber dem Bezugspunkt | | |
| 13 | R4/FS4 | <_RCO2> | REAL | Verrundungsradius oder Fasenbreite 4, außen gegenüber dem Bezugspunkt | | |
| 14 | U | <_FAL> | REAL | Schlichtaufmaß in X und Z, siehe <_VARI> (ZEHNTAUSENDER) (ohne Vorzeichen einzugeben) | | |
| 15 | D | <_IDEP1> | REAL | maximale Tiefenzustellung beim Eintauchen (ohne Vorzeichen einzugeben) | | |
| | | | | | 0 = | 1. Schnitt direkt auf ganze Tiefe |
| | | | | | > 0 = | 1. Schnitt <_IDEP1>, 2. Schnitt 2 · <_IDEP1> usw. |
| 16 | SC | <_SDIS> | REAL | Sicherheitsabstand (ohne Vorzeichen einzugeben) | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 17 | | <_VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | Technologische Bearbeitung | |
| | | | | | 1 = | Schruppen |
| | | | | | 2 = | Schlichten |
| | | | | | 3 = | Schruppen und Schlichten |
| | | | | HUNDERTER: | Lage Längs/Plan Außen/Innen +Z/+Z bzw. +X/-X | |
| | | | | | 1 = | Längs/Außen +Z |
| | | | | | 2 = | Plan/Innen -X |
| | | | | | 3 = | Längs/Innen +Z |
| | | | | | 4 = | Plan/Innen +X |
| | | | | | 5 = | Längs/Außen -Z |
| | | | | | 6 = | Plan/Außen -X |
| | | | | | 7 = | Längs/Innen -Z |
| | | | | | 8 = | Plan/Außen +X |
| | | | | TAUSENDER: | Lage Bezugspunkt | |
| | | | | | 0 = | Bezugspunkt oben |
| | | | | | 1 = | Bezugspunkt unten |
| | | | | ZEHNTAUSENDER: | Definition wie Schlichtaufmaße wirken | |
| | | | | | 0 = | Schlichtaufmaß U konturparallel |
| | | | | | 1 = | Schlichtaufmaße UX und UZ getrennt |
| 18 | | <_DN> | INT | D-Nummer für 2. Schneide des Werkzeugs | | |
| | | | | | > 0 = | D-Nummer für Werkzeugkorrektur der 2. Schneide des Einstechers |
| | | | | | 0 = | keine 2. Schneide programmiert |
| 19 | N | <_NUM> | INT | Anzahl der Einstiche (0 = 1 Einstich) | | |
| 20 | DP | <_DBH> | REAL | Abstand der Einstiche (nur erforderlich, wenn <_NUM> > 1) | | |
| 21 | F | <_FF1> | REAL | Vorschub | | |
| 22 | | <_NR> | INT | Kennung für die Einstichform entspricht den Vertikal-Softkeys zur Auswahl der Form | | |
| | | | | | 0 = | 90°-Flanken ohne Fasen/Verrundungen |
| | | | | | 1 = | schräge Flanken mit Fasen/ Verrundungen (ohne α0) |
| | | | | | 2 = | wie 1, aber auf Kegel (mit α0) |
| 23 | UX | <_FALX> | REAL | Schlichtaufmaß in X Achse, siehe <_VARI> (ZEHNTAUSENDER) (ohne Vorzeichen einzugeben) | | |
| 24 | UZ | <_FALZ> | REAL | Schlichtaufmaß in Z Achse, siehe <_VARI> (ZEHNTAUSENDER) (ohne Vorzeichen einzugeben) | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 25 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| 26 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Bemaßung Tiefe (nur für die Oberfläche) | |
| | | | | | 0 = | am Bezugspunkt |
| | | | | | 1 = | gegenüber dem Bezugspunkt |
| | | | | ZEHNER: | Tiefe | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |
| | | | | HUNDERTER: | Bemaßung Breite (nur für die Oberfläche) | |
| | | | | | 0 = | am Außendurchmesser (oben) |
| | | | | | 1 = | am Innendurchmesser (unten) |
| | | | | TAUSENDER: | Radius/Fase 1 (<_RCO1>) | |
| | | | | | 0 = | Radius |
| | | | | | 1 = | Fase |
| | | | | ZEHNTAUSENDER: | Radius/Fase 2 (<_RCI1>) | |
| | | | | | 0 = | Radius |
| | | | | | 1 = | Fase |
| | | | | HUNDERTTAUSENDER: | Radius/Fase 3 (<_RCI2>) | |
| | | | | | 0 = | Radius |
| | | | | | 1 = | Fase |
| | | | | EINEMILLION: | Radius/Fase 4 (<_RCO2>) | |
| | | | | | 0 = | Radius |
| | | | | | 1 = | Fase |

### 3.25.1.42 CYCLE940 - Freistich Form E und F / Freistich Gewinde

Mit dem Zyklus CYCLE940 können verschiedene Freistiche programmiert werden. Diese unterscheiden sich in der Parametrierung zum Teil erheblich.

Die zusätzlichen Spalten in der Tabelle zeigen, welcher Parameter bei welcher Freistichform benötigt wird. Sie entsprechen den vertikalen Auswahlsoftkeys in der Zyklenmaske:

- E: Freistich Form E
- F: Freistich Form F
- A-D: Freistich Gewinde DIN (Formen A-D)
- T: Freistich Gewinde (freie Definition der Form)

## Syntax

```
CYCLE940(<_SPD>, <_SPL>, <_FORM>, <_LAGE>, <_SDIS>, <_FFP>, <_VARI>,
<_EPD>, <_EPL>, <_R1>, <_R2>, <_STA>, <_VRT>, <_MID>, <_FAL>,
<_FALX>, <_FALZ>, <_PITI>, <_PTAB>, <_PTABA>, <_DMODE>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Progr. bei Form | | | | Bedeutung | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | E | F | A-D | T | | | |
| 1 | X0 | <_SPD> | REAL | x | x | x | x | Bezugspunkt in der Planachse (immer Durchmesser) | | |
| 2 | Z0 | <_SPL> | REAL | x | x | x | x | Bezugspunkt Längsachse (abs) | | |
| 3 | FORM | <_FORM> | CHAR | x | x | x | x | Form des Freistichs (Großbuchstaben, z.B. "T")<br>Auswahl, aus welcher Tabelle die Freistichwerte genommen werden sollen | | |
| | | | | | | | | | A = | außen, Bezug DIN76, A = normal |
| | | | | | | | | | B = | außen, Bezug DIN76, B = kurz |
| | | | | | | | | | C = | innen, Bezug DIN76, C = normal |
| | | | | | | | | | D = | innen, Bezug DIN76, D = kurz |
| | | | | | | | | | E = | Bezug DIN509 |
| | | | | | | | | | F = | Bezug DIN509 |
| | | | | | | | | | T = | Freiform |
| 4 | LAGE | <_LAGE> | INT | x | x | x | x | Lage des Freistichs (parallel Z) | 0 = | Außen +Z: ____\| |
| | | | | | | | | | 1 = | Außen -Z: \|____/ |
| | | | | | | | | | 2 = | Innen +Z: /-----\| |
| | | | | | | | | | 3 = | Innen -Z: \|-----\ |
| 5 | SC | <_SDIS> | | x | x | x | x | Sicherheitsabstand (ink) | | |
| 6 | F | <_FFP> | | x | x | x | x | Bearbeitungsvorschub (mm/U) | | |
| 7 | | <_VARI> | INT | - | - | x | x | Bearbeitungsart | | |
| | | | | | | | | EINER: | Bearbeitung | |
| | | | | | | | | | 1 = | Schruppen |
| | | | | | | | | | 2 = | Schlichten |
| | | | | | | | | | 3 = | Schruppen + Schlichten |
| | | | | | | | | ZEHNER: | Bearbeitungsstrategie | |
| | | | | | | | | | 0 = | konturparallel |
| | | | | | | | | | 1 = | längs |
| | | | | Freistiche Form E und F werden immer in einem Zug wie Schlichten bearbeitet. | | | | | | |
| 8 | X1 | <_EPD> | | x | x | - | - | Aufmaß X (abs/ink), siehe <_AMODE> | | |
| | | | | - | - | - | x | Freistichtiefe (abs/ink ), siehe <_AMODE> | | |
| 9 | Z1 | <_EPL> | | - | x | - | - | Aufmaß Z | | |
| | | | | - | - | - | x | Freistichbreite (abs/ink), siehe <_AMODE> | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Progr. bei Form | | | | Bedeutung | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | R1 | <_R1> | | - | - | - | x | Verrundungsradius an der Schrägen | | | |
| 11 | R2 | <_R2> | | - | - | - | x | Verrundungsradius in der Ecke | | | |
| 12 | α | <_STA> | | - | - | x | x | Eintauchwinkel | | | |
| 13 | VX | <_VRT> | | x | x | - | - | Planzug X (abs/ink), siehe <_AMODE> | | | |
| | | | | - | - | x | x | Planzug X bei Schlichten, (abs/ink), siehe <_AMODE> | | | |
| 14 | D | <_MID> | | - | - | x | x | Tiefenzustellung | | | |
| 15 | U | <_FAL> | | - | - | x | x | Schlichtaufmaß konturparallel, siehe <_AMODE> | | | |
| 16 | UX | <_FALX> | | - | - | x | x | Schlichtaufmaß X | | | |
| 17 | UZ | <_FALZ> | | - | - | x | x | Schlichtaufmaß Z | | | |
| 18 | P | <_PITI> | INT | - | - | x | - | Auswahl der Steigung, Form A-D, entspricht M1 ... M68 | | | |
| | | | | | | | | 0 = 0.20<br>1 = 0.25<br>2 = 0.30<br>3 = 0.35<br>4 = 0.40<br>5 = 0.45 | 6 = 0.50<br>7 = 0.60<br>8 = 0.70<br>9 = 0.75<br>10 = 0.80<br>11 = 1.00 | 12 = 1.25<br>13 = 1.50<br>14 = 1.75<br>15 = 2.00<br>16 = 2.50<br>17 = 3.00 | 18 = 3.50<br>19 = 4.00<br>20 = 4.50<br>21 = 5.00<br>22 = 5.50<br>23 = 6.00 |
| | | | | x | x | - | - | Auswahl Radius/Tiefe, Form E, F | | | |
| | | | | | | | | 0 = 0.6 · 0.3<br>1 = 1.0 · 0.4<br>2 = 1.0 · 0.2<br>3 = 1.6 ··0.3 | 4 = 2.5 · 0.4<br>5 = 4.0 · 0.5<br>6 = 0.4 · 0.2<br>7 = 0.6 · 0.2 | 8 = 0.1 · 0.1<br>9 = 0.2 ··0.1 | |
| 19 | | <_PTAB> | STRING [5] | | | | | String für Gewindetabelle ("", "ISO", "BSW", "BSP", "UNC") (nur für Oberfläche) | | | |
| 20 | | <_PTABA> | STRING [20] | | | | | String für Auswahl in der Gewindetabelle (z.B. "M 10", "M 12", ...) (nur für die Oberfläche) | | | |
| 21 | | <_DMODE> | INT | | | | | Anzeigemodus | | | |
| | | | | x | x | x | x | EINER: | Bearbeitungsebene G17/G18/G19 | | |
| | | | | | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv | |
| | | | | | | | | | 1 = | G17 (nur im Zyklus aktiv) | |
| | | | | | | | | | 2 = | G18 (nur im Zyklus aktiv) | |
| | | | | | | | | | 3 = | G19 (nur im Zyklus aktiv) | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Progr. bei Form | | | | Bedeutung | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 22 | | <_AMODE> | INT | | | | | Alternativmodus | | |
| | | | | x | x | - | x | EINER: | Parameter <_EPD> Aufmaß X oder Freistichtiefe | |
| | | | | | | | | | 0 = | absolut (immer Durchmesser) |
| | | | | | | | | | 1 = | inkrementell |
| | | | | x | x | - | x | ZEHNER: | Parameter <_EPL> Aufmaß Z oder Freistichbreite | |
| | | | | | | | | | 0 = | absolut |
| | | | | | | | | | 1 = | inkrementell |
| | | | | x | x | x | x | HUNDERTER: | Parameter <_VRT> Planzug X | |
| | | | | | | | | | 0 = | absolut (immer Durchmesser) |
| | | | | | | | | | 1 = | inkrementell |
| | | | | - | - | x | x | TAUSENDER: | Schlichtaufmaß | |
| | | | | | | | | | 0 = | Schlichtaufmaß konturparallel (<_FAL>) |
| | | | | | | | | | 1 = | Schlichtaufmaß getrennt (<_FALX>/<_FALZ>) |

### 3.25.1.43 CYCLE951 - Abspanen

## Syntax

```
CYCLE951(<_SPD>, <_SPL>, <_EPD>, <_EPL>, <_ZPD>, <_ZPL>, <_LAGE>,
<_MID>, <_FALX>, <_FALZ>, <_VARI>, <_RF1>, <_RF2>, <_RF3>, <_SDIS>,
<_FF1>, <_NR>, <_DMODE>, <_AMODE>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung |
|---|---|---|---|---|
| 1 | X0 | <_SPD> | REAL | Bezugspunkt (abs, immer Durchmesser) |
| 2 | Z0 | <_SPL> | REAL | Bezugspunkt (abs) |
| 3 | X1 | <_EPD> | REAL | Endpunkt |
| 4 | Z1 | <_EPL> | REAL | Endpunkt |
| 5 | XM<br>α1<br>α2 | <_ZPD> | REAL | Zwischenpunkt, siehe <_DMODE> (ZEHNER) |
| 6 | ZM<br>α1<br>α2 | <_ZPL> | REAL | Zwischenpunkt, siehe <_DMODE> (ZEHNER) |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 7 | Lage | <_LAGE> | INT | Lage der Abspanecke | 0 = | außen/hinten |
| | | | | | 1 = | außen/vorne |
| | | | | | 2 = | innen/hinten |
| | | | | | 3 = | innen/vorne |
| 8 | D | <_MID> | REAL | maximale Tiefenzustellung beim Eintauchen | | |
| 9 | UX | <_FALX> | REAL | Schlichtaufmaß in X | | |
| 10 | UZ | <_FALZ> | REAL | Schlichtaufmaß in Z | | |
| 11 | | <_VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | Abspanrichtung (längs oder plan) im Koordinatensystem | |
| | | | | | 1 = | längs |
| | | | | | 2 = | plan |
| | | | | ZEHNER: | | |
| | | | | | 1 = | Schruppen bis Schlichtaufmaß |
| | | | | | 2 = | Schlichten |
| | | | | HUNDERTER: | reserviert | |
| | | | | TAUSENDER: | reserviert | |
| | | | | ZEHNTAUSENDER: | reserviert | |
| 12 | R1/FS1 | <_RF1> | REAL | Verrundungsradius oder Fasenbreite 1, siehe <_AMODE> (ZEHNTAUSENDER) | | |
| 13 | R2/FS2 | <_RF2> | REAL | Verrundungsradius oder Fasenbreite 2, siehe <_AMODE> (HUNDERTTAUSENDER) | | |
| 14 | R3/FS3 | <_RF3> | REAL | Verrundungsradius oder Fasenbreite 3, siehe <_AMODE> (EINEMILLION) | | |
| 15 | SC | <_SDIS> | REAL | Sicherheitsabstand | | |
| 16 | F | <_FF1> | REAL | Vorschub für Schruppen/Schlichten | | |
| 17 | | <_NR> | INT | Kennung der Abspanart (entspricht den Vertikal-Softkeys zur Auswahl der Form): | | |
| | | | | | 0 = | Abspanen 1, 90 Grad Ecke ohne Fasen/Verrundungen |
| | | | | | 1 = | Abspanen 2, 90 Grad Ecke mit Fasen/Verrundungen |
| | | | | | 2 = | Abspanen 3, beliebige Ecke mit Fasen/Verrundungen |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 18 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | Eingabeform <_ZPD>/<_ZPL> | |
| | | | | | 0 = | Xm/Zm |
| | | | | | 1 = | Xm/α1 |
| | | | | | 2 = | Xm/α2 |
| | | | | | 3 = | α1/Zm |
| | | | | | 4 = | α2/Zm |
| | | | | | 5 = | α1/α2 |
| 21 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Zwischenpunkt in X | |
| | | | | | 0 = | absolut, Wert der Planachse im Durchmesser |
| | | | | | 1 = | inkrementell, Wert der Planachse im Radius |
| | | | | ZEHNER: | Zwischenpunkt in Z | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |
| | | | | HUNDERTER: | Endpunkt in X | |
| | | | | | 0 = | absolut, Wert der Planachse im Durchmesser |
| | | | | | 1 = | inkrementell, Wert der Planachse im Radius |
| | | | | TAUSENDER: | Endpunkt in Z | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |
| | | | | ZEHNTAUSENDER: | Radius/Fase 1 | |
| | | | | | 0 = | Radius |
| | | | | | 1 = | Fase |
| | | | | HUNDERTTAUSENDER: | Radius/Fase 2 | |
| | | | | | 0 = | Radius |
| | | | | | 1 = | Fase |
| | | | | EINEMILLION: | Radius/Fase 3 | |
| | | | | | 0 = | Radius |
| | | | | | 1 = | Fase |

### 3.25.1.44 CYCLE952 - Abspanen / Abspanen Rest / Stechen / Stechen Rest / Stechdrehen / Stechdrehen Rest

## Syntax

```
CYCLE952(<_PRG>, <_CON>, <_CONR>, <_VARI>, <_F>, <_FR>, <_RP>, <_D>,
<_DX>, <_DZ>, <_UX>, <_UZ>, <_U>, <_U1>, <_BL>, <_XD>, <_ZD>, <_XA>,
<_ZA>, <_XB>, <_ZB>, <_XDA>, <_XDB>, <_N>, <_DP>, <_DI>, <_SC>,
<_DN>, <_GMODE>, <_DMODE>, <_AMODE>, <_PK>, <_DCH>, <_FS>)
```

## Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung |
|---|---|---|---|---|
| 1 | PRG | <_PRG> | STRING[100] | Name des Abspanprogramms |
| 2 | CON | <_CON> | STRING[100] | Name des Programms, aus dem die aktualisierte Rohteilkontur gelesen wird (bei Restbearbeitung) |
| 3 | CONR | <_CONR> | STRING[100] | Name des Programms, in das die aktualisierte Rohteilkontur (siehe <_AMODE> ZEHNTAUSENDER) geschrieben wird |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 4 | | <_VARI> | INT | Bearbeitungsart | | |
| | | | | EINER: | Abspanart | |
| | | | | | 1 = | längs |
| | | | | | 2 = | plan |
| | | | | | 3 = | konturparallel |
| | | | | ZEHNER: | Technologische Bearbeitung (siehe <_GMODE> HUNDERTER) | |
| | | | | | 1 = | Schruppen |
| | | | | | 2 = | Schlichten |
| | | | | | 3 = | reserviert |
| | | | | | 4 = | Schruppen zweikanalig |
| | | | | | 5 = | Schlichten zweikanalig |
| | | | | HUNDERTER: | Bearbeitungsrichtung | |
| | | | | | 1 = | Bearbeitungsrichtung X - |
| | | | | | 2 = | Bearbeitungsrichtung X + |
| | | | | | 3 = | Bearbeitungsrichtung Z - |
| | | | | | 4 = | Bearbeitungsrichtung Z + |
| | | | | TAUSENDER: | Zustellrichtung | |
| | | | | | 1 = | Außen X - |
| | | | | | 2 = | Innen X + |
| | | | | | 3 = | Stirnseite Z - |
| | | | | | 4 = | Rückseite Z + |
| | | | | ZEHNTAUSENDER: | Definition, wie Schlichtaufmaße wirken | |
| | | | | | 0 = | Schlichtaufmaße UX und UZ getrennt |
| | | | | | 1 = | Schlichtaufmaß U konturparallel |
| | | | | HUNDERTTAUSENDER: | Nachziehen | |
| | | | | | 0 = | Kompatibilität, Nachziehen automatisch |
| | | | | | 1 = | mit Nachziehen an der Kontur |
| | | | | | 2 = | ohne Nachziehen |
| | | | | | 3 = | Nachziehen automatisch |
| | | | | EINEMILLION: | Hinterschnitte | |
| | | | | | 0 = | Stelle wird nicht ausgewertet bei Stechen, - Rest und Stechdrehen, - Rest |
| | | | | | 1 = | Hinterschnitte bearbeiten |
| | | | | | 2 = | Hinterschnitte nicht bearbeiten |
| | | | | ZEHNMILLIONEN: | hinter/vor Drehmitte | |
| | | | | | 0 = | Bearbeitung vor der Drehmitte |
| | | | | | 1 = | reserviert |
| 5 | F | <_F> | REAL | Vorschub für Schruppen/Schlichten | | |
| | FZ | | | Vorschub Abszisse Stechdrehen | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 6 | FR | <_FR> | REAL | Vorschub zum Eintauchen in Hinterschnitte Schruppen | | |
| | FX | | | Vorschub Ordinate Stechdrehen | | |
| 7 | RP | <_RP> | REAL | Rückzugsebene bei Innenbearbeitung (abs., immer Durchmesser) | | |
| 8 | D | <_D> | REAL | Zustellung Schruppen (siehe <_AMODE> EINER) | | |
| 9 | DX | <_DX> | REAL | Zustellung X (siehe <_AMODE> EINER) | | |
| 10 | DZ | <_DZ> | REAL | Zustellung Z (siehe <_AMODE> EINER) | | |
| 11 | UX | <_UX> | REAL | Schlichtaufmaß X (siehe <_VARI> ZEHNTAUSENDER) | | |
| 12 | UZ | <_UZ> | REAL | Schlichtaufmaß Z (siehe <_VARI> ZEHNTAUSENDER) | | |
| 13 | U | <_U> | REAL | Schlichtaufmaß konturparallel (siehe <_VARI> ZEHNTAUSENDER) | | |
| 14 | U1 | <_U1> | REAL | zusätzliches Schlichtaufmaß beim Schlichten (siehe <_AMODE> TAUSENDER) | | |
| 15 | BL | <_BL> | INT | Rohteildefinition | 1 = | Zylinder mit Aufmaß |
| | | | | | 2 = | Aufmaß auf Fertigteilkontur |
| | | | | | 3 = | Rohteilkontur ist angegeben |
| 16 | XD | <_XD> | REAL | Rohteildefinition X (siehe <_AMODE> HUNDERTTAUSENDER) | | |
| 17 | ZD | <_ZD> | REAL | Rohteildefinition Z (siehe <_AMODE> EINEMILLION) | | |
| 18 | XA | <_XA> | REAL | Grenze 1 X (abs., immer Durchmesser) | | |
| 19 | ZA | <_ZA> | REAL | Grenze 1 Z (abs.) | | |
| 20 | XB | <_XB> | REAL | Grenze 2 X (siehe <_AMODE> ZEHNMILLIONEN) | | |
| 21 | ZB | <_ZB> | REAL | Grenze 2 Z (siehe <_AMODE> HUNDERTMILLIONEN) | | |
| 22 | XDA | <_XDA> | REAL | Einstechgrenze 1 für die 1. Einstichposition auf der Stirnseite (abs., immer Durchmesser) | | |
| 23 | XDB | <_XDB> | REAL | Einstechgrenze 2 für die 1. Einstichposition auf der Stirnseite (abs., immer Durchmesser) | | |
| 24 | N | <_N> | INT | Anzahl der Einstiche | | |
| 25 | DP | <_DP> | REAL | Abstand der Einstiche | Längs-Einstich: parallel zur Z-Achse | |
| | | | | | Plan-Einstich: parallel zur X-Achse | |
| 26 | DI | <_DI> | REAL | Abstand für Vorschubunterbrechung | 0 = | keine Unterbrechung |
| | | | | | > 0 = | mit Unterbrechung |
| 27 | SC | <_SC> | REAL | Sicherheitsabstand zum Umfahren von Hindernissen, inkremental | | |
| 28 | D2 | <_DN> | INT | D-Nummer für 2. Schneide, wenn nicht programmiert ⇒ D+1 | | |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 29 | | <_GMODE> | INT | Geometriemodus (Bewertung der programmierten Geometriewerte) | | |
| | | | | EINER: | reserviert | |
| | | | | ZEHNER: | reserviert | |
| | | | | HUNDERTER: | Auswahl Bearbeitung / nur Startpunktberechnung | |
| | | | | | 0 = | normale Bearbeitung (kein Kompatibilitätsmodus erforderlich) |
| | | | | | 1 = | normale Bearbeitung |
| | | | | | 2 = | Startposition berechnen - keine Bearbeitung (nur für Aufruf aus ShopMill/ShopTurn) |
| | | | | TAUSENDER: | Eingrenzung | |
| | | | | | 0 = | nein |
| | | | | | 1 = | ja |
| | | | | ZEHNTAUSENDER: | Grenze 1 X eingeben | |
| | | | | | 0 = | nein |
| | | | | | 1 = | ja |
| | | | | HUNDERTTAUSENDER: | Grenze 2 X eingeben | |
| | | | | | 0 = | nein |
| | | | | | 1 = | ja |
| | | | | EINEMILLION: | Grenze 1 Z eingeben | |
| | | | | | 0 = | nein |
| | | | | | 1 = | ja |
| | | | | ZEHNMILLIONEN: | Grenze 2 Z eingeben | |
| | | | | | 0 = | nein |
| | | | | | 1 = | ja |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 30 | | <_DMODE> | INT | Anzeigemodus | | |
| | | | | EINER: | Bearbeitungsebene G17/G18/G19 | |
| | | | | | 0 = | Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv |
| | | | | | 1 = | G17 (nur im Zyklus aktiv) |
| | | | | | 2 = | G18 (nur im Zyklus aktiv) |
| | | | | | 3 = | G19 (nur im Zyklus aktiv) |
| | | | | ZEHNER: | Technologiemodus | |
| | | | | | 1 = | Konturabspanen |
| | | | | | 2 = | Konturstechen |
| | | | | | 3 = | Stechdrehen |
| | | | | HUNDERTER: | Restmaterial bearbeiten | |
| | | | | | 0 = | nein |
| | | | | | 1 = | ja |
| | | | | TAUSENDER: | --- | reserviert |
| | | | | ZEHNTAUSENDER: | Technologieskalierung innerhalb von Zyklenmasken (Seite 1172) | |
| | | | | | 0 = | Eingabe: komplett |
| | | | | | 1 = | Eingabe: einfach |
| | | | | HUNDERTTAUSENDER: | Programmname automatisch | |
| | | | | | 0 = | nein |
| | | | | | 1 = | ja |

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 31 | | <_AMODE> | INT | Alternativmodus | | |
| | | | | EINER: | Auswahl Zustellung | |
| | | | | | 0 = | Zustellung DX und DZ bei Abspanart konturparallel |
| | | | | | 1 = | Zustellung D |
| | | | | ZEHNER: | Zustellstrategie | |
| | | | | | 0 = | variable Schnitttiefe (90 ... 100 %) |
| | | | | | 1 = | konstante Schnitttiefe |
| | | | | HUNDERTER: | Schnittaufteilung | |
| | | | | | 0 = | gleichmäßig |
| | | | | | 1 = | an Kanten ausrichten |
| | | | | TAUSENDER: | Auswahl Konturaufmaß U1, Doppelschlichten | |
| | | | | | 0 = | nein |
| | | | | | 1 = | ja |
| | | | | ZEHNTAUSENDER: | Auswahl Rohteil aktualisieren | |
| | | | | | 0 = | nein |
| | | | | | 1 = | ja |
| | | | | HUNDERTTAUSENDER: | Auswahl Aufmaß auf Rohteil XD | |
| | | | | | 0 = | absolut, Wert der Planachse im Durchmesser |
| | | | | | 1 = | inkremental, Wert der Planachse im Radius |
| | | | | EINEMILLION: | Auswahl Aufmaß auf Rohteil ZD | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |
| | | | | ZEHNMILLIONEN: | Auswahl Grenze 2 XB | |
| | | | | | 0 = | absolut, Wert der Planachse im Durchmesser |
| | | | | | 1 = | inkremental, Wert der Planachse im Radius |
| | | | | HUNDERTMILLION: | Auswahl Grenze 2 ZB | |
| | | | | | 0 = | absolut |
| | | | | | 1 = | inkrementell |
| | | | | EINEMILLIARDE: | | |
| | | | | | 0 = | Führungskanal |
| | | | | | 1 = | Folgekanal |
| 32 | | <_PK> | INT | Nummer des Partnerkanals, wenn mehr als 2 Kanäle an der Maschine vorhanden sind | | |
| 33 | DCH | <_DCH> | REAL | Kanalversatz | | |
| 34 | FS | <_FS> | REAL | Vorschub Schlichten bei Komplettbearbeitung | | |

### 3.25.1.45 CYCLE4071 - Längsschleifen mit Zustellung am Umkehrpunkt

## Syntax

```
CYCLE4071(<S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>, <S_H>, <S_A1>,
<S_A2>)
```

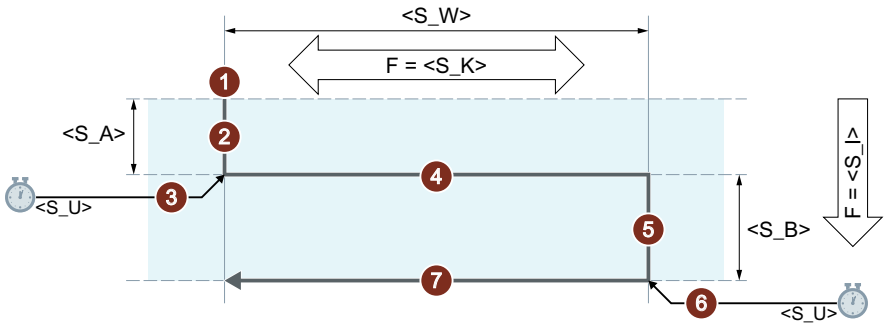
## Parameter

| Nr. | Parameter | Datentyp | Bedeutung |
|---|---|---|---|
| 1 | <S_A> | REAL | Zustelltiefe am Anfang |
| 2 | <S_B> | REAL | Zustelltiefe am Ende |
| 3 | <S_W> | REAL | Schleifbreite |
| 4 | <S_U> | REAL | Ausfeuerzeit |
| 5 | <S_I> | REAL | Vorschub für Zustellung |
| 6 | <S_K> | REAL | Vorschub für Querzustellung |
| 7 | <S_H> | INT | Anzahl Wiederholungen |
| 8 | <S_A1> | AXIS | Zustellachse (optional) bzw. 1. Geometrieachse |
| 9 | <S_A2> | AXIS | Pendelachse (optional) bzw. 2. Geometrieachse |

## Funktion

Der Zyklus dient zum Abarbeiten von sich wiederholenden Zustellungen. Dabei kann die Zustelltiefe am Anfang und am Ende unterschiedlich sein. Zwischen dem Zustellen erfolgt eine Tangentialbewegung.

## Ablauf




① Start des Zyklus an der aktuellen Position der Pendelachse.

② Verfahren der Zustellachse auf die Zustelltiefe am Anfang `<S_A>` mit dem Vorschub für Zustellung `<S_I>`.

③ Ausfeuern mit der Ausfeuerzeit `<S_U>`.

④ Verfahren der Pendelachse mit der Schleifbreite `<S_W>` als Verfahrweg und dem Vorschub für Querzustellung `<S_K>`.

⑤ Verfahren der Zustellachse auf die Zustelltiefe am Ende `<S_B>` mit dem Vorschub für Zustellung `<S_I>`.

⑥ Ausfeuern mit der Ausfeuerzeit `<S_U>`.

⑦ Verfahren der Pendelachse mit der Schleifbreite `<S_W>` als Verfahrweg auf den Anfangspunkt und dem Vorschub für Querzustellung `<S_K>`.

Kennzeichnet die sich wiederholenden Ablaufschritte.
Der Ablauf wird so oft wiederholt, bis die programmierte Anzahl Wiederholungen `<S_H>` erreicht ist.

### Hinweis

Der Ablauf ist mit Einzelsatz nicht unterbrechbar.

## Beispiel

Zwei Pendelbewegung mit folgenden Zyklenparameter ausführen:

- Zustelltiefe am Anfang: 0,02 mm
- Zustelltiefe am Ende: 0,01 mm
- Hub: 100 mm
- Ausfeuerzeit: 1 s
- Vorschub Zustellung: 1 mm/min
- Quervorschub: 1000 mm/min
- Wiederholungen: 2
- Pendel- und Zustellachse: Standardgeometrieachsen

**Programmcode**

```
N10 T1 D1
N20 CYCLE4071(0.02,0.01,100,1,1,1000,2)
N30 M30
```

### 3.25.1.46 CYCLE4072 - Längsschleifen mit Zustellung am Umkehrpunkt und Abbruchsignal

## Syntax

```
CYCLE4072(<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>,
<S_H>, <S_A1>, <S_A2>)
```

## Parameter

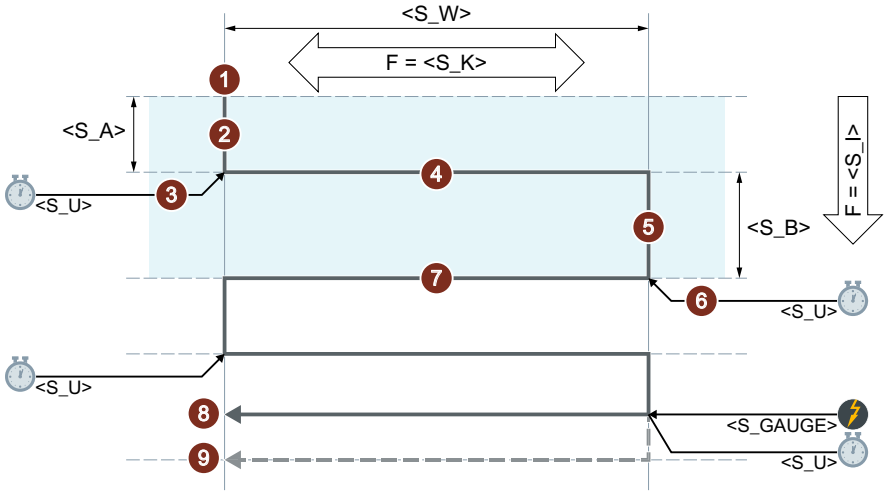
| Nr. | Parameter | Datentyp | Bedeutung |
|---|---|---|---|
| 1 | <S_GAUGE> | STRING | Abbruchbedingungen für Zustellung:<br>1. Nummer eines schnellen Eingangs<br>2. Logischer Ausdruck |
| 2 | <S_A> | REAL | Zustelltiefe am Anfang |
| 3 | <S_B> | REAL | Zustelltiefe am Ende |
| 4 | <S_W> | REAL | Schleifbreite |
| 5 | <S_U> | REAL | Ausfeuerzeit |
| 6 | <S_I> | REAL | Vorschub für Zustellung |
| 7 | <S_K> | REAL | Vorschub für Querzustellung |
| 8 | <S_H> | INT | Anzahl Wiederholungen |
| 9 | <S_A1> | AXIS | Zustellachse (optional) bzw. 1. Geometrieachse |
| 10 | <S_A2> | AXIS | Pendelachse (optional) bzw. 2. Geometrieachse |

## Funktion

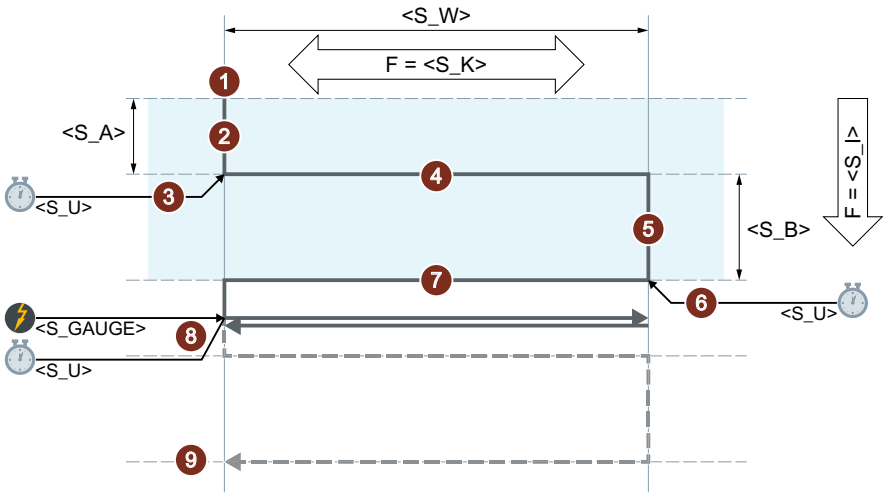
Der Zyklus dient zum Abarbeiten von sich wiederholenden Zustellungen unter Berücksichtigung eines externen Abbruchsignals. Die Zustelltiefe kann am Anfang und am Ende unterschiedlich sein. Zwischen dem Zustellen erfolgt eine Tangentialbewegung. Die Tiefenzustellung wird abgebrochen, wenn die Abbruchbedingung erfüllt ist. Nach dem Abbruch der Tiefenzustellung wird immer ein kompletter Hub durchgeführt.

## Ablauf

### Abbruch der Zustellung am Ende



### Abbruch der Zustellung am Anfang




① Start des Zyklus an der aktuellen Position der Pendelachse.

② Verfahren der Zustellachse auf die Zustelltiefe am Anfang `<S_A>` mit dem Vorschub für Zustellung `<S_I>`.

③ Ausfeuern mit der Ausfeuerzeit `<S_U>`.

④ Verfahren der Pendelachse mit der Schleifbreite `<S_W>` als Verfahrweg und dem Vorschub für Querzustellung `<S_K>`.

⑤ Verfahren der Zustellachse auf die Zustelltiefe am Ende `<S_B>` mit dem Vorschub für Zustellung `<S_I>`.

⑥ Ausfeuern mit der Ausfeuerzeit `<S_U>`.

⑦ Verfahren der Pendelachse mit der Schleifbreite **<**`S_W`**>** als Verfahrweg auf den Anfangspunkt und dem Vorschub für Querzustellung `<S_K>`.

⑧ Abbruchsignal: Die Bearbeitung endet nach Erreichen des nächsten Startpunkts.

⑨ Ohne Abbruchsignal: Der Ablauf wird so oft wiederholt, bis die programmierte Anzahl Wiederholungen `<S_H>` erreicht ist.

Kennzeichnet die sich wiederholenden Ablaufschritte.

**Hinweis**

Der Ablauf ist mit Einzelsatz nicht unterbrechbar.

## Ressourcen

Als Ressourcen verwendet der Zyklus eine satzübergreifende Synchronaktion und eine Synchronaktionsvariable. Die Synchronaktion wird dynamisch aus dem freien Bereich des Synchronaktionsbandes ermittelt (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR – 1199 ...). Als Synchronaktionsvariable wird SYG_IS[1] verwendet.

## Beispiele

### Beispiel 1: Oszillieren mit zwei Hüben

Zyklenparameter:

- Zustelltiefe am Anfang: 0,02 mm
- Zustelltiefe am Ende: 0,01 mm
- Hub: 100 mm
- Ausfeuerzeit: 1 s
- Vorschub Zustellung: 1 mm/min
- Quervorschub: 1000 mm/min
- Wiederholungen: 2
- Pendel- und Zustellachse: Standardgeometrieachsen

Abbruchsignal: schneller Eingang 1 ($A_IN[1] )

**Programmcode**

```
N10 T1 D1
N20 CYCLE4072("1",0.02,0.01,100,1,1,1000,2)
N30 M30
```

### Beispiel 2: Oszillieren mit zwei Hüben

Zyklenparameter:

- Zustelltiefe am Anfang: 0,02 mm
- Zustelltiefe am Ende: 0,01 mm
- Hub: 100 mm
- Ausfeuerzeit: 1 s
- Vorschub Zustellung: 1 mm/min
- Quervorschub: 1000 mm/min
- Wiederholungen: 2
- Pendel- und Zustellachse: Standardgeometrieachsen

Abbruchsignal: Variable $A_DBR[20] < 0,01

**Programmcode**

```
N10 T1 D1
N20 CYCLE4072("($A_DBR[20]<0.01)",0.02,0.01,100,1,1,1000,2)
N30 M30
```

### 3.25.1.47 CYCLE4073 - Längschleifen mit kontinuierlicher Zustellung

## Syntax

```
CYCLE4073(<S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>, <S_A1>, <S_A2>)
```

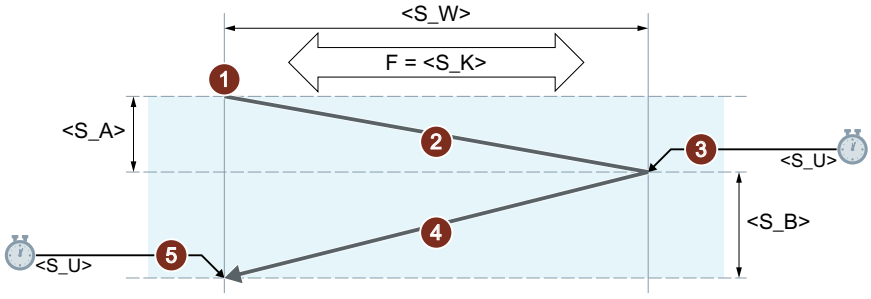
## Parameter

| Nr. | Parameter | Datentyp | Bedeutung |
|---|---|---|---|
| 1 | <S_A> | REAL | Zustelltiefe am Anfang |
| 2 | <S_B> | REAL | Zustelltiefe am Ende |
| 3 | <S_W> | REAL | Schleifbreite |
| 4 | <S_U> | REAL | Ausfeuerzeit |
| 5 | <S_K> | REAL | Vorschub für Querzustellung |
| 6 | <S_H> | INT | Anzahl Wiederholungen |
| 7 | <S_A1> | AXIS | Zustellachse (optional) bzw. 1. Geometrieachse |
| 8 | <S_A2> | AXIS | Pendelachse (optional) bzw. 2. Geometrieachse |

## Funktion

Der Zyklus dient zum Abarbeiten von sich wiederholenden Zustellungen. Dabei kann die Zustellung vom Anfang zum Ende und vom Ende zum Anfang unterschiedlich sein.

## Ablauf




① Start des Zyklus an der aktuellen Position der Pendelachse mit Zustelltiefe 0.

② Verfahren der Pendelachse mit der Schleifbreite <S_W> als Verfahrweg und Vorschub für Querzustellung <S_K> mit kontinuierlicher Zunahme der Zustelltiefe bis zur Zustelltiefe am Anfang <S_A>.

③ Ausfeuern mit der Ausfeuerzeit <S_U>.

④ Verfahren der Pendelachse mit der Schleifbreite <S_W> als Verfahrweg auf den Anfangspunkt und Vorschub für Querzustellung <S_K> mit kontinuierlicher Zunahme der Zustelltiefe bis zur Zustelltiefe am Ende <S_B>.

⑤ Ausfeuern mit der Ausfeuerzeit <S_U>.

Kennzeichnet die sich wiederholenden Ablaufschritte.

Der Ablauf wird so oft wiederholt, bis die programmierte Anzahl Wiederholungen <S_H> erreicht ist.

**Hinweis**

Der Ablauf ist mit Einzelsatz nicht unterbrechbar.

## Beispiel

**Oszillieren mit zwei Hüben**

Zyklenparameter:

- Zustelltiefe am Anfang: 0,02 mm
- Zustelltiefe am Ende: 0,01 mm
- Hub: 100 mm
- Ausfeuerzeit: 1 s
- Quervorschub: 1000 mm/min
- Wiederholungen: 2
- Pendel- und Zustellachse: Standardgeometrieachsen

**Programmcode**

```
N10 T1 D1
N20 CYCLE4073(0.02,0.01,100,1,1000,2)
N30 M30
```

### 3.25.1.48 CYCLE4074 - Längschleifen mit kontinuierlicher Zustellung und Abbruchsignal

## Syntax

```
CYCLE4074(<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>,
<S_A1>, <S_A2>)
```

## Parameter

| Nr. | Parameter | Datentyp | Bedeutung |
|---|---|---|---|
| 1 | <S_GAUGE> | STRING | Abbruchbedingungen für Zustellung:<br>1. Nummer eines schnellen Eingangs<br>2. Logischer Ausdruck |
| 2 | <S_A> | REAL | Zustelltiefe am Anfang |
| 3 | <S_B> | REAL | Zustelltiefe am Ende |
| 4 | <S_W> | REAL | Schleifbreite |
| 5 | <S_U> | REAL | Ausfeuerzeit |
| 6 | <S_K> | REAL | Vorschub für Querzustellung |
| 7 | <S_H> | INT | Anzahl Wiederholungen |

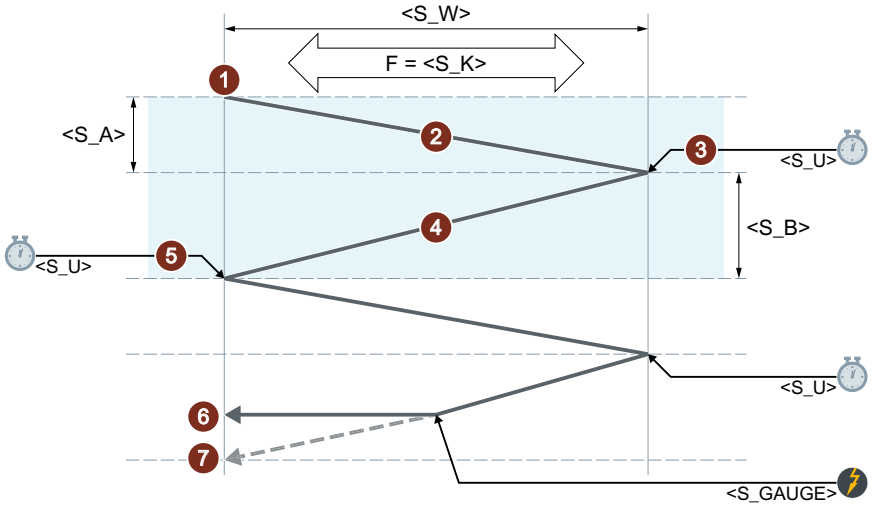
| Nr. | Parameter | Datentyp | Bedeutung |
|---|---|---|---|
| 8 | <S_A1> | AXIS | Zustellachse (optional) bzw. 1. Geometrieachse |
| 9 | <S_A2> | AXIS | Pendelachse (optional) bzw. 2. Geometrieachse |

## Funktion

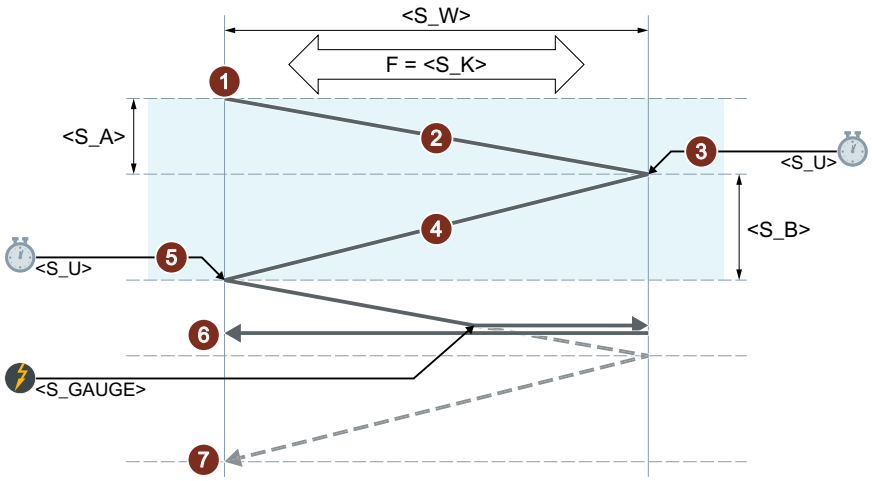
Der Zyklus dient zum Abarbeiten von sich wiederholenden Zustellungen unter Berücksichtigung z. B. eines externen Abbruchsignals. Die Zustelltiefe kann am Anfang und am Ende unterschiedlich sein. Die Tiefenzustellung wird abgebrochen, wenn die Abbruchbedingung erfüllt ist. Nach dem Abbruch der Tiefenzustellung wird immer ein kompletter Hub durchgeführt.

## Ablauf

### Abbruch der Zustellung vom Ende zum Anfang



### Abbruch der Zustellung vom Anfang zum Ende




① Start des Zyklus an der aktuellen Position der Pendelachse mit Zustelltiefe 0.

② Verfahren der Pendelachse mit der Schleifbreite <S_W> als Verfahrweg und Vorschub für Querzustellung <S_K> mit kontinuierlicher Zunahme der Zustelltiefe bis zur Zustelltiefe am Anfang <S_A>.

③ Ausfeuern mit der Ausfeuerzeit <S_U>.

④ Verfahren der Pendelachse mit der Schleifbreite <S_W> als Verfahrweg auf den Anfangspunkt und Vorschub für Querzustellung <S_K> mit kontinuierlicher Zunahme der Zustelltiefe bis zur Zustelltiefe am Ende <S_B>.

⑤ Ausfeuern mit der Ausfeuerzeit <S_U>.

⑥ Abbruchsignal: Die Tiefenzustellung wird abgebrochen. Die Bearbeitung endet nach Erreichen des nächsten Startpunkts.

⑦ Ohne Abbruchsignal: Der Ablauf wird so oft wiederholt, bis die programmierte Anzahl Wiederholungen <S_H> erreicht ist.

Kennzeichnet die sich wiederholenden Ablaufschritte.

**Hinweis**

Der Ablauf ist mit Einzelsatz nicht unterbrechbar.

## Ressourcen

Als Ressourcen verwendet der Zyklus eine satzübergreifende Synchronaktion und eine Synchronaktionsvariable. Die Synchronaktion wird dynamisch aus dem freien Bereich des Synchronaktionsbandes ermittelt (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR – 1199 ...). Als Synchronaktionsvariable wird SYG_IS[1] verwendet.

## Beispiele

**Beispiel 1: Oszillieren mit zwei Hüben**

Zyklenparameter:

- Zustelltiefe am Anfang: 0,02 mm
- Zustelltiefe am Ende: 0,01 mm
- Hub: 100 mm
- Ausfeuerzeit: 1 s
- Quervorschub: 1000 mm/min
- Wiederholungen: 2
- Pendel- und Zustellachse: Standardgeometrieachsen

Abbruchsignal: schneller Eingang 1 ($A_IN[1] )

**Programmcode**

```
N10 T1 D1
N20 CYCLE4074("1",0.02,0.01,100,1,1000,2)
N30 M30
```

**Beispiel 2: Oszillieren mit zwei Hüben**

Zyklenparameter:

- Zustelltiefe am Anfang: 0,02 mm
- Zustelltiefe am Ende: 0,01 mm
- Hub: 100 mm
- Ausfeuerzeit: 1 s
- Quervorschub: 1000 mm/min
- Wiederholungen: 2
- Pendel- und Zustellachse: Standardgeometrieachsen

Abbruchsignal: Variable $A_DBR[20] < 0,01

**Programmcode**

```
N10 T1 D1
N20 CYCLE4074("($A_DBR[20]<0.01)",0.02,0.01,100,1,1000,2)
N30 M30
```

### 3.25.1.49 CYCLE4075 - Flachschleifen mit Zustellung am Umkehrpunkt

## Syntax

```
CYCLE4075(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>, <S_A1>,
<S_A2>)
```

## Parameter

| Nr. | Parameter | Datentyp | Bedeutung |
|---|---|---|---|
| 1 | <S_I> | REAL | Zustelltiefe am Anfang |
| 2 | <S_J> | REAL | Zustelltiefe am Ende |
| 3 | <S_K> | REAL | Gesamtzustelltiefe |
| 4 | <S_A> | REAL | Schleifbreite |
| 5 | <S_R> | REAL | Vorschub für Zustellung |
| 6 | <S_F> | REAL | Vorschub für Querzustellung |
| 7 | <S_P> | REAL | Ausfeuerzeit |
| 8 | <S_A1> | AXIS | Zustellachse (optional) |
| 9 | <S_A2> | AXIS | Pendelachse (optional) |

## Funktion

Der Zyklus dient zum Bearbeiten mit einer Gesamtzustelltiefe in Zustellschritten. Die Zustelltiefen am Anfang und am Ende können unterschiedlich sein. Zwischen dem Zustellen erfolgt eine Tangentialbewegung.

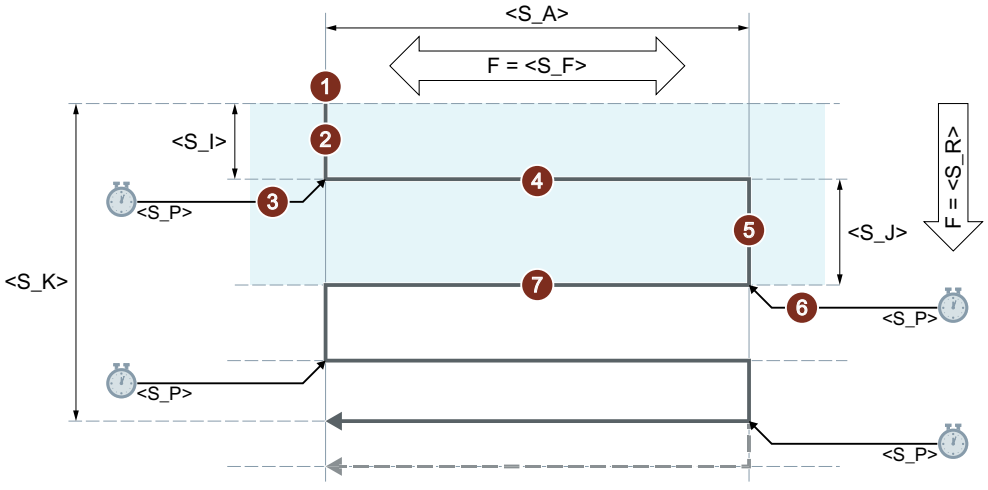
Die Wegangaben P1 bis P4 können negativ oder positiv sein.

Die Angabe der Zustellachse und/oder Pendelachse sind optional. Wird ein bzw. beide Parameter nicht angegeben, verwendet der Zyklus die ersten beiden Geometrieachsen des Kanals.

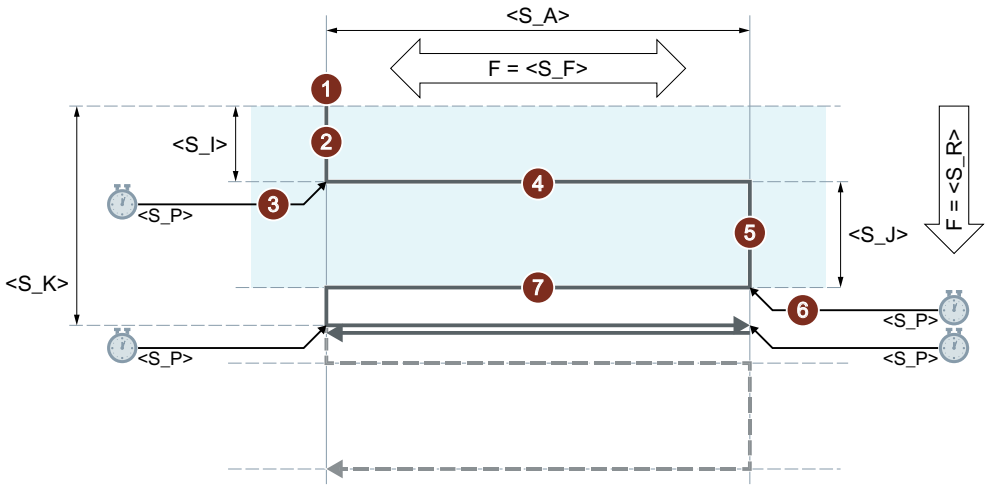
Ist die Summe aus Zustelltiefe am Anfang und am Ende gleich 0 bzw. die Gesamtzustelltiefe gleich 0, wird nur ein Ausfeuerhub durchgeführt.

## Ablauf

**Gesamtzustelltiefe erreicht bei Zustellung am zweiten Umkehrpunkt**



### Gesamtzustelltiefe erreicht bei Zustellung am ersten Umkehrpunkt




① Start des Zyklus an der aktuellen Position der Pendelachse.

② Verfahren der Zustellachse auf die Zustelltiefe am Anfang `<S_I>` mit dem Vorschub für Zustellung `<S_R>`.

③ Ausfeuern mit der Ausfeuerzeit `<S_P>`.

④ Verfahren der Pendelachse mit der Schleifbreite `<S_A>` als Verfahrweg und dem Vorschub für Querzustellung `<S_F>`.

⑤ Verfahren der Zustellachse auf die Zustelltiefe am Ende `<S_J>` mit dem Vorschub für Zustellung `<S_R>`.

⑥ Ausfeuern mit der Ausfeuerzeit `<S_P>`.

⑦ Verfahren der Pendelachse mit der Schleifbreite `<S_A>` als Verfahrweg auf den Anfangspunkt und dem Vorschub für Querzustellung `<S_F>`.

Kennzeichnet die sich wiederholenden Ablaufschritte.

Der Ablauf wird so oft wiederholt, bis die Gesamtzustelltiefe `<S_K>` erreicht wurde. Der letzte Hub wird dann ungleichmäßig aufgeteilt.

**Hinweis**

Der Ablauf ist mit Einzelsatz nicht unterbrechbar.

## Beispiel

Oszillieren mit:

- 0,02 mm Zustelltiefe am Anfang
- 0,01 mm Zustelltiefe am Ende
- Gesamtzustelltiefe 1 mm
- Hub 100 mm
- Vorschub Zustellung 1 mm/min
- Quervorschub 1000 mm/min

- Ausfeuerzeit 1 Sekunde
- Standardgeometrieachsen

**Programmcode**

```
N10 T1 D1
N20 CYCLE4075(0.02,0.01,1,100,1,1000,1)
N30 M30
```

### 3.25.1.50 CYCLE4077 - Flachschleifen mit Zustellung am Umkehrpunkt und Abbruchsignal

## Syntax

```
CYCLE4077(<S_GAUGE>, <S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>,
<S_P>, <S_A1>, <S_A2>)
```

## Parameter

| Nr. | Parameter | Datentyp | Bedeutung |
|---|---|---|---|
| 1 | <S_GAUGE> | STRING | Abbruchbedingung für Zustellung:<br>• Nummer eines schnellen Eingangs<br>• Logischer Ausdruck |
| 2 | <S_I> | REAL | Zustelltiefe am Anfang |
| 3 | <S_J> | REAL | Zustelltiefe am Ende |
| 4 | <S_K> | REAL | Gesamtzustelltiefe |
| 5 | <S_A> | REAL | Schleifbreite |
| 6 | <S_R> | REAL | Vorschub für Zustellung |
| 7 | <S_F> | REAL | Vorschub für Querzustellung |
| 8 | <S_P> | REAL | Ausfeuerzeit |
| 9 | <S_A1> | AXIS | Zustellachse (optional) |
| 10 | <S_A2> | AXIS | Pendelachse (optional) |

## Funktion

Der Zyklus dient zum Bearbeiten mit einer Gesamtzustelltiefe in Zustellschritten. Die Zustelltiefen am Anfang und am Ende können unterschiedlich sein. Zwischen dem Zustellen erfolgt eine Tangentialbewegung. Die Tiefenzustellung wird abgebrochen, wenn das Abbruchsignal des schnellen Eingangs gleich 1 oder die Abbruchbedingung erfüllt ist. Nach dem Abbruch wird ein kompletter Hub ausgeführt.

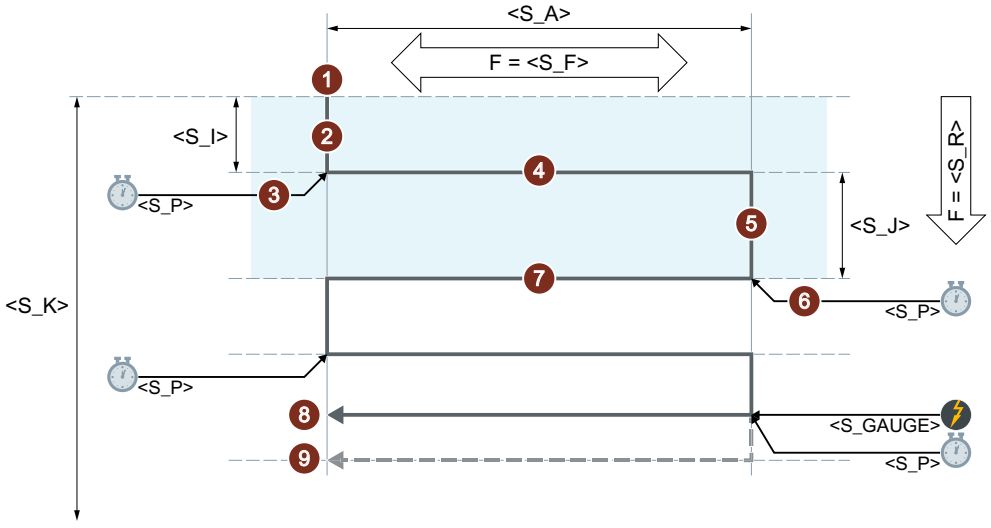
Die Wegangaben P2 bis P5 können negativ oder positiv sein.

Die Angabe der Zustellachse und/oder Pendelachse sind optional. Wird ein bzw. beide Parameter nicht angegeben, verwendet der Zyklus die ersten beiden Geometrieachsen des Kanals.

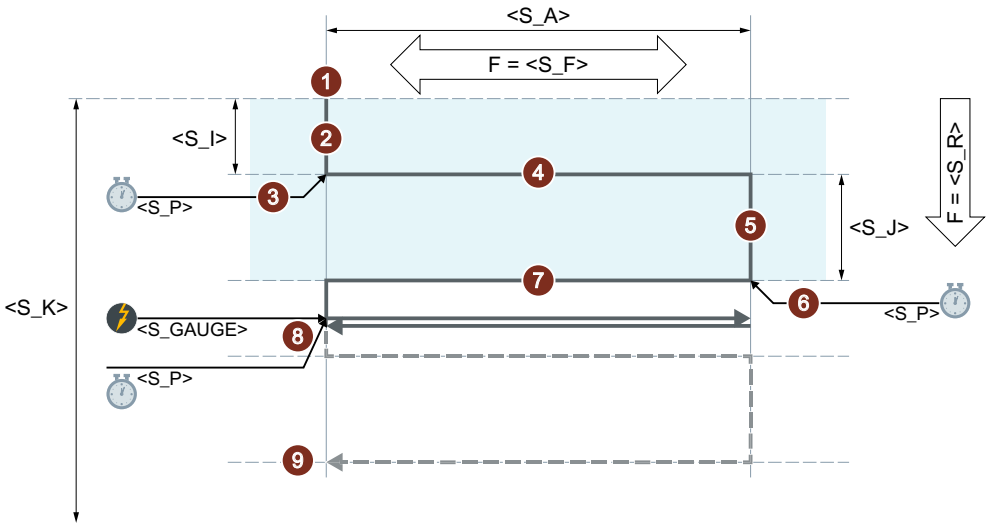
Ist die Summe aus Zustelltiefe am Anfang und am Ende gleich 0 bzw. die Gesamtzustelltiefe gleich 0, wird nur ein Ausfeuerhub durchgeführt.

## Ablauf

### Abbruch der Zustellung am Ende



### Abbruch der Zustellung am Anfang




① Start des Zyklus an der aktuellen Position der Pendelachse.

② Verfahren der Zustellachse auf die Zustelltiefe am Anfang `<S_I>` mit dem Vorschub für Zustellung `<S_R>`.

③ Ausfeuern mit der Ausfeuerzeit `<S_P>`.

④ Verfahren der Pendelachse mit der Schleifbreite `<S_A>` als Verfahrweg und dem Vorschub für Querzustellung `<S_F>`.

⑤ Verfahren der Zustellachse auf die Zustelltiefe am Ende `<S_J>` mit dem Vorschub für Zustellung `<S_R>`.

⑥ Ausfeuern mit der Ausfeuerzeit `<S_P>`.

⑦ Verfahren der Pendelachse mit der Schleifbreite `<S_A>` als Verfahrweg auf den Anfangspunkt und dem Vorschub für Querzustellung `<S_F>`.

⑧ Abbruchsignal: Die Bearbeitung endet nach Erreichen des nächsten Startpunkts.

⑨ Ohne Abbruchsignal: Der Ablauf wird so oft wiederholt, bis die Gesamtzustelltiefe `<S_K>` erreicht wurde. Der letzte Hub wird dann ungleichmäßig aufgeteilt.

Kennzeichnet die sich wiederholenden Ablaufschritte.

**Hinweis**

Der Ablauf ist mit Einzelsatz nicht unterbrechbar.

## Ressourcen

Als Ressourcen verwendet der Zyklus eine satzübergreifende Synchronaktion und eine Synchronaktionsvariable. Die Synchronaktion wird dynamisch aus dem freien Bereich des Synchronaktionsbandes ermittelt (CUS.DIR - 1 ..., CMA.DIR - 1000 ... , CST.DIR – 1199 ...). Als Synchronaktionsvariable wird SYG_IS[1] verwendet.

## Beispiele

### Beispiel 1

Oszillieren mit:

- 0,02 mm Zustelltiefe am Anfang
- 0,01 mm Zustelltiefe am Ende
- Gesamtzustelltiefe 1 mm
- Hub 100 mm
- Vorschub Zustellung 1 mm/min
- Quervorschub 1000 mm/min
- Ausfeuerzeit 1 Sekunde
- Standardgeometrieachsen

Abbruchsignal: schneller Eingang 1 ($A_IN[1] )

**Programmcode**

```
N10 T1 D1
N20 CYCLE4077("1",0.02,0.01,1,100,1,1000,1)
N30 M30
```

### Beispiel 2

Oszillieren mit:

- 0,02 mm Zustelltiefe am Anfang
- 0,01 mm Zustelltiefe am Ende
- Gesamtzustelltiefe 1 mm
- Hub 100 mm
- Vorschub Zustellung 1 mm/min
- Quervorschub 1000 mm/min
- Ausfeuerzeit 1 Sekunde
- Standardgeometrieachsen

Abbruchsignal: Dualport-RAM-Variable 20 kleiner als 0,01 ($A_DBR[20] < 0,01)

**Programmcode**

```
N10 T1 D1
N20 CYCLE4077("($A_DBR[20]<0.01)",0.02,0.01,1,100,1,1000,1)
N30 M30
```

### 3.25.1.51 CYCLE4078 - Flachschleifen mit kontinuierlicher Zustellung

## Syntax

```
CYCLE4078(<S_I>, <S_J>, <S_K>, <S_A>, <S_F>, <S_P>, <S_A1>, <S_A2>)
```

## Parameter

| Nr. | Parameter | Datentyp | Bedeutung |
|---|---|---|---|
| 1 | <S_I> | REAL | Zustelltiefe vom Anfang zum Ende |
| 2 | <S_J> | REAL | Zustelltiefe vom Ende zum Anfang |
| 3 | <S_K> | REAL | Gesamtzustelltiefe |
| 4 | <S_A> | REAL | Schleifbreite |
| 5 | <S_F> | REAL | Vorschub |
| 6 | <S_P> | REAL | Ausfeuerzeit |
| 7 | <S_A1> | AXIS | Zustellachse (optional) |
| 8 | <S_A2> | AXIS | Pendelachse (optional) |

## Funktion

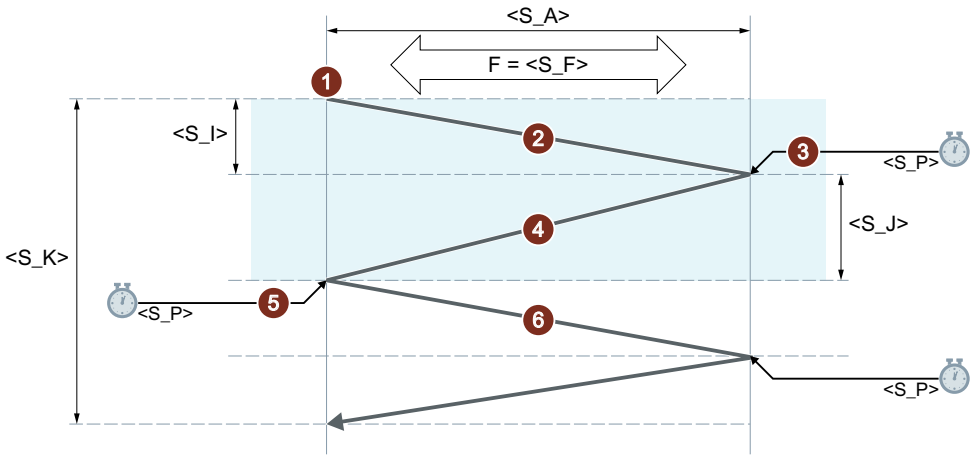
Der Zyklus dient zum Bearbeiten mit einer Gesamtzustelltiefe mittels kontinuierlicher Zustellung. Die Zustelltiefen vom Anfang zum Ende und vom Ende zum Anfang können unterschiedlich sein.

Die Wegangaben P1 bis P4 können negativ oder positiv sein.

Die Angabe der Zustellachse und/oder Pendelachse sind optional. Wird ein bzw. beide Parameter nicht angegeben, verwendet der Zyklus die ersten beiden Geometrieachsen des Kanals.

Ist die Summe der Zustelltiefen P1 und P2 gleich 0 bzw. die Gesamtzustelltiefe gleich 0 wird nur ein Ausfeuerhub durchgeführt.

## Ablauf




① Start des Zyklus an der aktuellen Position der Pendelachse mit Zustelltiefe 0.

② Verfahren der Pendelachse mit der Schleifbreite <S_A> als Verfahrweg und Vorschub <S_F> mit kontinuierlicher Zunahme der Zustelltiefe bis zur Zustelltiefe am Anfang **<S_I>**.

③ Ausfeuern mit der Ausfeuerzeit <S_P>.

④ Verfahren der Pendelachse mit der Schleifbreite <S_A> als Verfahrweg auf den Anfangspunkt und Vorschub <S_F> mit kontinuierlicher Zunahme der Zustelltiefe bis zur Zustelltiefe am Ende <S_J>.

⑤ Ausfeuern mit der Ausfeuerzeit <S_P>.

⑥ Verfahren der Pendelachse mit der Schleifbreite <S_A> als Verfahrweg auf den Anfangspunkt und Vorschub <S_F>.

Kennzeichnet die sich wiederholenden Ablaufschritte.

Der Ablauf wird so oft wiederholt, bis die Gesamtzustelltiefe <S_K> erreicht wurde. Der letzte Hub wird dann ungleichmäßig aufgeteilt.

**Hinweis**

Der Ablauf ist mit Einzelsatz nicht unterbrechbar.

## Beispiel

Oszillieren mit:

- 20 mm Zustelltiefe am Anfang
- 10 mm Zustelltiefe am Ende
- Gesamtzustelltiefe 100 mm
- Hub 100 mm
- Vorschub 1000 mm/min
- Ausfeuerzeit 1 Sekunde
- Standardgeometrieachsen

**Programmcode**

```
N10 T1 D1
N20 CYCLE4078(20,10,100,100,1000,1)
N30 M30
```

### 3.25.1.52 CYCLE4079 - Flachschleifen mit intermittierender Zustellung

## Syntax

```
CYCLE4079(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>, <S_A1>,
<S_A2>)
```

## Parameter

| Nr. | Parameter | Datentyp | Bedeutung |
|---|---|---|---|
| 1 | <S_I> | REAL | Zustelltiefe am Anfang |
| 2 | <S_J> | REAL | Zustelltiefe am Ende |
| 3 | <S_K> | REAL | Gesamtzustelltiefe |
| 4 | <S_A> | REAL | Schleifbreite |
| 5 | <S_R> | REAL | Vorschub für Zustellung |
| 6 | <S_F> | REAL | Vorschub für Querzustellung |
| 7 | <S_P> | REAL | Ausfeuerzeit |
| 8 | <S_A1> | AXIS | Zustellachse (optional) |
| 9 | <S_A2> | AXIS | Pendelachse (optional) |

## Funktion

Der Zyklus dient zum Bearbeiten mit einer Gesamtzustelltiefe in Zustellschritten. Die Zustelltiefen am Anfang und am Ende können unterschiedlich sein. Zwischen dem Zustellen erfolgt eine Tangentialbewegung.

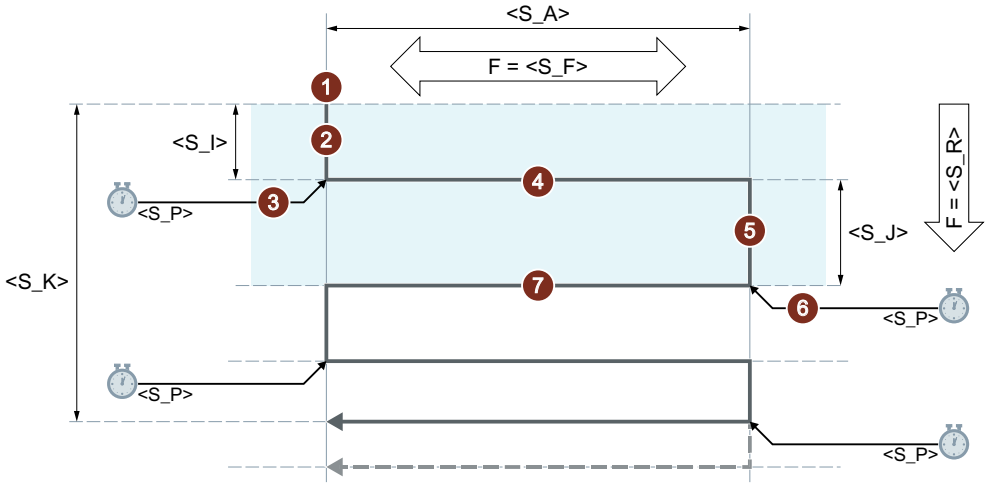
Die Wegangaben P1 bis P4 können negativ oder positiv sein.

Die Angabe der Zustellachse und/oder Pendelachse sind optional. Wird ein bzw. beide Parameter nicht angegeben, verwendet der Zyklus die ersten beiden Geometrieachsen des Kanals.

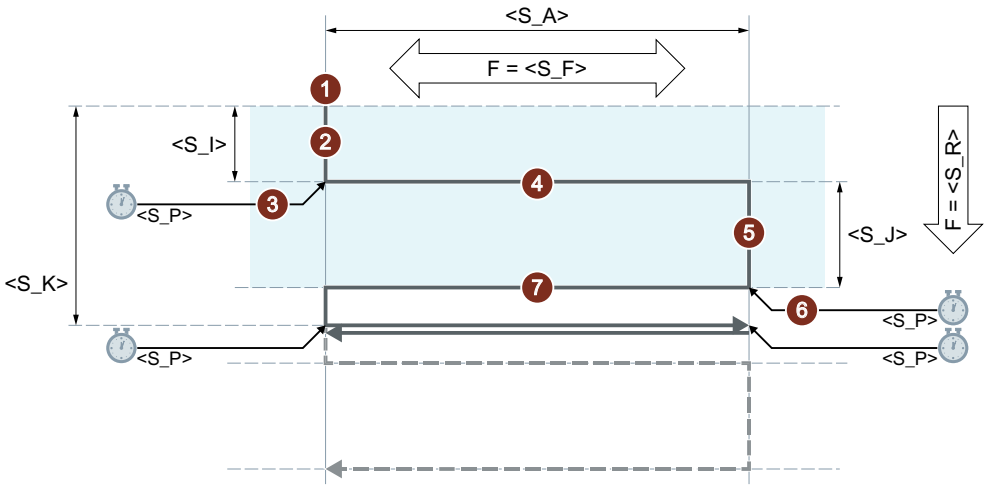
Ist die Summe aus Zustelltiefe am Anfang und am Ende gleich 0 bzw. die Gesamtzustelltiefe gleich 0, wird nur ein Ausfeuerhub durchgeführt.

## Ablauf

**Gesamtzustelltiefe erreicht bei Zustellung am zweiten Umkehrpunkt**



**Gesamtzustelltiefe erreicht bei Zustellung am ersten Umkehrpunkt**




① Start des Zyklus an der aktuellen Position der Pendelachse.

② Verfahren der Zustellachse auf die Zustelltiefe am Anfang `<S_I>` mit dem Vorschub für Zustellung `<S_R>`.

③ Ausfeuern mit der Ausfeuerzeit `<S_P>`.

④ Verfahren der Pendelachse mit der Schleifbreite `<S_A>` als Verfahrweg und dem Vorschub für Querzustellung `<S_F>`.

⑤ Verfahren der Zustellachse auf die Zustelltiefe am Ende `<S_J>` mit dem Vorschub für Zustellung `<S_R>`.

⑥ Ausfeuern mit der Ausfeuerzeit `<S_P>`.

⑦ Verfahren der Pendelachse mit der Schleifbreite `<S_A>` als Verfahrweg auf den Anfangspunkt und dem Vorschub für Querzustellung `<S_F>`.

Kennzeichnet die sich wiederholenden Ablaufschritte.

Der Ablauf wird so oft wiederholt, bis die Gesamtzustelltiefe `<S_K>` erreicht wurde. Der letzte Hub wird dann ungleichmäßig aufgeteilt.

**Hinweis**

Der Ablauf ist mit Einzelsatz nicht unterbrechbar.

## Beispiel

Oszillieren mit:

- 0,02 mm Zustelltiefe am Anfang
- 0,01 mm Zustelltiefe am Ende
- Gesamtzustelltiefe 1 mm
- Hub 100 mm
- Vorschub Zustellung 1 mm/min
- Quervorschub 1000 mm/min

- Ausfeuerzeit 1 Sekunde
- Standardgeometrieachsen

**Programmcode**

```
N10 T1 D1
N20 CYCLE4079(0.02,0.01,1,100,1,1000,1)
N30 M30
```

### 3.25.1.53 GROUP_BEGIN - Anfang Programmblock

#### Syntax

```
GROUP_BEGIN(<_LEVEL>, <_NAME>, <_SP>, <_MODE>, <S_ICON>)
```

#### Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | | |
|---|---|---|---|---|---|---|
| 1 | | <_LEVEL> | INT | Ebene | | |
| | | | | 0 = | Hauptebene | |
| | | | | 1 = | 1. Unterebene | |
| 2 | | <_NAME> | STRING[128] | Blockname | | |
| 3 | | <_SP> | INT | Spindel | | |
| | | | | 0 = | keine Spindel | |
| | | | | 1 = | Hauptspindel | |
| | | | | 2 = | Gegenspindel | |
| 4 | | <_MODE> | INT | Modus | | |
| | | | | Bit 0 | = 1 | GROUP_ADDEND existiert |
| | | | | Bit 1 | = 1 | ShopTurn: automatischer Rückzug (auf Werkzeugwechselpunkt fahren) |
| | | | | Bit 12 | reserviert | |
| | | | | Bit 13 | reserviert | |
| 5 | | <S_ICON> | STRING[32] | Name des Icons (nur für Bedienoberfläche) | | |

### 3.25.1.54 GROUP_END - Ende Programmblock

#### Syntax

```
GROUP_END(<_LEVEL>, <_SP>)
```

### Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | |
|---|---|---|---|---|---|
| 1 | | <_LEVEL> | INT | Ebene | |
| | | | | 0 = | Hauptebene |
| | | | | 1 = | 1. Unterebene |
| 2 | | <_SP> | INT | Spindel | |
| | | | | 0 = | keine Spindel |
| | | | | 1 = | Hauptspindel |
| | | | | 2 = | Gegenspindel |

## 3.25.1.55 GROUP_ADDEND - Ende Einfahrzusatz

### Syntax

```
GROUP_ADDEND(<_LEVEL>, <_SP>)
```

### Parameter

| Nr. | Parameter Maske | Parameter intern | Datentyp | Bedeutung | |
|---|---|---|---|---|---|
| 1 | | <_LEVEL> | INT | Ebene | |
| | | | | 0 = | Hauptebene |
| | | | | 1 = | 1. Unterebene |
| 2 | | <_SP> | INT | Spindel | |
| | | | | 0 = | keine Spindel |
| | | | | 1 = | Hauptspindel |
| | | | | 2 = | Gegenspindel |

## 3.25.1.56 Randbedingungen

### Technologieskalierung innerhalb von Zyklenmasken

Ist die Technologieskalierung aktiv, kann bei verschiedenen Zyklenmasken die vereinfachte Eingabe angewählt werden, bei der nur die wesentlichsten Zyklenparameter angezeigt werden.

Bei folgenden Zyklenmasken kann z. B. die vereinfachte Eingabe angewählt werden:

| Technologie | Zyklenmaske |
|---|---|
| Bohren | Tieflochbohren |
| | Gewindebohren |
| Fräsen | Rechtecktasche |
| | Konturfräsen: Tasche |

| Technologie | Zyklenmaske |
|---|---|
| Drehen | Gewindedrehen: Längs |
| | Konturdrehen: Abspanen |
| | Konturdrehen: Stechen |
| | Konturdrehen: Stechdrehen |

Bei den betreffenden Zyklenmasken steht an der Bedienoberfläche die Auswahlmöglichkeit "Eingabe: **einfach**" und "Eingabe: **komplett**" zur Verfügung.

#### Nicht angezeigte Zyklenparameter

Die bei der vereinfachten Eingabe nicht angezeigten Zyklenparameter werden entweder mit festen, technologisch sinnvollen, aber nicht änderbaren Werten vorbelegt. Oder den Zyklenparametern werden über kanalspezifische Zyklen-Settingdaten parametrierbare Werte zugewiesen. Siehe unten Absatz "Inbetriebnahme" > "Kanalspezifische Zyklen-Settingdaten"

#### Umschaltung "Eingabe: komplett" > "Eingabe: einfach"

Wird eine Zyklenmaske unter der Einstellung "Eingabe komplett" ausgefüllt und anschließend auf "Eingabe einfach" umgeschaltet, werden bei der Generierung des Zyklenaufrufs für die dann nicht mehr angezeigten Parameter, die Default- oder Settingdatenwerte verwendet.

### Inbetriebnahme

#### Kanalspezifische Konfigurationsmaschinendaten

Mit dem Maschinendatum wird die Möglichkeit zur Technologieskalierung innerhalb von Zyklenmasken aktiviert:

MD52210 $MCS_FUNCTION_MASK_DISP, Bit 9 = 1 (Auswahl "Eingabe einfach" anzeigen)

#### Kanalspezifische Zyklen-Settingdaten

Ist die vereinfachte Eingabe innerhalb von Zyklenmasken aktiv, können die Werte für bestimmte Zyklenparameter über folgende Settingdaten vorgegeben werden:

| Nummer | Bezeichner | Bedeutung |
|---|---|---|
| SD55300 | $SCS_EASY_SAFETY_CLEARANCE | Sicherheitsabstand |
| SD55301 | $SCS_EASY_DWELL_TIME | Verweilzeit |
| SD55305 | $SCS_EASY_DRILL_DEEP_FD1 | Tieflochbohren: Prozentsatz: 1. Vorschub |
| SD55306 | $SCS_EASY_DRILL_DEEP_DF | Tieflochbohren: Prozentsatz: Zustellung |
| SD55307 | $SCS_EASY_DRILL_DEEP_V1 | Tieflochbohren: minimale Tiefenzustellung |
| SD55308 | $SCS_EASY_DRILL_DEEP_V2 | Tieflochbohren: Rückzugsbetrag |
| SD55309 | $SCS_EASY_THREAD_RETURN_DIST | Gewindedrehen: Rücklaufabstand |

## 3.25.2 Messzyklen

Messzyklen sind spezielle von Siemens bereitgestellte Unterprogramme zur Lösung bestimmter Messaufgaben. Wie bei Zyklen allgemein üblich, können auch die Messzyklen über Parameter an das konkrete Problem angepasst werden.

Messzyklen stehen für Messungen in folgenden Bereichen und Technologien zur Verfügung:

- Werkzeugmessungen Drehen / Fräsen
- Werkstückmessungen Drehen / Fräsen

### Weitere Informationen

Programmierhandbuch Messzyklen

## 3.25.3 Übersicht Messzyklenparameter

### 3.25.3.1 Messzyklenparameter CYCLE973

```
PROC CYCLE973(INT S_MVAR,INT S_PRNUM,INT S_CALNUM,REAL S_SETV,INT S_MA,INT S_MD,REAL
S_FA,REAL S_TSA,REAL S_VMS,INT S_NMSP,INT S_MCBIT,INT _DMODE,INT _AMODE)
```

Tabelle 3-8 Aufrufparameter CYCLE973 [1)]

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 1 | | S_MVAR | Messvariante (default=0012103) | |
| | | | Werte: | EINER: Kalibrieren an einer Fläche, Kante oder in einer Nut<br>0 = Länge an Fläche/Kante (im WKS) mit bekanntem Sollwert<br>1 = Radius an Fläche (im WKS) mit bekanntem Sollwert<br>2 = Länge in Nut (im MKS), siehe S_CALNUM<br>3 = Radius in Nut (im MKS), siehe S_CALNUM |
| | | | | ZEHNER: reserviert<br>0 = 0 |
| | | | | HUNDERTER: reserviert<br>0 = 0 |
| | | | | TAUSENDER: Auswahl Messachse und Messrichtung bei Kalibrieren [2)]<br>0 = keine Angabe (bei Kalibrieren Fläche am Nutboden keine Auswahl der Messachse und Messrichtung) [4)]<br>1 = Auswahl Messachse und Messrichtung angeben, siehe S_MA, S_MD (eine Messrichtung in einer Messachse)<br>2 = Auswahl Messachse angeben, siehe S_MA (zwei Messrichtungen in einer Messachse) |
| | | | | ZEHNTAUSENDER: Ermittlung Lageabweichung (Schieflage Messtaster) [2), 3)]<br>0 = Lageabweichung ermitteln<br>1 = keine Lageabweichung ermitteln |
| | | | | HUNDERTTAUSENDER: reserviert<br>0 = 0 |
| | | | | EINEMILLION:Werkzeuglänge anpassen [7)]<br>0 = Werkzeuglänge nicht anpassen (nur Triggerpunkte)<br>1 = Werkzeuglänge anpassen |
| 2 | icon+ Anzahl | S_PRNUM | Nummer des Feldes der Messtasterparameter (nicht Messtasternummer) (default=1) | |
| 3 | | S_CALNUM | Nummer der Kalibriernut bei Kalibrieren an einer Nut (default=1) [5)] | |
| 4 | | S_SETV | Sollwert bei Kalibrieren an einer Fläche | |
| 5 | X0 | S_MA | Messachse (Nummer der Achse) [6)] (default=1) | |
| | | | Werte: | 1 = 1. Achse der Ebene (bei G18 Z)<br>2 = 2. Achse der Ebene (bei G18 X)<br>3 = 3. Achse der Ebene (bei G18 Y) [6)] |
| 6 | +- | S_MD | Messrichtung (default=1) | |
| | | | Werte: | 0 = positive Messrichtung<br>1 = negative Messrichtung |
| 7 | DFA | S_FA | Messweg | |
| 8 | TSA | S_TSA | Vertrauensbereich | |
| 9 | VMS | S_VMS | Variable Messgeschwindigkeit bei Kalibrieren [2)] | |
| 10 | Messungen | S_NMSP | Anzahl Messungen am selben Ort [2)] (default=1) | |
| 11 | | S_MCBIT | reserviert | |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 12 | | _DMODE | Displaymode | |
| | | | Werte: | EINER: Bearbeitungsebene G17/G18/G19<br>0 = Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv<br>1 = G17 (nur im Zyklus aktiv)<br>2 = G18 (nur im Zyklus aktiv)<br>3 = G19 (nur im Zyklus aktiv) |
| 13 | | _AMODE | Alternativmode | |

1) alle Defaultwerte = 0 oder als default=x gekennzeichnet

2) Anzeige abhängig vom allgemeinen SD54760 $SNS_MEA_FUNCTION_MASK_PIECE

3) nur relevant bei Kalibrieren in zwei Achsrichtungen

4) nur Messachse und Messrichtung wird automatisch aus der Schneidenlage (SL) des Messtasters ermittelt. SL=8 → -X , SL=7 → -Z

5) Die Nummer der Kalibriernut (n) verweist auf folgende allgemeine Settingdaten (alle Positionen in MKS):
bei Schneidenlage SL=7:
SD54615 $SNS_MEA_CAL_EDGE_BASE_AX1[n] Position des Bodens der Nut in der 1. Achse der Ebene (bei G18 Z)
SD54621 $SNS_MEA_CAL_EDGE_PLUS_DIR_AX2[n] Position der Nutwand in Plusrichtung der 2. Achse der Ebene (bei G18 X)
SD54622 $SNS_MEA_CAL_EDGE_MINUS_DIR_AX2[n] Position der Nutwand in Minusrichtung der 2. Achse der Ebene
bei Schneidenlage SL=8:
SD54619 $SNS_MEA_CAL_EDGE_BASE_AX2[n] Position des Bodens der Nut in der 2. Achse der Ebene
SD54620 $SNS_MEA_CAL_EDGE_UPPER_AX2[n] Position der Oberkante der Nut in der 2. Achse der Ebene, (nur zum Vorpositionieren des Messtasters)
SD54617 $SNS_MEA_CAL_EDGE_PLUS_DIR_AX1[n] Position der Nutwand in Plusrichtung der 1. Achse der Ebene
SD54618 $SNS_MEA_CAL_EDGE_MINUS_DIR_AX1[n] Position der Nutwand in Minusrichtung der 1. Achse der Ebene
**Hinweis:**
Die Positionswerte für die Nutwand +- können grob bestimmt sein.
Die Nutbreite aus der Differenz der Positionswerte der Nutwand muss exakt (Feinmessuhr) bestimmt sein.
Bei Kalibrieren in der Nut wird davon ausgegangen, dass die Werkzeuglänge des Messtasters der kalibrierten Achse = 0 ist.
Die Positionswerte für den Nutboden müssen ebenfalls exakt an der Maschine bestimmt werden (keine Zeichnungsmaße).

6) Messachse `S_MA=3` bei Kalibrieren an einer Fläche und an einer Drehmaschine mit realer 3. Achse der Ebene (bei G18 Y).

7) Werkzeuglänge anpassen bei Abgleich Länge in Nut oder bei Längen an Fläche.
Werkstückmesstaster in Drehmaschinen kann mit 2 Längen (X Z) beschrieben sein.
Messtaster Drehen Typ 580
Schneidenlage 7: Bei Abgleich Länge wird wahlweise die Z-Länge korrigiert.
Messtaster Drehen Typ 580
Schneidenlage 8: Bei Abgleich Länge wird wahlweise die X-Länge korrigiert
Bei den Messvarianten Radius an Nut oder Radius an Fläche erfolgt keine Anpassung der Werkzeuglänge.
Es werden immer nur die entsprechenden Triggerpunkte abgespeichert.

### 3.25.3.2 Messzyklenparameter CYCLE974

```
PROC CYCLE974(INT S_MVAR,INT S_KNUM,INT S_KNUM1,INT S_PRNUM,REAL S_SETV,INT S_MA,REAL
S_FA,REAL S_TSA,REAL S_STA1,INT S_NMSP,STRING[32] S_TNAME,INT S_DLNUM,REAL S_TZL,REAL
S_TDIF,REAL S_TUL,REAL S_TLL,REAL S_TMV,INT S_K,INT S_EVNUM,INT S_MCBIT,INT _DMODE,INT
_AMODE,INT _DP)
```

Tabelle 3-9 Aufrufparameter CYCLE974 [1)]

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 1 | | S_MVAR | Messvariante | |
| | | | Werte: | EINER:<br>0 = Stirnfläche messen<br>1 = Innenmessung<br>2 = Außenmessung |
| | | | | ZEHNER: reserviert |
| | | | | HUNDERTER: Korrekturziel<br>0 = nur Messen (keine Korrektur der NV bzw. keine Werkzeugkorrektur)<br>1 = Messen, Ermittlung und Korrektur der NV (siehe S_KNUM) [3)]<br>2 = Messen und Werkzeugkorrektur (siehe S_KNUM1) |
| | | | | TAUSENDER: reserviert |
| | | | | ZEHNTAUSENDER: Messen mit oder ohne Umschlag der Hauptspindel (Drehspindel)<br>0 = Messen ohne Umschlag<br>1 = Messen mit Umschlag |
| 2 | Auswahl | S_KNUM | Korrektur in Nullpunktverschiebung (NV) oder Basis NV oder Basisbezug [2)] | |
| | | | Werte: | EINER: |
| | | | | ZEHNER:<br>0 = keine Korrektur<br>1 bis max. 99 Nummer der Nullpunktverschiebung oder<br>1 bis max. 16 Nummer der Basisverschiebung |
| | | | | HUNDERTER: reserviert |
| | | | | TAUSENDER: Korrektur in NV oder Basis NV oder Basisbezug<br>0 = Korrektur einstellbare NV<br>1 = Korrektur kanalspezifische Basis NV<br>2 = Korrektur Basisbezug<br>3 = Korrektur globale Basis NV<br>9 = Korrektur aktive NV bzw. bei G500 letzte aktive kanalspezifische Basis NV |
| | | | | ZEHNTAUSENDER: Korrektur grob oder fein in NV, Basis NV oder Basisbezug<br>0 = Korrektur fein [6)]<br>1 = Korrektur grob |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 3 | Auswahl | S_KNUM1 | Korrektur in Werkzeugkorrektur [2), 4)] | |
| | | | Werte: | EINER: |
| | | | | ZEHNER: |
| | | | | HUNDERTER:<br>0 = keine Korrektur<br>1 bis max. 999 D-Nummer (Schneidennummer) bei Werkzeugkorrektur;<br>bei Summen- und Einrichtekorrektur siehe auch S_DLNUM |
| | | | | TAUSENDER: 0 oder eindeutige D-Nummer |
| | | | | ZEHNTAUSENDER: 0 oder eindeutige D-Nummer<br>1 bis max. 32000 wenn eindeutige D-Nummern in MD eingerichtet sind |
| | | | | HUNDERTTAUSENDER: Werkzeugkorrektur [2)]<br>0 = keine Angabe (Korrektur in Werkzeuggeometrie)<br>1 = Korrektur Länge L1<br>2 = Korrektur Länge L2<br>3 = Korrektur Länge L3<br>4 = Korrektur Radius |
| | | | | EINEMILLION: Werkzeugkorrektur [2)]<br>0 = keine Angabe (Korrektur des Werkzeuglängen-Verschleiß)<br>1 = Werkzeugkorrektur Summenkorrektur (SK) [5)]<br>Werkzeugkorrekturwert wird zur bestehenden SK addiert<br>2 = Werkzeugkorrektur Einrichtekorrektur (EK) [5)]<br>EK (neu) = EK (alt) + SK (alt) Korrekturwert, SK (neu) = 0<br>3 = Werkzeugkorrektur Einrichtekorrektur (EK) [5)]<br>Werkzeugkorrekturwert wird zur bestehenden EK addiert<br>4 = Werkzeugkorrektur Geometrie |
| | | | | ZEHNMILLION: Werkzeugkorrektur [2)]<br>0 = keine Angabe (Korrektur in Werkzeuggeometrie normal (nicht invertiert))<br>1 = Korrektur invertiert |
| | | | | EINHUNDERTMILLIONEN: Werkzeugkorrektur<br>0 = Werkzeugkorrektur ohne Duplo-Werkzeuge<br>1 = Werkzeugkorrektur in Duplo-Werkzeug (_DP) |
| 4 | icon+ Anzahl | S_PRNUM | Nummer des Feldes der Messtasterparameter (nicht Messtasternummer) (default=1) | |
| 5 | X0 | S_SETV | Sollwert | |
| 6 | X | S_MA | Messachse (Nummer der Achse) (default=1) | |
| | | | Werte: | 1 = 1. Achse der Ebene (bei G18 Z)<br>2 = 2. Achse der Ebene (bei G18 X)<br>3 = 3. Achse der Ebene (bei G18 Y) [5)] |
| 7 | DFA | S_FA | Messweg | |
| 8 | TSA | S_TSA | Vertrauensbereich | |
| 9 | α | S_STA1 | Startwinkel beim Messen mit Umschlag | |
| 10 | Messungen | S_NMSP | Anzahl Messungen am selben Ort [2)] (default=1) | |
| 11 | T | S_TNAME | Werkzeugname [2)] | |
| 12 | DL | S_DLNUM | Einrichte- Summenkorrektur DL-Nummer [5)] | |
| 13 | ST | _DP | Nummer des zu korrigierenden Schwesterwerkzeugs (Duplo-Nummer) | |
| 14 | TZL | S_TZL | Nullkorrektur [2), 4)] | |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 15 | DIF | S_TDIF | Maßdifferenzkontrolle [2), 4)] | |
| 16 | TUL | S_TUL | Toleranzobergrenze (inkrementell zum Sollwert) [4)] | |
| 17 | TLL | S_TLL | Toleranzuntergrenze (inkrementell zum Sollwert) [4)] | |
| 18 | TMV | S_TMV | Korrekturbereich bei Mittelwertbildung [2)] | |
| 19 | FW | S_K | Wichtungsfaktor für Mittelwertbildung [2)] | |
| 20 | EVN | S_EVNUM | Nummer Erfahrungs-Mittelwertwertspeicher [2), 7)] | |
| 21 | | S_MCBIT | reserviert | |
| 22 | | _DMODE | Displaymode | |
| | | | Werte: | EINER: Bearbeitungsebene G17/G18/G19<br>0 = Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv<br>1 = G17 (nur im Zyklus aktiv)<br>2 = G18 (nur im Zyklus aktiv)<br>3 = G19 (nur im Zyklus aktiv) |
| 23 | | _AMODE | Alternativmode | |
| | | | Werte: | EINER: Maßtoleranz ja/nein<br>0 = nein<br>1 = ja |

1) alle Defaultwerte = 0 oder als default=x gekennzeichnet

2) Anzeige abhängig vom allgemeinen SD54760 $SNS_MEA_FUNCTION_MASK_PIECE

3) Korrektur in NV nur bei Messen ohne Umschlag möglich

4) bei Werkzeugkorrektur im kanalspezifischen MD 20360 TOOL_PARAMETER_DEF_MASK Bit0 und Bit1 beachten

5) nur wenn Funktion "Einrichte-Summenkorrektur" im allgemeinen MD 18108 $MN_MM_NUM_SUMCORR eingerichtet ist. Zusätzlich muss im allgemeinen MD 18080 $MN_MM_TOOL_MANAGEMENT_MASK Bit8=1 gesetzt sein.

6) wenn NVP "fein" in MDs nicht eingerichtet ist, dann wird nach NV "grob" korrigiert

7) Erfahrungs-Mittelwertbildung nur bei Werkzeugkorrektur möglich
Wertbereich der Erfahrungs-Mittelwertspeicher:
1 bis 20 Nummer (n) des Erfahrungswertspeichers, siehe kanalspezifisches SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1]
10000 bis 200000 Nummer (n) des Mittelwertspeichers, siehe kanalspezifisches SD55625 $SCS_MEA_AVERAGE_VALUE[n-1]

### 3.25.3.3 Messzyklenparameter CYCLE994

```
PROC CYCLE994(INT S_MVAR,INT S_KNUM,INT S_KNUM1,INT S_PRNUM,REAL S_SETV,INT S_MA,REAL
S_SZA,REAL S_SZO,REAL S_FA,REAL S_TSA,INT S_NMSP,STRING[32] S_TNAME,INT S_DLNUM,REAL
S_TZL,REAL S_TDIF,REAL S_TUL,REAL S_TLL,REAL S_TMV,INT S_K,INT S_EVNUM,INT S_MCBIT,INT
_DMODE,INT _AMODE,INT _DP)
```

Tabelle 3-10 Aufrufparameter CYCLE994 [1)]

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 1 | | S_MVAR | Messvariante | |
| | | | Werte: | EINER: Innen- oder Außenmessung (default = 1)<br>1 = Innenmessung<br>2 = Außenmessung |
| | | | | ZEHNER: reserviert |
| | | | | HUNDERTER: Korrekturziel<br>0 = nur Messen (keine Korrektur der NV bzw. keine Werkzeugkorrektur)<br>1 = Messen und Ermittlung und Korrektur der NV (siehe S_KNUM) [3)]<br>2 = Messen und Werkzeugkorrektur (siehe S_KNUM1) |
| | | | | TAUSENDER: Umfahrbereich<br>0 = kein Umfahrbereich<br>1 = Umfahrachse 1. Achse der Ebene (bei G18 Z). Messachse siehe S_MA.<br>2 = Umfahrachse 2. Achse der Ebene (bei G18 X). Messachse siehe S_MA.<br>3 = Umfahrachse 3. Achse der Ebene (bei G18 Y). Messachse siehe S_MA. [8)] |
| 2 | Auswahl | S_KNUM | Korrektur Nullpunktverschiebung (NV) oder Basis NV oder Basisbezug [2)] | |
| | | | Werte: | EINER: |
| | | | | ZEHNER:<br>0 = keine Korrektur<br>1 bis max. 99 Nummer der Nullpunktverschiebung oder<br>1 bis max. 16 Nummer der Basisverschiebung |
| | | | | HUNDERTER: reserviert |
| | | | | TAUSENDER: Korrektur NV oder Basis oder Basisbezug<br>0 = Korrektur einstellbare NV<br>1 = Korrektur kanalspezifische Basis NV<br>2 = Korrektur Basisbezug<br>3 = Korrektur globale Basis NV<br>9 = Korrektur aktive NV bzw. bei G500 in letzte aktive kanalspezifische Basis NV |
| | | | | ZEHNTAUSENDER: Korrektur grob oder fein in NV, Basis NV oder Basisbezug<br>0 = Korrektur fein [6)]<br>1 = Korrektur grob |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 3 | Auswahl | S_KNUM1 | Korrektur in Werkzeugkorrektur [2), 4)] | |
| | | | Werte: | EINER: |
| | | | | ZEHNER: |
| | | | | HUNDERTER:<br>0 = keine Korrektur<br>1 bis max. 999 D-Nummer (Schneidennummer) bei Werkzeugkorrektur<br>bei Summen- und Einrichtekorrektur siehe auch S_DLNUM |
| | | | | TAUSENDER: 0 oder eindeutige D-Nummern |
| | | | | ZEHNTAUSENDER: 0 oder eindeutige D-Nummern<br>1 bis max. 32000, wenn eindeutige D-Nummern in MD eingerichtet sind |
| | | | | HUNDERTTAUSENDER: Werkzeugkorrektur [2)]<br>0 = keine Angabe (Korrektur Werkzeuggeometrie)<br>1 = Korrektur Länge L1<br>2 = Korrektur Länge L2<br>3 = Korrektur Länge L3<br>4 = Korrektur Radius |
| | | | | EINEMILLION: Werkzeugkorrektur [2)]<br>0 = keine Angabe (Korrektur des Werkzeuglängen-Verschleiß)<br>1 = Werkzeugkorrektur Summenkorrektur (SK) [5)]<br>Werkzeugkorrekturwert wird zur bestehenden SK addiert<br>2 = Werkzeugkorrektur Einrichtekorrektur (EK) [5)]<br>EK (neu) = EK (alt) + SK (alt) Korrekturwert, SK (neu) = 0<br>3 = Werkzeugkorrektur Einrichtekorrektur (EK) [5)]<br>Werkzeugkorrekturwert wird zur bestehenden EK addiert<br>4 = Werkzeugkorrektur Geometrie |
| | | | | ZEHNMILLION: Werkzeugkorrektur [2)]<br>0 = keine Angabe (Korrektur in Werkzeuggeometrie normal, nicht invertiert)<br>1 = Korrektur invertiert |
| | | | | EINHUNDERTMILLIONEN: Werkzeugkorrektur<br>0 = Werkzeugkorrektur ohne Duplo-Werkzeuge<br>1 = Werkzeugkorrektur in Duplo-Werkzeug (_DP) |
| 4 | icon+ Anzahl | S_PRNUM | Nummer des Feldes der Messtasterparameter (nicht Messtasternummer) (default=1) | |
| 5 | X0 | S_SETV | Sollwert | |
| 6 | X | S_MA | Nummer der Messachse (default=1) | |
| | | | Werte: | 1 = 1. Achse der Ebene (bei G18 Z)<br>2 = 2. Achse der Ebene (bei G18 X)<br>3 = 3. Achse der Ebene (bei G18 Y) [8)] |
| 7 | X1 | S_SZA | Umfahrweite in der Messachse | |
| 8 | Y1 | S_SZO | Umfahrweite in der Umfahrachse | |
| 9 | DFA | S_FA | Messweg | |
| 10 | TSA | S_TSA | Vertrauensbereich | |
| 11 | Messungen | S_NMSP | Anzahl Messungen am selben Ort [2)] (default=1) | |
| 12 | T | S_TNAME | Werkzeugname [2)] | |
| 13 | DL | S_DLNUM | Einrichte- Summenkorrektur DL-Nummer [5)] | |
| 14 | ST | _DP | Nummer des zu korrigierenden Schwesterwerkzeugs (Duplo-Nummer) | |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 15 | TZL | S_TZL | Nullkorrektur [2), 4)] | |
| 16 | DIF | S_TDIF | Maßdifferenzkontrolle [2), 4)] | |
| 17 | TUL | S_TUL | Toleranzobergrenze (inkrementell zum Sollwert) [4)] | |
| 18 | TLL | S_TLL | Toleranzuntergrenze (inkrementell zum Sollwert) [4)] | |
| 19 | TMV | S_TMV | Korrekturbereich bei Mittelwertbildung [2)] | |
| 20 | FW | S_K | Wichtungsfaktor für Mittelwertbildung [2)] | |
| 21 | EVN | S_EVNUM | Nummer Erfahrungswertspeicher [2), 7)] | |
| 22 | | S_MCBIT | reserviert | |
| 23 | | _DMODE | Displaymode | |
| | | | Werte: | EINER: Bearbeitungsebene G17/G18/G19<br>0 = Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv<br>1 = G17 (nur im Zyklus aktiv)<br>2 = G18 (nur im Zyklus aktiv)<br>3 = G19 nur im Zyklus aktiv) |
| 24 | | _AMODE | Alternativmode | |
| | | | Werte: | EINER: Maßtoleranz ja/nein<br>0 = nein<br>1 = ja |

1) alle Defaultwerte = 0 oder als default=x gekennzeichnet

2) Anzeige abhängig vom allgemeinen SD54760 $SNS_MEA_FUNCTION_MASK_PIECE

3) Korrektur in NV nur bei Messen ohne Umschlag möglich

4) bei Werkzeugkorrektur das Kanal-MD 20360 TOOL_PARAMETER_DEF_MASK beachten

5) nur wenn Funktion "Einrichte-Summenkorrektur" im allgemeinen MD 18108 $MN_MM_NUM_SUMCORR eingerichtet ist. Zusätzlich muss das allgemeine MD 18080 $MN_MM_TOOL_MANAGEMENT_MASK Bit8=1 gesetzt sein.

6) wenn NVP "fein" in MDs nicht eingerichtet ist, dann wird nach NV "grob" korrigiert

7) Erfahrungs-Mittelwertbildung nur bei Werkzeugkorrektur möglich
Wertbereich der Erfahrungs-Mittelwertspeicher:
1 bis 20 Nummer (n) des Erfahrungswertspeichers, siehe kanalspezifisches SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1]
10000 bis 200000 Nummer (n) des Mittelwertspeichers, siehe kanalspezifisches SD55625 $SCS_MEA_AVERAGE_VALUE[n-1]

8) wenn Y-Achse an Maschine vorhanden ist

### 3.25.3.4 Messzyklenparameter CYCLE976

```
PROC CYCLE976(INT S_MVAR,INT S_PRNUM,REAL S_SETV,REAL S_SETV0,INT S_MA,INT S_MD,REAL
S_FA,REAL S_TSA,REAL S_VMS,REAL S_STA1,INT S_NMSP,INT S_SETV1,INT _DMODE,INT _AMODE)
```

Tabelle 3-11 Aufrufparameter CYCLE976 [1)]

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 1 | | S_MVAR | Messvariante (default=1000) | |
| | | | Werte: | EINER: Kalibrieren an Fläche, Kalibrierkugel oder in Kalibrierring [2)]<br>0 = Länge an Fläche mit bekanntem Sollwert<br>1 = Radius in Kalibrierring mit bekanntem Durchmesser (Sollwert) und bekanntem Mittelpunkt.<br>2 = Radius in Kalibrierring mit bekanntem Durchmesser (Sollwert) und unbekanntem Mittelpunkt<br>3 = Radius und Länge an Kalibrierkugel<br>4 = Radius an Kante mit bekanntem Sollwert. Auswahl Messachse und Messrichtung beachten. [3)]<br>5 = Radius zwischen zwei Kanten mit bekannten Sollwert und Kantenabstand. Messachse ist auszuwählen. |
| | | | | ZEHNER: reserviert<br>0 = 0 |
| | | | | HUNDERTER: reserviert<br>0 = 0 |
| | | | | TAUSENDER: Auswahl Messachse und Messrichtung beim Kalibrieren<br>0 = keine Angabe (keine Auswahl Messachse und Messrichtung erforderlich) [8)]<br>1 = Auswahl Messachse und Messrichtung angeben, siehe S_MA, S_MD (eine Messrichtung in einer Messachse)<br>2 = Auswahl Messachse angeben, siehe S_MA (zwei Messrichtungen in einer Messachse |
| | | | | ZEHNTAUSENDER: Ermittlung Lageabweichung (Schieflage Messtaster) [2)]<br>0 = Lageabweichung des Messtasters ermitteln [6)]<br>1 = keine Lageabweichung ermitteln |
| | | | | HUNDERTTAUSENDER: Kalibrieren achsparallel oder unter Winkel<br>0 = Kalibrieren achsparallel im aktiven WKS<br>1 = Kalibrieren unter Winkel [7)] |
| | | | | EINEMILLION: Ermittlung Werkzeuglänge bei Kalibrieren an Fläche oder an Kugel<br>0 = keine Ermittlung der Werkzeuglänge<br>1 = Ermittlung der Werkzeuglänge [4)]<br>2 = Kalibrieren der Zustellachse an der Kugel, Ermittlung der Werkzeug-Länge, Eintrag der Werkzeuglängen-Messdifferenz in die Kalibrierdaten |
| 2 | icon+ Anzahl | S_PRNUM | Nummer des Feldes der Messtasterparameter (nicht Messtasternummer) (default=1) | |
| 3 | | S_SETV | Sollwert | |
| 4 | Z0 | S_SETV0 | Sollwert des Längenbezugs beim Abgleich Kugel | |
| 5 | X / Y / Z | S_MA | Messachse (Nummer der Achse) [2), 6)] (default=1) | |
| | | | Werte: | 1 = 1. Achse der Ebene (bei G17 X)<br>2 = 2. Achse der Ebene (bei G17 Y)<br>3 = 3. Achse der Ebene (bei G17 Z) |
| 6 | + - | S_MD | Messrichtung [2), 6)] | |
| | | | Werte: | 0 = positiv<br>1 = negativ |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 7 | DFA | S_FA | Messweg | |
| 8 | TSA | S_TSA | Vertrauensbereich | |
| 9 | VMS | S_VMS | Variable Messgeschwindigkeit bei Kalibrieren [2] | |
| 10 | α | S_STA1 | Startwinkel [2, 5] | |
| 11 | Messungen | S_NMSP | Anzahl Messungen am selben Ort [2] (default=1) | |
| 12 | X0 | S_SETV1 | Bezugspunkt Kante bei Abgleich zwischen 2 Kanten [3] | |
| 13 | | _DMODE | Displaymode | |
| | | | Werte: | EINER: Bearbeitungsebene G17/G18/G19<br>0 = Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv<br>1 = G17 (nur im Zyklus aktiv)<br>2 = G18 (nur im Zyklus aktiv)<br>3 = G19 (nur im Zyklus aktiv) |
| 14 | | _AMODE | Alternativmode | |

[1] alle Defaultwerte = 0 oder als default=x gekennzeichnet

[2] Anzeige abhängig vom allgemeinen SD54760 $SNS_MEA_FUNCTION_MASK_PIECE

[3] Beim Kalibrieren "Radius im Kalibrierring" müssen der Durchmesser und der Mittelpunkt des Ringes bekannt sein (4 Messrichtungen).
Beim Kalibrieren "Radius an zwei Kanten" muss nur der Abstand der Kanten in Richtung der Messachse bekannt sein (2 Messrichtungen).
Beim Kalibrieren "Radius an einer Kante" muss der Sollwert der Fläche bekannt sein.

[4] Messvariante nur Kalibrieren an einer Fläche (Länge an Fläche) korrigierte Werkzeuglänge ergibt sich aus S_MD und S_MA.

[5] nur bei Messvariante "Kalibrierring, ... und bekanntem Mittelpunkt" (S_MVAR=1xxx02).

[6] Messachse nur bei Messvariante S_MVAR=0 oder =xx1x01 oder =xx2x01 oder =20000
Messvariante: "Kalibrieren an einer Fläche" → Auswahl Messachse und Messrichtung
oder am "Kalibrierring, ... und bekannten Mittelpunkt" → Auswahl eine Achsrichtung und Auswahl Messachse und Messrichtung
oder am "Kalibrierring, ... und bekannten Mittelpunkt" → Auswahl zwei Achsrichtungen und Auswahl Messachse
oder "Ermittlung der Messtasterlänge" → S_MA=3 → 3. Achse der Ebene (bei G17 Z)

[7] Messvariante nur Kalibrieren im Kalibrierring oder an Kalibrierkugel
Beim "Kalibrieren an Kalibrierkugel" wird beim Messen unter Winkel die Kugel am Äquator umfahren.

[8] Bei Kalibrieren "Radius in Kalibrierring" mit unbekannten Mittelpunkt vier Messrichtungen in der Ebene (bei G17 +-X +-Y).
Bei Kalibrieren "Länge an Fläche" in Minusrichtung der Werkzeugachse (bei G17 -Z).

### 3.25.3.5 Messzyklenparameter CYCLE978

```
PROC CYCLE978(INT S_MVAR,INT S_KNUM,INT S_KNUM1,INT S_PRNUM,REAL S_SETV,REAL S_FA,REAL
S_TSA,INT S_MA,INT S_MD,INT S_NMSP,STRING[32] S_TNAME,INT S_DLNUM,REAL S_TZL,REAL
S_TDIF,REAL S_TUL,REAL S_TLL,REAL S_TMV,INT S_K,INT S_EVNUM,INT S_MCBIT,INT _DMODE,INT
_AMODE,INT _DP)
```

Tabelle 3-12 Aufrufparameter CYCLE978 [1)]

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 1 | | S_MVAR | Messvariante | |
| | | | Werte: | EINER: Konturelement<br>0 = Fläche messen |
| | | | | ZEHNER: reserviert |
| | | | | HUNDERTER: Korrekturziel<br>0 = nur Messen (keine Korrektur der NV bzw. keine Werkzeugkorrektur)<br>1 = Messen, Ermittlung und Korrektur der NV (siehe S_KNUM)<br>2 = Messen und Werkzeugkorrektur (siehe S_KNUM1) |
| | | | | TAUSENDER: reserviert |
| | | | | ZEHNTAUSENDER: Messen mit/ohne Spindelumschlag oder Messtaster in Schaltrichtung ausrichten [9)]<br>0 = Messen ohne Spindelumschlag, ohne Messtaster ausrichten<br>1 = Messen mit Spindelumschlag<br>2 = Messtaster in Schaltrichtung ausrichten |
| 2 | Auswahl | S_KNUM | Korrektur Nullpunktverschiebung (NV) oder Basis NV oder Basisbezug [2)] | |
| | | | Werte: | EINER: |
| | | | | ZEHNER:<br>0 = keine Korrektur<br>1 bis max. 99 Nummer der Nullpunktverschiebung oder<br>1 bis max. 16 Nummer der Basisverschiebung |
| | | | | HUNDERTER: reserviert |
| | | | | TAUSENDER: Korrektur NV oder Basis oder Basisbezug<br>0 = Korrektur einstellbare NV<br>1 = Korrektur kanalspezifische Basis NV<br>2 = Korrektur Basisbezug<br>3 = Korrektur globale Basis NV<br>9 = Korrektur aktive NV bzw. bei G500 in letzte aktive kanalspezifische Basis NV |
| | | | | ZEHNTAUSENDER: Korrektur grob oder fein in NV, Basis NV oder Basisbezug<br>0 = Korrektur fein [6)]<br>1 = Korrektur grob |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 3 | Auswahl | S_KNUM1 | Korrektur in Werkzeugkorrektur [2] | |
| | | | Werte: | EINER: |
| | | | | ZEHNER: |
| | | | | HUNDERTER:<br>0 = keine Korrektur<br>1 bis max. 999 D-Nummer (Schneidennummer) bei Werkzeugkorrektur, bei Summen- und Einrichtekorrektur siehe auch S_DLNUM |
| | | | | TAUSENDER: 0 oder eindeutige D-Nummern |
| | | | | ZEHNTAUSENDER: 0 oder eindeutige D-Nummern<br>1 bis max. 32000, wenn eindeutige D-Nummern in MDs eingerichtet sind |
| | | | | HUNDERTTAUSENDER: Werkzeugkorrektur [2]<br>0 = keine Angabe (Korrektur in Werkzeuggeometrie)<br>1 = Korrektur Länge L1<br>2 = Korrektur Länge L2<br>3 = Korrektur Länge L3<br>4 = Korrektur Radius |
| | | | | EINEMILLION: Werkzeugkorrektur [2]<br>0 = keine Angabe (Korrektur des Werkzeugradius-Verschleiß)<br>1 = Werkzeugkorrektur Summenkorrektur (SK) [5]<br>Werkzeugkorrekturwert wird zur bestehenden SK addiert<br>2 = Werkzeugkorrektur Einrichtekorrektur (EK) [5]<br>EK (neu) = EK (alt) + SK (alt) Korrekturwert, SK (neu) = 0<br>3 = Werkzeugkorrektur Einrichtekorrektur (EK) [5]<br>Werkzeugkorrekturwert wird zur bestehenden EK addiert<br>4 = Werkzeugkorrektur Geometrie |
| | | | | ZEHNMILLION: Werkzeugkorrektur [2]<br>0 = keine Angabe (Korrektur in Werkzeuggeometrie normal, nicht invertiert)<br>1 = Korrektur invertiert |
| | | | | EINHUNDERTMILLIONEN: Werkzeugkorrektur<br>0 = Werkzeugkorrektur ohne Duplo-Werkzeuge<br>1 = Werkzeugkorrektur in Duplo-Werkzeug (_DP) |
| 4 | icon+ Anzahl | S_PRNUM | Nummer des Feldes der Messtasterparameter (nicht Messtasternummer) (Wertebereich 1 bis 40) | |
| 5 | X0 | S_SETV | Sollwert | |
| 6 | DFA | S_FA | Messweg | |
| 7 | TSA | S_TSA | Vertrauensbereich | |
| 8 | X | S_MA | Nummer der Messachse [7] (Wertebereich 1 bis 3) | |
| | | | Werte: | 1 = 1. Achse der Ebene (bei G17 X)<br>2 = 2. Achse der Ebene (bei G17 Y)<br>3 = 3. Achse der Ebene (bei G17 Z) Messen in Werkzeugrichtung |
| 9 | | S_MD | Messrichtung der Messachse | |
| | | | Werte: | 1 = positive Messrichtung<br>2 = negative Messrichtung |
| 10 | Messungen | S_NMSP | Anzahl Messungen am selben Ort [2] (Wertebereich 1 bis 9) | |
| 11 | TR | S_TNAME | Werkzeugname [3] | |
| 12 | DL | S_DLNUM | Einrichte- Summenkorrektur DL-Nummer [5] | |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 13 | ST | _DP | Nummer des zu korrigierenden Schwesterwerkzeugs (Duplo-Nummer) | |
| 14 | TZL | S_TZL | Nullkorrektur [2), 3)] | |
| 15 | DIF | S_TDIF | Maßdifferenzkontrolle [2), 3)] | |
| 16 | TUL | S_TUL | Toleranzobergrenze (inkrementell zum Sollwert) [3)] | |
| 17 | TLL | S_TLL | Toleranzuntergrenze (inkrementell zum Sollwert) [3)] | |
| 18 | TMV | S_TMV | Korrekturbereich bei Mittelwertbildung [2)] | |
| 19 | FW | S_K | Wichtungsfaktor für Mittelwertbildung [2)] | |
| 20 | EVN | S_EVNUM | Datensatz Erfahrungswertspeicher [2), 8)] | |
| 21 | | S_MCBIT | reserviert | |
| 22 | | _DMODE | Displaymode | |
| | | | Werte: | EINER: Bearbeitungsebene G17 / G18 / G19<br>0 = Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv<br>1 = G17 (nur im Zyklus aktiv)<br>2 = G18 (nur im Zyklus aktiv)<br>3 = G19 (nur im Zyklus aktiv) |
| 23 | | _AMODE | Alternativmode | |
| | | | Werte: | EINER: Maßtoleranz ja/nein<br>0 = nein<br>1 = ja |

1) alle Defaultwerte = 0 oder als Wertebereich a bis b gekennzeichnet

2) Anzeige abhängig vom allgemeinen SD54760 $SNS_MEA_FUNCTION_MASK_PIECE

3) nur bei Korrektur in Werkzeug, ansonsten Parameter = ""

4) nur bei Korrektur in Werkzeug und Maßtoleranz "ja", ansonsten Parameter = 0

5) nur wenn Funktion "Einrichte-Summenkorrektur" im allgemeinen MD 18108 $MN_MM_NUM_SUMCORR eingerichtet ist. Zusätzlich muss im allgemeinen MD 18080 $MN_MM_TOOL_MANAGEMENT_MASK bit8=1 gesetzt sein.

6) wenn NVP "fein" in MDs nicht eingerichtet ist, dann wird nach NV "grob" korrigiert

7) Korrektur in Werkzeuggeometrie:
- bei Messen in der Ebene (S_MA=1 oder S_MA=2) Korrektur in Werkzeugradius
- bei Messen in Werkzeugrichtung (S_MA=3) Korrektur in Werkzeuglänge L1

8) Erfahrungs-Mittelwertbildung bei Werkzeugkorrektur und Korrektur in NV möglich
Wertbereich der Erfahrungs-Mittelwertspeicher:
1 bis 20 Nummer (n) des Erfahrungswertspeichers, siehe kanalspezifisches SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1]
10000 bis 200000 Nummer (n) des Mittelwertspeichers, siehe kanalspezifisches SD55625 $SCS_MEA_AVERAGE_VALUE[n-1]

9) Beim Messen mit Spindelumschlag muss der Radius/Durchmesser des Messtasters exakt bestimmt sein. Dies sollte mit einer Kalibriervariante des CYCLE976 Radius an Ring oder an Kante oder an Kugel erfolgen. Ansonsten wird das Messergebnis verfälscht.

### 3.25.3.6 Messzyklenparameter CYCLE998

```
PROC CYCLE998(INT S_MVAR,INT S_KNUM,INT S_RA,INT S_PRNUM,REAL S_SETV,REAL S_STA1,REAL
S_INCA,REAL S_FA,REAL S_TSA,INT S_MA,INT S_MD,REAL S_ID,REAL S_SETV0,REAL S_SETV1,REAL
S_SETV2,REAL S_SETV3,INT S_NMSP,INT S_EVNUM,INT _DMODE,INT _AMODE)
```

Tabelle 3-13 Aufrufparameter CYCLE998 [1)]

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 1 | | S_MVAR | Messvariante (default=5) | |
| | | | Werte: | EINER: Konturelement<br>5 = Kante messen (ein Winkel)<br>6 = Ebene messen (zwei Winkel) |
| | | | | ZEHNER: reserviert |
| | | | | HUNDERTER: Korrekturziel<br>0 = nur Messen und keine Korrektur der NV<br>1 = Messen und Ermittlung und Korrektur der NV (siehe S_KNUM) |
| | | | | TAUSENDER: Schutzzone<br>0 = keine Berücksichtigung einer Schutzzone<br>1 = Berücksichtigung einer Schutzzone |
| | | | | ZEHNTAUSENDER: Messen mit Spindelumschlag (Differenzmessung)<br>0 = Messen ohne Spindelumschlag<br>1 = Messen mit Spindelumschlag |
| | | | | HUNDERTTAUSENDER: Messen unter Winkel oder achsparallel<br>0 = Messen unter Winkel<br>1 = Messen achsparallel |
| 2 | Auswahl | S_KNUM | Korrektur Nullpunktverschiebung (NV) oder Basis NV oder Basisbezug [2)] | |
| | | | Werte: | EINER: |
| | | | | ZEHNER:<br>0 = keine Korrektur<br>1 bis max. 99 Nummer der Nullpunktverschiebung oder<br>1 bis max. 16 Nummer der Basisverschiebung |
| | | | | HUNDERTER: reserviert |
| | | | | TAUSENDER: Korrektur NV oder Basis oder Basisbezug<br>0 = Korrektur einstellbare NV<br>1 = Korrektur kanalspezifische Basis NV<br>2 = Korrektur Basisbezug<br>9 = Korrektur aktive NV bzw. bei G500 in letzte aktive kanalspezifische Basis NV |
| | | | | ZEHNTAUSENDER: Korrektur grob oder fein in NV oder Basis NV oder Basisbezug [3)]<br>0 = Korrektur fein<br>1 = Korrektur grob |
| 3 | A, B, C | S_RA | Korrekturziel Koordinatendrehung oder Rundachse [14)] | |
| | | | Werte: | 0 = Korrekturziel Koordinatendrehung um die Achse die sich aus Parameter S_MA ergibt [4)]<br>>0 = Korrekturziel Rundachse. Nummer der Kanal-Achsnummer der Rundachse (vorzugsweise Rundtisch). Die Winkelkorrektur erfolgt im translatorischen Anteil der NV der Rundachse. |
| 4 | icon+ Anzahl | S_PRNUM | Nummer des Feldes der Messtasterparameter (default=1) | |
| 5 | DX / DY / DZ | S_SETV | Weg (inkrementell) von der Startposition bis Messpunkt P1 der Messachse (S_MA) [5)] | |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 6 | α | S_STA1 | Winkelsollwert bei "Kante ausrichten" bzw. bei "Ebene ausrichten" um 1. Achse der Ebene (bei G17 X) [9] | |
| 7 | β | S_INCA | Winkelsollwert bei "Ebene ausrichten" um 2. Achse der Ebene (bei G17 Y) [9] | |
| 8 | DFA | S_FA | Messweg | |
| 9 | TSA | S_TSA | Vertrauensbereich<br>Überwachung der Winkeldifferenz zum Winkelsollwert [Grad] [6] | |
| 10 | X / Y / Z | S_MA | Messachse, Versetzachse [7] (default=201) | |
| | | | Werte: | EINER: Nummer der Messachse<br>1 = 1. Achse der Ebene (bei G17 X)<br>2 = 2. Achse der Ebene (bei G17 Y)<br>3 = 3. Achse der Ebene (bei G17 Z) |
| | | | | ZEHNER: reserviert |
| | | | | HUNDERTER: Nummer der Versetzachse<br>1 = 1. Achse der Ebene (bei G17 X)<br>2 = 2. Achse der Ebene (bei G17 Y)<br>3 = 3. Achse der Ebene (bei G17 Z) |
| 11 | +- | S_MD | Messrichtung der Messachse [8] | |
| | | | Werte: | 0 = Messrichtung wird aus dem Sollwert und der Istposition der Messachse ermittelt (Kompatibilität)<br>1 = positive Messrichtung<br>2 = negative Messrichtung |
| 12 | L2 | S_ID | Bei Messvariante "Kante ausrichten":<br>Abstand (inkrementell) zwischen den Messpunkten P1 und P2 in Versetzachse (Wert >0)<br>Bei Messvariante "Ebene ausrichten" gelten die nachfolgend aufgeführten Parameter. | |
| 13 | L2 | S_SETV0 | Abstand zwischen den Messpunkten P1 und P2 in der 1. Achse der Ebene [10] | |
| 14 | | S_SETV1 | Abstand zwischen den Messpunkten P1 und P2 in der 2. Achse der Ebene [11], [12] | |
| 15 | L3x | S_SETV2 | Abstand zwischen den Messpunkten P1 und P3 in der 1. Achse der Ebene [11] | |
| 16 | L3y | S_SETV3 | Abstand zwischen den Messpunkten P1 und P3 in der 2. Achse der Ebene [10] | |
| 17 | Messungen | S_NMSP | Anzahl Messungen am selben Ort [2] (default=1) | |
| 18 | | S_EVNUM | Datensatz Erfahrungswertspeicher [2], [13] | |
| 19 | | _DMODE | Displaymode | |
| | | | Werte: | EINER: Bearbeitungsebene G17/G18/G19<br>0 = Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv<br>1 = G17 (nur im Zyklus aktiv)<br>2 = G18 (nur im Zyklus aktiv)<br>3 = G19 (nur im Zyklus aktiv) |
| 20 | | _AMODE | Reserviert (Alternativmode) | |

[1] alle Defaultwerte = 0 oder als default=x gekennzeichnet

[2] Anzeige abhängig vom allgemeinen SD54760 $SNS_MEA_FUNCTION_MASK_PIECE

[3] NVP "fein" nur, wenn Korrekturziel Rundachse und MD 52207 $MCS_AXIS_USAGE_ATTRIB[n] Bit6=1.
Wenn NV in MDs nicht eingerichtet ist, dann wird nach NV "grob" korrigiert.

[4] Beispiel für Korrektur in Koordinatendrehung: S_MA=102 Messachse Y Versetzachse X ergibt Koordinatendrehung um Z (bei G17)

[5] Wert nur relevant bei Schutzzone "ja" (S_MVAR TAUSENDER-Stelle = 1)

[6] Beim Positionieren von Messpunkt P1 zum Messpunkt P2 in der Versetzachse werden die Winkel in den Parametern `S_STA1` und `S_TSA` addiert.

[7] Nummer der Messachse muss ungleich Nummer der Versetzachse sein (z. B. 101 nicht erlaubt)

[8] Messrichtung nur "Kante ausrichten" und "Messen achsparallel" (`S_MVAR=10x105`)

[9] Winkelbereich `S_STA1` ±45 Grad bei "Kante ausrichten"
Winkelbereich `S_STA1` 0 bis +60 Grad und `S_INCA` ±30 Grad bei "Ebene ausrichten"

[10] Bei Messvariante "Ebene ausrichten" und "Kante ausrichten"

[11] Bei Messvariante "Ebene messen" und "Messen achsparallel"

[12] nicht für Messzyklenstand SW04.04.

[13] Erfahrungswertbildung bei Korrektur in NV; Wertbereich der Erfahrungs- Mittelwertspeicher:
1 bis 20 Nummer (n) des Erfahrungswertspeichers siehe kanalspezifisches SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1]

[14] Der Parameter S_RA ist nur für der Messvariante "Kante ausrichten" relevant (S_MVAR xxxxx5).

### 3.25.3.7 Messzyklenparameter CYCLE977

```
PROC CYCLE977(INT S_MVAR,INT S_KNUM,INT S_KNUM1,INT S_PRNUM,REAL S_SETV,REAL S_SETV0,REAL
S_SETV1,REAL S_FA,REAL S_TSA,REAL S_STA1,REAL S_ID,REAL S_SZA,REAL S_SZO,INT S_MA,INT
S_NMSP,STRING[32] S_TNAME,INT S_DLNUM,REAL S_TZL,REAL S_TDIF,REAL S_TUL,REAL S_TLL,REAL
S_TMV,INT S_K,INT S_EVNUM,INT S_MCBIT,INT _DMODE,INT _AMODE,REAL S_XM,REAL_S_YM,INT _DP)
```

Tabelle 3-14 Aufrufparameter CYCLE977 [1]

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 1 | | `S_MVAR` | Messvariante | |
| | | | Werte: | EINER: Konturelement (Wertebereich 1 bis 6)<br>1 = Bohrung messen<br>2 = Zapfen (Welle) messen<br>3 = Nut messen<br>4 = Steg messen<br>5 = Rechteck messen, innen<br>6 = Rechteck messen, außen |
| | | | | ZEHNER: reserviert |
| | | | | HUNDERTER: Korrekturziel<br>0 = nur Messen (keine Korrektur der NV bzw. keine Werkzeugkorrektur)<br>1 = Messen und Ermittlung und Korrektur der NV (siehe `S_KNUM`)<br>2 = Messen und Werkzeugkorrektur (siehe `S_KNUM1`) |
| | | | | TAUSENDER: Schutzzone<br>0 = keine Berücksichtigung einer Schutzzone<br>1 = Berücksichtigung einer Schutzzone |
| | | | | ZEHNTAUSENDER: Messen mit/ohne Spindelumschlag (Differenzmessung) oder Messtaster in Schaltrichtung ausrichten<br>0 = Messen ohne Spindelumschlag, Messtaster nicht ausrichten<br>1 = Messen mit Spindelumschlag<br>2 = Messtaster in Schaltrichtung ausrichten |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 2 | Auswahl | S_KNUM | Korrektur Nullpunktverschiebung (NV) oder Basis NV oder Basisbezug [2] | |
| | | | Werte: | EINER: |
| | | | | ZEHNER:<br>0 = keine Korrektur<br>1 bis max. 99 Nummer der Nullpunktverschiebung oder<br>1 bis max. 16 Nummer der Basisverschiebung |
| | | | | HUNDERTER: reserviert |
| | | | | TAUSENDER: Korrektur NV oder Basis oder Basisbezug<br>0 = Korrektur einstellbare NV<br>1 = Korrektur kanalspezifische Basis NV<br>2 = Korrektur Basisbezug<br>3 = Korrektur globale Basis NV<br>9 = Korrektur aktive NV bzw. bei G500 in letzte aktive kanalspezifische Basis NV |
| | | | | ZEHNTAUSENDER: Korrektur grob oder fein in NV, Basis NV oder Basisbezug<br>0 = Korrektur fein [6]<br>1 = Korrektur grob |
| 3 | Auswahl | S_KNUM1 | Korrektur in Werkzeugkorrektur [2] | |
| | | | Werte: | EINER: |
| | | | | ZEHNER: |
| | | | | HUNDERTER:<br>0 = keine Korrektur<br>1 bis max. 999 D-Nummer (Schneidennummer) bei Werkzeugkorrektur; bei Summen- und Einrichtekorrektur siehe auch S_DLNUM |
| | | | | TAUSENDER: 0 oder eindeutige D-Nummern |
| | | | | ZEHNTAUSENDER: 0 oder eindeutige D-Nummern<br>1 bis max. 32000, wenn eindeutige D-Nummern in MDs eingerichtet sind |
| | | | | HUNDERTTAUSENDER: Werkzeugkorrektur [2]<br>0 = keine Angabe (Korrektur Werkzeugradius)<br>1 = Korrektur Länge L1<br>2 = Korrektur Länge L2<br>3 = Korrektur Länge L3<br>4 = Korrektur Radius |
| | | | | EINEMILLION: Werkzeugkorrektur [2]<br>0 = keine Angabe (Korrektur des Werkzeugradius-Verschleißes)<br>1 = Werkzeugkorrektur Summenkorrektur (SK) [5]<br>Werkzeugkorrekturwert wird zur bestehenden SK addiert<br>2 = Werkzeugkorrektur Einrichtekorrektur (EK) [5]<br>EK (neu) = EK (alt) + SK (alt) Korrekturwert, SK (neu) = 0<br>3 = Werkzeugkorrektur Einrichtekorrektur (EK) [5]<br>Werkzeugkorrekturwert wird zur bestehenden EK addiert<br>4 = Werkzeugkorrektur Geometrie |
| | | | | ZEHNMILLION: Werkzeugkorrektur [2]<br>0 = keine Angabe (Korrektur in Werkzeuggeometrie normal, nicht invertiert)<br>1 = Korrektur invertiert |
| | | | | EINHUNDERTMILLIONEN: Werkzeugkorrektur<br>0 = Werkzeugkorrektur ohne Duplo-Werkzeuge<br>1 = Werkzeugkorrektur in Duplo-Werkzeug (_DP) |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung |
|---|---|---|---|
| 4 | icon+ Anzahl | S_PRNUM | Nummer des Feldes der Messtasterparameter (nicht Messtasternummer) (Wertebereich 1 bis 40) |
| 5 | X0 | S_SETV | Sollwert |
| 6 | X0 | S_SETV0 | Sollwert bei Rechteck 1. Achse der Ebene (X bei G17) |
| 7 | Y0 | S_SETV1 | Sollwert bei Rechteck 2. Achse der Ebene (Y bei G17) |
| 8 | XM | S_XM | Sollwert-Mittelpunktvorgabe Geometrieachse X |
| 9 | YM | S_YM | Sollwert-Mittelpunktvorgabe Geometrieachse Y |
| 10 | DFA | S_FA | Messweg |
| 11 | TSA | S_TSA | Vertrauensbereich |
| 12 | α 0 | S_STA1 | Startwinkel |
| 13 | DZ | S_ID | Inkrementeller Betrag<br>1. Inkrementelle Zustellung der 3. Achse der Ebene (Z bei G17)<br>Zustellrichtung über Vorzeichen von S_ID. Bei Messen Zapfen, Steg und Rechteck außen wird mit S_ID das Absenken auf Messhöhe definiert.<br>2. Berücksichtigen einer Schutzzone<br>Bei Messen Bohrung, Nut und Rechteck innen und einer Schutzzone wird mit S_ID die Überfahrhöhe definiert. |
| 14 | X1 | S_SZA | Durchmesser oder Länge (Breite) der Schutzzone [7] |
| 15 | Y1 | S_SZO | Bei Rechteck messen: Breite der Schutzzone der 2. Achse der Ebene |
| 16 | X | S_MA | Nummer der Messachse [7] (nur bei Messen Nut oder Steg) |
| | | | Werte: 1 = 1. Achse der Ebene (bei G17 X)<br>2 = 2. Achse der Ebene (bei G17 Y) |
| 17 | ST | _DP | Nummer des zu korrigierenden Schwesterwerkzeugs (Duplo-Nummer) |
| 18 | Messungen | S_NMSP | Anzahl Messungen am selben Ort [2] (Wertebereich 1 bis 9) |
| 19 | TR | S_TNAME | Werkzeugname [2] |
| 20 | DL | S_DLNUM | Einrichte- Summenkorrektur DL-Nummer [5] |
| 21 | TZL | S_TZL | Nullkorrektur [2], [4] |
| 22 | DIF | S_TDIF | Maßdifferenzkontrolle [2], [4] |
| 23 | TUL | S_TUL | Toleranzobergrenze (inkrementell zum Sollwert) [4] |
| 24 | TLL | S_TLL | Toleranzuntergrenze (inkrementell zum Sollwert) [4] |
| 25 | TMV | S_TMV | Korrekturbereich bei Mittelwertbildung [2] |
| 26 | FW | S_K | Wichtungsfaktor für Mittelwertbildung [2] |
| 27 | | S_EVNUM | Datensatz Erfahrungs-Mittelwertspeicher [2], [8] |
| 28 | | S_MCBIT | reserviert |
| 29 | | _DMODE | Displaymode |
| | | | Werte: EINER: Bearbeitungsebene G17/G18/G19<br>0 = Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv<br>1 = G17 (nur im Zyklus aktiv)<br>2 = G18 (nur im Zyklus aktiv)<br>3 = G19 (nur im Zyklus aktiv) |
| 30 | | _AMODE | Alternativmode |
| | | | Werte: EINER: Maßtoleranz ja/nein<br>0 = nein<br>1 = ja |

1) alle Defaultwerte = 0 oder als Wertebereich a bis b gekennzeichnet

2) Anzeige abhängig vom allgemeinen SD54760 $SNS_MEA_FUNCTION_MASK_PIECE

3) nur bei Korrektur in Werkzeug, ansonsten Parameter = ""

4) nur bei Korrektur in Werkzeug und Maßtoleranz "ja", ansonsten Parameter = 0

5) nur wenn Funktion "Einrichte-Summenkorrektur" im allgemeinen MD 18108 $MN_MM_NUM_SUMCORR eingerichtet ist. Zusätzlich muss im allgemeinen MD 18080 $MN_MM_TOOL_MANAGEMENT_MASK Bit8=1 gesetzt sein.

6) wenn NVP "fein" in MDs nicht eingerichtet ist, dann wird nach NV "grob" korrigiert

7) Durchmesser bzw. Breite der Schutzzone innerhalb von Bohrung oder Nut.
Durchmesser bzw. Breite der Schutzzone außerhalb von Zapfen oder Steg

8) Erfahrungs-Mittelwertbildung bei Werkzeugkorrektur möglich
Wertbereich der Erfahrungs-Mittelwertspeicher:
1 bis 20 Nummer (n) des Erfahrungswertspeichers, siehe kanalspezifisches SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1]
10000 bis 200000 Nummer (n) des Mittelwertspeichers, siehe kanalspezifisches SD55625 $SCS_MEA_AVERAGE_VALUE[n-1]

### 3.25.3.8 Messzyklenparameter CYCLE961

```
PROC CYCLE961(INT S_MVAR,INT S_KNUM,INT S_PRNUM,REAL S_SETV0,REAL S_SETV1,REAL
S_SETV2,REAL S_SETV3,REAL S_SETV4,REAL S_SETV5,REAL S_SETV6,REAL S_SETV7,REAL S_SETV8,REAL
S_SETV9,REAL S_STA1,REAL S_INCA,REAL S_ID,REAL S_FA,REAL S_TSA,INT S_NMSP,INT S_MCBIT,INT
_DMODE,INT _AMODE)
```

Tabelle 3-15 Aufrufparameter CYCLE961 [1)]

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 1 | | S_MVAR | Messvariante (default ≥ 6) | |
| | | | Werte: | EINER: Konturelement<br>5 = Einrichten rechtwinklige Innenecke, Sollwertvorgabe Winkel und Abstände A1 bis A3<br>6 = Einrichten rechtwinklige Außenecke, Sollwertvorgabe Winkel und Abstände A1 bis A3<br>7 = Einrichten Innenecke, Vorgabe Winkel und Abstände A1 bis A4<br>8 = Einrichten Außenecke, Vorgabe Winkel und Abstände A1 bis A4 |
| | | | | ZEHNER: Sollwertvorgabe als Abstand oder über vier Punkte<br>0 = Sollwertvorgabe als Abstand (polar)<br>1 = Sollwertvorgabe über vier Punkte (Messpunkte P1 bis P4) |
| | | | | HUNDERTER: Korrekturziel<br>0 = nur Messen (keine Korrektur der NV bzw. keine Werkzeugkorrektur)<br>1 = Messen und Ermittlung und Korrektur der NV, siehe S_KNUM |
| | | | | TAUSENDER: Schutzzone<br>0 = keine Berücksichtigung einer Schutzzone (Hindernis)<br>1 = Berücksichtigung einer Schutzzone (Hindernis), siehe S_ID |
| | | | | ZEHNTAUSENDER: Lage der Ecke im WKS<br>0 = Lage der Ecke wird über Parameter S_STA1 bestimmt (Kompatibilität)<br>1 = Lage 1 der Ecke im positionierten Startpunkt der Messung [6)]<br>2 = Lage 2 der Ecke, Abstände in der 1. Achse der Ebene (bei G17 X) sind negativ (siehe S_SETV0, S_SETV1)<br>3 = Lage 3 der Ecke, Abstände in der 1. und 2. Achse der Ebene (bei G17 XY) sind negativ (siehe S_SETV0 bis S_SETV3)<br>4 = Lage 4 der Ecke, Abstände in der 2. Achse der Ebene (bei G17 Y) sind negativ (siehe S_SETV2, S_SETV3) |
| 2 | Auswahl | S_KNUM | Korrektur Nullpunktverschiebung (NV) oder Basis NV oder Basisbezug [2)] | |
| | | | Werte: | EINER: |
| | | | | ZEHNER:<br>0 = keine Korrektur<br>1 bis max. 99 Nummer der Nullpunktverschiebung oder<br>1 bis max. 16 Nummer der Basisverschiebung |
| | | | | HUNDERTER: reserviert |
| | | | | TAUSENDER: Korrektur NV oder Basis oder Basisbezug<br>0 = Korrektur einstellbare NV<br>1 = Korrektur kanalspezifische Basis NV<br>2 = Korrektur Basisbezug<br>9 = Korrektur aktive NV bzw. bei G500 letzte aktive kanalspezifische Basis NV |
| | | | | ZEHNTAUSENDER: Korrektur grob oder fein in NV, Basis NV oder Basisbezug<br>0 = Korrektur fein [5)]<br>1 = Korrektur grob |
| 3 | icon+ Anzahl | S_PRNUM | Nummer des Feldes der Messtasterparameter (nicht Messtasternummer) (Wertebereich 1 bis 40) | |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung |
|---|---|---|---|
| 4 | L1/X1 | S_SETV0 | Abstand L1 zwischen dem Pol und Messpunkt P1 in Richtung der 1. Achse der Ebene (bei G17 X) [3]<br>(Ist Abstand L1=0, wird automatisch L1 = M_SETV1 / 2 berechnet) oder<br>Startpunkt P1x der 1. Achse der Ebene (bei G17 X) [4] |
| 5 | L2/Y1 | S_SETV1 | Abstand L2 zwischen dem Pol und Messpunkt P2 in Richtung der 1. Achse der Ebene [3]<br>oder Startpunkt P1y der 2. Achse der Ebene (bei G17 Y) [4] |
| 6 | L3/X2 | S_SETV2 | Abstand L3 zwischen dem Pol und Messpunkt P3 in Richtung der 2. Achse der Ebene [3]<br>(Ist Abstand L3=0, wird bei einer nicht rechtwinkligen Ecke automatisch L3 = M_SETV3 / 2 berechnet)<br>oder Startpunkt P2x der 1. Achse der Ebene [4] |
| 7 | L4/Y2 | S_SETV3 | Abstand L4 zwischen dem Pol und Messpunkt P3 in Richtung der 2. Achse der Ebene bei nichtrechtwinkliger Ecke [3]<br>oder Startpunkt P2y der 2. Achse der Ebene [4] |
| 8 | XP/X3 | S_SETV4 | Lage des Pols in der 1. Achse der Ebene [3]<br>oder Startpunkt P3x der 1. Achse der Ebene [4] |
| 9 | XP/Y3 | S_SETV5 | Lage des Pols in der 2. Achse der Ebene [3]<br>oder Startpunkt P3y der 2. Achse der Ebene [4] |
| 10 | X4 | S_SETV6 | Startpunkt P4x der 1. Achse der Ebene [4] |
| 11 | Y4 | S_SETV7 | Startpunkt P4y der 2. Achse der Ebene [4] |
| 12 | X0 | S_SETV8 | Sollwert der gemessenen Ecke in der 1. Achse der Ebene beim Korrigieren in NV |
| 13 | Y0 | S_SETV9 | Sollwert der gemessenen Ecke in der 2. Achse der Ebene beim Korrigieren in NV |
| 14 | α0 | S_STA1 | Startwinkel von positiver Richtung der 1. Achse der Ebene zur Bezugskante des Werkstücks im MKS (+-270 Grad) |
| 15 | α1 | S_INCA | Winkel zwischen Werkstückbezugskanten beim Messen einer nicht rechtwinkligen Ecke [7] |
| 16 | DZ | S_ID | Zustellbetrag auf Messhöhe bei jedem Messpunkt bei aktivem Schutzbereich<br>(siehe S_MVAR). |
| 17 | DFA | S_FA | Messweg |
| 18 | TSA | S_TSA | Vertrauensbereich<br>Überwachung der Winkeldifferenz zum Winkelsollwert [Grad] |
| 19 | Messungen | S_NMSP | Anzahl Messungen am selben Ort [2] (Wertebereich 1 bis 9) [2] |
| 20 | | S_MCBIT | reserviert |
| 21 | | _DMODE | Displaymode |
| | | | Werte: EINER: Bearbeitungsebene G17/G18/G19<br>0 = Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv<br>1 = G17 (nur im Zyklus aktiv)<br>2 = G18 (nur im Zyklus aktiv)<br>3 = G19 (nur im Zyklus aktiv) |
| 22 | | _AMODE | Alternativmode |

1) alle Defaultwerte = 0 oder als Wertebereich a bis b gekennzeichnet

2) Anzeige abhängig vom allgemeinen SD54760 $SNS_MEA_FUNCTION_MASK_PIECE

3) Vorgabe der Messpunkte in Polarkoordinaten unter Berücksichtigung des Startwinkels S_STA1 bei Messpunkt 3 oder 4 des Fortschaltwinkels S_INCA.

4) Vorgabe der Messpunkte im rechtwinkligen Koordinatensystem (Vorgabe über 4 Punkte)

5) Wenn NV "fein" in MDs nicht eingerichtet ist, dann wird nach NV "grob" korrigiert.

6) Wertebereich Winkel S_INCA: -180 bis +180 Grad

7) Startwinkel S_STA1, Wertebereich: rechtwinklige Ecke: +- 90 Grad, beliebige Ecke: +- 45 Grad

### 3.25.3.9 Messzyklenparameter CYCLE979

```
PROC CYCLE979(INT S_MVAR,INT S_KNUM,INT S_KNUM1,INT S_PRNUM,REAL S_SETV,REAL S_FA,REAL
S_TSA,REAL S_CPA,REAL S_CPO,REAL S_STA1,REAL S_INCA,INT S_NMSP,STRING[32] S_TNAME,REAL
S_DLNUM,REAL S_TZL,REAL S_TDIF,REAL S_TUL,REAL S_TLL,REAL S_TMV,INT S_K,INT S_EVNUM,INT
S_MCBIT,INT _DMODE,INT _AMODE,REAL S_XM,REAL S_YM,INT _DP)
```

Tabelle 3-16 Aufrufparameter CYCLE979 [0)]

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 1 | | S_MVAR | Messvariante | |
| | | | Werte: | EINER: Konturelement<br>1 = Bohrung messen<br>2 = Zapfen (Welle) messen |
| | | | | ZEHNER: reserviert |
| | | | | HUNDERTER: Korrekturziel<br>0 = nur Messen (keine Korrektur der NV bzw. keine Werkzeugkorrektur)<br>1 = Messen und Ermittlung und Korrektur der NV (siehe S_KNUM)<br>2 = Messen und Werkzeugkorrektur (siehe S_KNUM1) |
| | | | | TAUSENDER: Anzahl der Messpunkte<br>0 = 3 Messpunkte<br>1 = 4 Messpunkte |
| | | | | ZEHNTAUSENDER: Messen mit/ohne Spindelumschlag (Differenzmessung) oder Messtaster in Schaltrichtung ausrichten<br>0 = Messen ohne Spindelumschlag, ohne Messtaster ausrichten<br>1 = Messen mit Spindelumschlag<br>2 = Messtaster in Schaltrichtung ausrichten |
| 2 | Auswahl | S_KNUM | Korrektur Nullpunktverschiebung (NV) oder Basis NV oder Basisbezug [2)] | |
| | | | Werte: | EINER: |
| | | | | ZEHNER: 0 = keine Korrektur<br>1 bis max. 99 Nummer der Nullpunktverschiebung oder<br>1 bis max. 16 Nummer der Basisverschiebung |
| | | | | HUNDERTER: reserviert |
| | | | | TAUSENDER: Korrektur NV oder Basis oder Basisbezug<br>0 = Korrektur einstellbare NV<br>1 = Korrektur kanalspezifische Basis NV<br>2 = Korrektur Basisbezug<br>3 = Korrektur globale Basis NV<br>9 = Korrektur aktive NV bzw. bei G500 in letzte aktive kanalspezifische Basis NV |
| | | | | ZEHNTAUSENDER: Korrektur grob oder fein in NV, Basis NV oder Basisbezug<br>0 = Korrektur fein [6)]<br>1 = Korrektur grob |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 3 | Auswahl | S_KNUM1 | Korrektur in Werkzeugkorrektur [2] | |
| | | | Werte: | EINER: |
| | | | | ZEHNER: |
| | | | | HUNDERTER:<br>0 = keine Korrektur<br>1 bis max. 999 D-Nummer (Schneidennummer) bei Werkzeugkorrektur; bei Summen- und Einrichtekorrektur siehe auch S_DLNUM |
| | | | | TAUSENDER: 0 oder eindeutige D-Nummern |
| | | | | ZEHNTAUSENDER: 0 oder eindeutige D-Nummern<br>1 bis max. 32000 wenn eindeutige D-Nummern in MDs eingerichtet sind |
| | | | | HUNDERTTAUSENDER: Werkzeugkorrektur [2]<br>0 = keine Angabe (Korrektur in Werkzeugradius)<br>1 = Korrektur Länge L1<br>2 = Korrektur Länge L2<br>3 = Korrektur Länge L3<br>4 = Korrektur Radius |
| | | | | EINEMILLION: Werkzeugkorrektur [2]<br>0 = keine Angabe (Korrektur des Werkzeugradius-Verschleiß)<br>1 = Werkzeugkorrektur Summenkorrektur (SK) [5]<br>Werkzeugkorrekturwert wird zur bestehenden SK addiert<br>2 = Werkzeugkorrektur Einrichtekorrektur (EK) [5]<br>EK (neu) = EK (alt) + SK (alt) Korrekturwert, SK (neu) = 0<br>3 = Werkzeugkorrektur Einrichtekorrektur (EK) [5]<br>Werkzeugkorrekturwert wird zur bestehenden EK addiert<br>4 = Werkzeugkorrektur Geometrie |
| | | | | ZEHNMILLION: Werkzeugkorrektur [2]<br>0 = keine Angabe (Korrektur in Werkzeuggeometrie normal, nicht invertiert)<br>1 = Korrektur invertiert |
| | | | | EINHUNDERTMILLIONEN: Werkzeugkorrektur<br>0 = Werkzeugkorrektur ohne Duplo-Werkzeuge<br>1 = Werkzeugkorrektur in Duplo-Werkzeug (_DP) |
| 4 | icon+ Anzahl | S_PRNUM | Nummer des Feldes der Messtasterparameter (nicht Messtasternummer)<br>(Wertebereich 1 bis 40) | |
| 5 | X0 | S_SETV | Sollwert | |
| 6 | DFA | S_FA | Messweg | |
| 7 | TSA | S_TSA | Vertrauensbereich | |
| 8 | X0 | S_CPA | Mittelpunkt der 1. Achse der Ebene (bei G17 X) | |
| 9 | Y0 | S_CPO | Mittelpunkt der 2. Achse der Ebene (bei G17 Y) | |
| 10 | XM | S_XM | Sollwert-Mittelpunktvorgabe Geometrieachse X | |
| 11 | YM | S_YM | Sollwert-Mittelpunktvorgabe Geometrieachse Y | |
| 12 | alpha 0 | S_STA1 | Startwinkel [7] | |
| 13 | alpha 1 | S_INCA | Fortschaltwinkel [8] | |
| 14 | Messungen | S_NMSP | Anzahl Messungen am selben Ort [1] (Wertebereich 1 bis 9) | |
| 15 | ST | _DP | Nummer des zu korrigierenden Schwesterwerkzeugs (Duplo-Nummer) | |
| 16 | T | S_TNAME | Werkzeugname [2] | |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 17 | DL | S_DLNUM | Einrichte- Summenkorrektur DL-Nummer [1), 4)] | |
| 18 | TZL | S_TZL | Nullkorrektur [1), 2)] | |
| 19 | DIF | S_TDIF | Maßdifferenzkontrolle [1), 2)] | |
| 20 | TUL | S_TUL | Toleranzobergrenze (inkrementell zum Sollwert) [2)] | |
| 21 | TLL | S_TLL | Toleranzuntergrenze (inkrementell zum Sollwert) [2)] | |
| 22 | TMV | S_TMV | Korrekturbereich bei Mittelwertbildung [1)] | |
| 23 | FW | S_K | Wichtungsfaktor für Mittelwertbildung [1)] | |
| 24 | | S_EVNUM | Datensatz Erfahrungswertspeicher [1), 6)] | |
| 25 | | S_MCBIT | reserviert | |
| 26 | | _DMODE | Displaymode | |
| | | | Werte: | EINER: Bearbeitungsebene G17/G18/G19<br>0 = Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv<br>1 = G17 (nur im Zyklus aktiv)<br>2 = G18 (nur im Zyklus aktiv)<br>3 = G19 (nur im Zyklus aktiv) |
| 27 | | _AMODE | Alternativmode | |
| | | | Werte: | EINER: Maßtoleranz ja/nein<br>0 = nein<br>1 = ja |

0) Alle Defaultwerte = 0 oder als Wertebereich a bis b gekennzeichnet.

1) Anzeige abhängig vom allgemeinen SD54760 $SNS_MEA_FUNCTION_MASK_PIECE

2) nur bei Korrektur in Werkzeug, ansonsten Parameter = ""

3) nur bei Korrektur in Werkzeug und Maßtoleranz "ja", ansonsten Parameter = 0

4) nur wenn Funktion "Einrichte-Summenkorrektur" im allgemeinen MD 18108 $MN_MM_NUM_SUMCORR eingerichtet ist.

5) wenn NV "fein" in MDs nicht eingerichtet ist, dann wird nach NV "grob" korrigiert

6) Erfahrungs-Mittelwertbildung nur bei Werkzeugkorrektur möglich
Wertbereich der Erfahrungs-Mittelwertspeicher:
1 bis 20 Nummer (n) des Erfahrungswertspeichers, siehe kanalspezifisches SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1]
10000 bis 200000 Nummer (n) des Mittelwertspeichers, siehe kanalspezifisches SD55625 $SCS_MEA_AVERAGE_VALUE[n-1]

7) Wertebereich Startwinkel -360 bis +360 Grad

8) Wertebereich Fortschaltwinkel >0 bis ≤90 Grad bei 4 Messpunkten bzw. >0 bis ≤120 Grad bei 3 Messpunkten.

### 3.25.3.10 Messzyklenparameter CYCLE997

```
PROC CYCLE997 (INT S_MVAR,INT S_KNUM,INT S_PRNUM,REAL S_SETV,REAL S_FA,REAL S_TSA,REAL
S_STA1,REAL S_INCA,REAL S_SETV0,REAL S_SETV1,REAL S_SETV2,REAL S_SETV3,REAL S_SETV4,REAL
S_SETV5,REAL S_SETV6,REAL S_SETV7,REAL S_SETV8,REAL S_TNVL,INT S_NMSP,INT S_MCBIT,INT
_DMODE,INT _AMODE)
```

Tabelle 3-17 Aufrufparameter CYCLE997 [1), 2)]

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 1 | | S_MVAR | Messvariante (default =9) | |
| | | | Werte: | EINER: Konturelement<br>9 = Kugel messen |
| | | | | ZEHNER: Messwiederholung<br>0 = ohne Messwiederholung<br>1 = mit Messwiederholung |
| | | | | HUNDERTER: Korrekturziel<br>0 = nur Messen (keine Korrektur der NV)<br>1 = Messen und Ermittlung und Korrektur der NV (siehe S_KNUM) |
| | | | | TAUSENDER: Messstrategie<br>0 = Messen achsparallel, ohne Startwinkel, Messtaster-Ausrichtung entsprechend SD55740, Bit 1<br>1 = Messen umkreisen, mit Startwinkel, Messtaster-Ausrichtung entsprechend SD55740, Bit 1<br>2 = Messen umkreisen, mit Startwinkel, Messtaster in Schaltrichtung ausrichten<br>3 = Messen achsparallel, mit Startwinkel, Messtaster-Ausrichtung entsprechend SD55740, Bit 1<br>4 = Messen achsparallel, mit Startwinkel, Messtaster in Schaltrichtung ausrichten |
| | | | | ZEHNTAUSENDER: Anzahl der zu messenden Kugeln<br>0 = eine Kugel messen<br>1 = Drei Kugeln messen |
| | | | | HUNDERTTAUSENDER: Anzahl der Messpunkte, nur bei Messen unter Winkel (beachte Messstrategie: TAUSENDER-Stelle > 0)<br>0 = Drei Messpunkte beim Messen unter Winkel (Umkreisen der Kugel)<br>1 = Vier Messpunkte beim Messen unter Winkel (Umkreisen der Kugel) |
| | | | | EINEMILLION: Bestimmung des Solldurchmessers der Kugel<br>0 = keine Bestimmung des Solldurchmessers der Kugel<br>1 = Bestimmung des Solldurchmessers der Kugel |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 2 | Auswahl | S_KNUM | Korrektur in Nullpunktverschiebung (NV) oder Basis oder Basisbezug [3] | |
| | | | Werte: | EINER: |
| | | | | ZEHNER:<br>0 = keine Korrektur<br>1 bis max. 99 Nummer der Nullpunktverschiebung oder<br>1 bis max. 16 Nummer der Basisverschiebung |
| | | | | HUNDERTER: reserviert |
| | | | | TAUSENDER: Korrektur in NV oder Basis NV oder Basisbezug<br>0 = Korrektur in einstellbare NV<br>1 = Korrektur in kanalspezifische Basis NV<br>2 = Korrektur in Basisbezug<br>3 = Korrektur in globale Basis NV [7]<br>9 = Korrektur in aktive NV bzw. bei G500 in letzte aktive kanalspezifische Basis NV |
| | | | | ZEHNTAUSENDER: Korrektur in NV oder Basis NV oder Basisbezug grob oder fein<br>0 = Korrektur fein [6]<br>1 = Korrektur grob |
| 3 | icon+ Anzahl | S_PRNUM | Nummer des Feldes der Messtasterparameter (nicht Messtasternummer) (Wertebereich 1 bis 40) | |
| 4 | | S_SETV | Durchmesser der Kugel(n) [4] | |
| 5 | DFA | S_FA | Messweg | |
| 6 | TSA | S_TSA | Vertrauensbereich | |
| 7 | alpha 0 | S_STA1 | Startwinkel beim Messen unter Winkel | |
| 8 | alpha 1 | S_INCA | Fortschaltwinkel beim Messen unter Winkel | |
| 9 | X1 | S_SETV0 | Sollposition der 1. Kugel der 1. Achse der Ebene (bei G17 X) beim Messen 3 Kugeln | |
| 10 | Y1 | S_SETV1 | Sollposition der 1. Kugel der 2. Achse der Ebene (bei G17 Y) beim Messen 3 Kugeln | |
| 11 | Z1 | S_SETV2 | Sollposition der 1. Kugel der 3. Achse der Ebene (bei G17 Z) beim Messen 3 Kugeln | |
| 12 | X2 | S_SETV3 | Sollposition der 2. Kugel der 1. Achse der Ebene beim Messen 3 Kugeln | |
| 13 | Y2 | S_SETV4 | Sollposition der 2. Kugel der 2. Achse der Ebene beim Messen 3 Kugeln | |
| 14 | Z2 | S_SETV5 | Sollposition der 2. Kugel der 3. Achse der Ebene beim Messen 3 Kugeln | |
| 15 | X3 | S_SETV6 | Sollposition der 3. Kugel der 1. Achse der Ebene beim Messen 3 Kugeln | |
| 16 | Y3 | S_SETV7 | Sollposition der 3. Kugel der 2. Achse der Ebene beim Messen 3 Kugeln | |
| 17 | Z3 | S_SETV8 | Sollposition der 3. Kugel der 3. Achse der Ebene beim Messen 3 Kugeln | |
| 18 | TVL | S_TNVL | Grenzwert für Verzerrung des Dreiecks (Summe der Abweichungen) beim Messen 3 Kugeln [5] | |
| 19 | Messungen | S_NMSP | Anzahl Messungen am selben Ort [2] (Wertebereich 1 bis 9) | |
| 20 | | S_MCBIT | reserviert | |
| 21 | | _DMODE | Displaymode | |
| | | | Werte: | EINER: Bearbeitungsebene G17/G18/G19<br>0 = Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv<br>1 = G17 (nur im Zyklus aktiv)<br>2 = G18 (nur im Zyklus aktiv)<br>3 = G19 (nur im Zyklus aktiv) |
| 22 | | _AMODE | Alternativmode | |

[1] alle Defaultwerte = 0 oder als Wertebereich a bis b gekennzeichnet

2) Anzeige abhängig vom allgemeinen SD54760 $SNS_MEA_FUNCTION_MASK_PIECE

3) Zwischenpositionierung mit Umkreisen der Kugel am Äquator

4) Messen 3 Kugeln: Es gilt für alle Kugeln der gleiche Solldurchmesser (_SETV)

5) Defaultwert für `S_TNVL=1.2`
Korrektur in NV: Nur wenn die ermittelte Verzerrung unter dem Grenzwert `S_TNVL` liegt, wird in die NV korrigiert.

6) wenn NVP "fein" in MDs nicht eingerichtet ist, dann wird nach NV "grob" korrigiert

7) Bei der Messvariante "Drei Kugeln messen" ist eine Korrektur in ein globales Basisframe nicht möglich (`S_KNUM` = 3001 bis 3016), da das Frame keine Rotationskomponente besitzt.

### 3.25.3.11 Messzyklenparameter CYCLE995

```
PROC CYCLE995 (INT S_MVAR,INT S_KNUM,INT S_PRNUM,REAL S_SETV,REAL S_FA,REAL S_TSA,REAL
S_STA1,REAL S_INCA,REAL S_DZ,REAL S_SETV0,REAL S_SETV1,REAL S_SETV2,REAL S_TUL,REAL
S_TZL,INT S_NMSP,INT S_MCBIT,INT _DMODE,INT _AMODE)
```

Tabelle 3-18 Aufrufparameter CYCLE995 1)

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 1 | | `S_MVAR` | Messvariante (default=5) | |
| | | | Werte: | EINER: Konturelement<br>5 = Spindelgeometrie (Parallelität zur Werkzeugachse) |
| | | | | ZEHNER: Messwiederholung<br>1 = mit Messwiederholung |
| | | | | HUNDERTER: kein Korrekturziel<br>0 = nur Messen |
| | | | | TAUSENDER: Messstrategie<br>2 = Messen unter Winkel, Messtaster in Schaltrichtung ausrichten |
| | | | | ZEHNTAUSENDER: Anzahl der zu messenden Kugeln<br>0 = eine Kugel messen |
| | | | | HUNDERTTAUSENDER: Anzahl der Messpunkte<br>1 = 4 Messpunkte beim Messen unter Winkel (Umkreisen der Kugel) |
| | | | | EINEMILLION: Bestimmung des Solldurchmessers der Kugel<br>0 = keine Bestimmung des Solldurchmessers der Kugel<br>1 = Bestimmung des Solldurchmessers der Kugel |
| 2 | Auswahl | `S_KNUM` | Korrekturziel<br>0 = 0 | |
| 3 | icon+ Anzahl | `S_PRNUM` | Nummer des Feldes der Messtasterparameter (nicht Messtasternummer)<br>(Wertebereich 1 bis 40) | |
| 4 | DM | `S_SETV` | Durchmesser der Kalibrierkugel 4) | |
| 5 | DFA | `S_FA` | Messweg | |
| 6 | TSA | `S_TSA` | Vertrauensbereich 5) | |
| 7 | alpha 0 | `S_STA1` | Startwinkel beim Messen unter Winkel 3) | |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 8 | | S_INCA | Fortschaltwinkel beim Messen unter Winkel [2] | |
| 9 | DZ | S_DZ | Abstand 1. Messung P1 zur 2. Messung P2 am Schaft des Messtasters | |
| 10 | | S_SETV0 | Sollposition der Kugel der 1. Achse der Ebene (bei G17 X) [2] | |
| 11 | | S_SETV1 | Sollposition der Kugel der 2. Achse der Ebene (bei G17 Y) [2] | |
| 12 | | S_SETV2 | Sollposition der Kugel der 3. Achse der Ebene (bei G17 Z) [2] | |
| 13 | TUL | S_TUL | Toleranzobergrenze der Winkelabweichung | |
| 14 | TZL | S_TZL | Nullkorrekturbereich [1, 4] | |
| 15 | Anzahl | S_NMSP | Anzahl Messungen am selben Ort [2] (Wertebereich 1 bis 9) | |
| 16 | | S_MCBIT | reserviert [2] | |
| 17 | | _DMODE | Displaymode | |
| | | | Werte: | EINER: Bearbeitungsebene G17/G18/G19<br>0 = Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv<br>1 = G17 (nur im Zyklus aktiv)<br>2 = G18 (nur im Zyklus aktiv)<br>3 = G19 (nur im Zyklus aktiv) |
| 18 | | _AMODE | Alternativmode | |
| | | | Werte: | EINER: Maßtoleranz ja/nein<br>0 = nein<br>1 = ja |

alle Defaultwerte = 0 oder als Wertebereich a bis b gekennzeichnet

1) Anzeige abhängig vom allgemeinen SD54760 $SNS_MEA_FUNCTION_MASK_PIECE.

2) Parameter werden z. Z. nicht genutzt und auch nicht in der Eingabemaske angezeigt.
Der Parameter Fortschaltwinkel S_INCA ist auf 90 Grad fest eingestellt.

3) Wertebereich Startwinkel -360 bis +360 Grad

4) bei Maßtoleranz ja:
Sind die gemessenen Winkel kleiner als der Wert des Nullkorrekturbereiches TZL, werden die Ergebnisparameter für die Winkel (_OVR[2], _OVR[3]) und die Abweichungen (_OVR[7], _OVR[8]) gleich Null gesetzt.
Anzeige TZL erfolgt über das allgemeine SD54760 $SNS_MEA_FUNCTION_MASK_PIECE Bit25=1.
(Auswahl Nullkorrektur freigeben bei messen Winkligkeit Spindel)

5) Parameter TSA bezieht sich auf die 1. Messung der Kalibrierkugel.

### 3.25.3.12 Messzyklenparameter CYCLE996

```
PROC CYCLE996(INT S_MVAR,INT S_TC,INT S_PRNUM,REAL S_SETV,REAL S_STA1,REAL S_SETV0,REAL
S_SETV1,REAL S_SETV2,REAL S_SETV3,REAL S_SETV4,REAL S_SETV5,REAL S_TNVL,REAL S_FA,REAL
S_TSA,INT S_NMSP,INT S_MCBIT,INT _DMODE,INT _AMODE)
```

Tabelle 3-19 Aufrufparameter CYCLE996 [1)]

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 1 | | S_MVAR | Messvariante (default=1) | |
| | | | Werte: | EINER: Messreihenfolge<br>0 = Kinematik berechnen (Auswahl mit: Ergebnisanzeige, Protokoll, Änderung des Schwenkdatensatzes, ggf. mit Bedienerquittung), siehe _AMODE<br>1 = 1. Messung<br>2 = 2. Messung<br>3 = 3. Messung |
| | | | | ZEHNER: reserviert<br>0 = 0 |
| | | | | HUNDERTER: Messvariante bei 1. bis 3. Messung<br>0 = Messen der Kalibrierkugel achsparallel<br>1 = Messen der Kalibrierkugel unter Winkel und kein Nachführen Spindel [3)]<br>2 = Messen der Kalibrierkugel und Nachführen der Spindel in Schaltrichtung des Messtasters [3)]<br>3 = Messen achsparallel, mit Startwinkel [8)]<br>4 = Messen achsparallel, mit Startwinkel, Nachführen Spindel in Schaltrichtung des Messtasters [8)] |
| | | | | TAUSENDER: Korrekturziel bei Kinematik berechnen [4)]<br>0 = nur Messen. Schwenkdatensätze werden berechnet aber bleiben unverändert<br>1 = Schwenkdatensatz berechnen. Schwenkdatensätze werden, ggf. nach Bedienerquittung, geändert [4)] |
| | | | | ZEHNTAUSENDER: Messachse (Rundachse 1 oder 2) oder Vektorkette offen oder geschlossen bei Kinematik berechnen<br>0 = Vektorkette geschlossen (nur bei Kinematik berechnen)<br>1 = Rundachse 1 (nur bei 1. bis 3. Messung)<br>2 = Rundachse 2 (nur bei 1. bis 3. Messung) [5)]<br>3 = Vektorkette offen (nur bei Kinematik berechnen) |
| | | | | HUNDERTTAUSENDER: Normierung Rundachse 1 bei Kinematik berechnen<br>0 = keine Normierung Rundachse 1<br>1 = Normierung in Richtung 1. Achse der Ebene (bei G17 X)<br>2 = Normierung in Richtung 2. Achse der Ebene (bei G17 Y)<br>3 = Normierung in Richtung 3. Achse der Ebene (bei G17 Z) |
| | | | | EINEMILLION: Normierung Rundachse 2 bei Kinematik berechnen [5)]<br>0 = keine Normierung Rundachse 2<br>1 = Normierung in Richtung 1. Achse der Ebene (bei G17 X)<br>2 = Normierung in Richtung 2. Achse der Ebene (bei G17 Y)<br>3 = Normierung in Richtung 3. Achse der Ebene (bei G17 Z) |
| | | | | ZEHNMILLION: Protokollfile<br>0 = keine Protokollfile<br>1 = Protokollfile mit den berechneten Vektoren (Toolcarrier) und der 1. dynamischen 5-Achstransformation (TRAORI(1)), falls in MDs eingerichtet. |
| 2 | | S_TC | Nummer des Schwenkdatensatzes (Toolcarrier) | |
| 3 | icon+ Anzahl | S_PRNUM | Nummer des Feldes der Messtasterparameter (nicht Messtasternummer) (default=1) | |
| 4 | | S_SETV | Durchmesser der Kalibrierkugel | |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 5 | alpha 0 | S_STA1 | Startwinkel beim Messen unter Winkel | |
| 6 | alpha 0 | S_SETV0 | Positionswert Rundachse 1 (wenn Rundachse manuell oder halbautomatisch) | |
| 7 | alpha 1 | S_SETV1 | Positionswert Rundachse 2 (wenn Rundachse manuell oder halbautomatisch) [6] | |
| 8 | XN | S_SETV2 | Positionswert für Normierung Rundachse 1 | |
| 9 | XN | S_SETV3 | Positionswert für Normierung Rundachse 2 [6] | |
| 10 | delta | S_SETV4 | Toleranzwert der Offsetvektoren I1 bis I4 | |
| 11 | delta | S_SETV5 | Toleranzwert der Rundachsvektoren V1 und V2 | |
| 12 | TVL | S_TNVL | Grenzwert des Winkelsegments der Rundachse (Wertebereich 1 bis 60 Grad), (default=20) [7] | |
| 13 | DFA | S_FA | Messweg | |
| 14 | TSA | S_TSA | Vertrauensbereich | |
| 15 | Messungen | S_NMSP | Anzahl Messungen am selben Ort [2] (default=1) | |
| 16 | | S_MCBIT | reserviert | |
| 17 | | _DMODE | Displaymode | |
| | | | Werte: | EINER: Bearbeitungsebene G17/G18/G19<br>0 = Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv<br>1 = G17 (nur im Zyklus aktiv)<br>2 = G18 (nur im Zyklus aktiv)<br>3 = G19 (nur im Zyklus aktiv) |
| 18 | | _AMODE | Alternativmode | |
| | | | Werte: | EINER: Toleranzkontrolle ja/nein<br>0 = nein<br>1 = ja: Auswertung der Toleranzwerte der Vektoren S_SETV4, S_SETV5 |
| | | | | ZEHNER: Bedienerquittung beim Eintragen der berechneten Vektoren in den Schwenkdatensatz [4]<br>0 = ja: Bediener muss Änderung quittieren<br>1 = nein: berechnete Vektoren werden sofort eingetragen (nur wirksam, wenn HUNDERTER und TAUSENDER-Stelle = 0) |
| | | | | HUNDERTER: Messergebnisanzeige [5]<br>0 = nein<br>1 = ja |
| | | | | TAUSENDER: Messergebnisanzeige editierbar<br>0 = nein<br>1 = ja, und editierbar (nur wirksam, wenn HUNDERTER-Stelle = 1) |

[1] alle Defaultwerte = 0 oder als default=x gekennzeichnet

[2] Anzeige abhängig vom allgemeinen SD54760 $SNS_MEA_FUNCTION_MASK_PIECE.

[3] Mit dieser Variante kann z. B. bei 90 Grad-Stellungen der Kinematik an der Kalibrierkugel gemessen werden, ohne mit dem Befestigungsschaft der Kalibrierkugel zu kollidieren. Ein Startwinkel S_STA1 (0 bis 360 Grad) kann vorgegeben werden. Der Fortschaltwinkel beim Umkreisen der Kugel ist gleich 90 Grad.
Als Vorschub auf der Kreisbahn wird das kanalspezifische SD55634 $SCS_MEA_FEED_PLANE_VALUE verwendet

[4] Vor dem Eintragen erfolgt eine Bedienerabfrage mit M0. Erst mit NC-Start werden die Vektoren eingetragen.
Wird das Messprogramm mit RESET abgebrochen, so werden keine berechneten Vektoren eingetragen.
Vektoren werden nur eingetragen, wenn die Toleranz der Offsetvektoren bei der Berechnung nicht überschritten ist.

[5] Messergebnisanzeige nur bei der Messvariante Kinematik berechnen.
Soll auch nach der 1. bis 3. Messung das Messergebnis angezeigt werden, erfolgt dies durch Setzen des kanalspezifischen SD55613 $SCS_MEA_RESULT_DISPLAY.

[6] Rundachse 2 nur bei Kinematiken mit zwei Rundachsen

[7] Grenzwert Winkelsegment der Rundachse. Wertebereich von S_TNVL zwischen 20 bis 60 Grad. Bei Werten von S_TNVL < 20 Grad ist mit Ungenauigkeiten, bedingt durch die Messungenauigkeiten im Mikrometerbereich des Messtasters, zu rechnen. Ist der Grenzwert überschritten, erfolgt die Fehlermeldung 61430 mit Anzeige des Minimalwertes für den Grenzwert.

[8] Nachführen Spindel in Schaltrichtung des Messtasters, wenn SD54760 Bit 17 = 1

### 3.25.3.13 Messzyklenparameter CYCLE9960

```
PROC CYCLE9960(INT S_MVAR,STRING[40] S_TNAME,INT S_PRNUM,REAL S_SETV,REAL S_SETV1,REAL
S_START_RA1,REAL S_END_RA1,INT S_CMEA_RA1,REAL S_POS_RA2,REAL S_SETV2,REAL
S_START_RA2,REAL S_END_RA2,INT S_CMEA_RA2,REAL S_POS_RA1,REAL S_SETV4,REAL S_FA,REAL
S_TSA,INT S_NMSP,INT S_DMODE,INT S_AMODE,INT S_KNUM)
```

Tabelle 3-20 Aufrufparameter CYCLE9960 [1]

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 1 | Auswahl | S_MVAR | Messvariante (default=1) | |
| | | | Werte: | EINER: Messvariante<br>0 = Kinematik messen und berechnen (Auswahl mit: Ergebnisanzeige, Protokoll, Änderung des Schwenkdatensatzes (siehe tausender S_MVAR))<br>1 = Referenzkopf vermessen<br>2 = Kopf an Referenzkopf anpassen<br>3 = Stützpunkte messen und berechnen (E996) |
| | | | | ZEHNER: reserviert |
| | | | | HUNDERTER: Messvariante bei Kugelmessung<br>2 = Messen der Kalibrierkugel und Nachführen der Spindel in Schaltrichtung des Messtasters<br>4 = Messen achsparallel, mit Startwinkel, Nachführen Spindel in Schaltrichtung des Messtasters [2] |
| | | | | TAUSENDER: Korrekturziel bei Kinematik berechnen [3]<br>0 = Messen und berechnen. Datensätze werden berechnet und bleiben unverändert<br>1 = Berechnete Datensätze werden, ggf. nach Bedienerquittung, geändert<br>2 = Bereits vermessene Kinematik wird berechnet und ggf. nach Bedienerquittung geändert |
| | | | | ZEHNTAUSENDER:Messachse (Rundachse 1 oder 2)<br>1 = Alle vorhandenen Rundachsen vermessen und berechnen<br>4 = Nur Rundachse 1 vermessen und berechnen<br>5 = Nur Rundachse 2 vermessen und berechnen |
| 2 | | S_TNAME | Name der Trafo (Schwenkdatensatz oder Trafo auf Basis kinematischer Ketten) | |
| 3 | icon+ Anzahl | S_PRNUM | Nummer des Feldes der Messtasterparameter (nicht Messtasternummer) (default=1) | |
| 4 | | S_SETV | Durchmesser der Kalibrierkugel | |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 5 | alpha 1 | S_SETV1 | Startwinkel beim Messen unter Winkel für 1.Rundachse | |
| 6 | | | S_START_RA1 : Startwinkel der 1. Rundachse | |
| 7 | | | S_END_RA1 : Endwinkel der 1. Rundachse | |
| 8 | | | S_CMEA_RA1 : Anzahl der Messungen der 1.Rundachse, 3 für Kinematik Messen und Berechnen. Für Stützpunkte bis zu 12 Messungen möglich. | |
| 9 | | | S_POS_RA2 : Position der 2. Rundachse bei Messung der 1.Rundachse [4] | |
| 10 | alpha 2 | S_SETV2 | Startwinkel beim Messen unter Winkel für 2.Rundachse [4] | |
| 11 | | | S_START_RA2 : Startwinkel der 2.Rundachse [4] | |
| 12 | | | S_END_RA2 : Endwinkel der 2. Rundachse [4] | |
| 13 | | | S_CMEA_RA2 : Anzahl der Messungen der 2. Rundachse, 3 für Kinematik Messen und Berechnen. Für Stützpunkte bis zu 12 Messungen möglich.[4] | |
| 14 | | | S_POS_RA1 : Position der 1. Rundachse bei Messung der 2.Rundachse | |
| 15 | delta | S_SETV4 | Toleranzwert der Offsetvektoren | |
| 16 | DFA | S_FA | Messweg | |
| 17 | TSA | S_TSA | Vertrauensbereich | |
| 18 | Anzahl | S_NMSP | Anzahl Messungen am selben Ort [5]<br>(default=1) | |
| 19 | | _DMODE | Displaymode | |
| | | | Werte: | EINER: Bearbeitungsebene G17/G18/G19<br>0 = Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv<br>1 = G17 (nur im Zyklus aktiv)<br>2 = G18 (nur im Zyklus aktiv)<br>3 = G19 (nur im Zyklus aktiv) |
| 20 | | _AMODE | Alternativmode | |
| | | | Werte: | EINER : Toleranzkontrolle ja/nein<br>0 = nein<br>1 = ja : Auswertung der Toleranzwerte der Vektoren -> S_SETV4, S_SETV5 |
| | | | | ZEHNER : Automatisches Kalibrieren<br>0 = ohne automatisches Kalibrieren<br>1 = Automatisches Kalibrieren |
| | | | | HUNDERTER : Startwinkelautomatik<br>0 = eingestellte Startwinkel wird verwendet<br>1 = automatisierte Berechnung des Startwinkels, bei jedem Messpunkt |
| 21 | | S_KNUM | Nummer der zu korrigierten NPV bei Referenzkopf vermessen | |
| | | | Werte | EINER: Korrektur erfolgt immer in die Grobverschiebung, Feinverschiebung wird gelöscht |
| | | | | ZEHNER : 1..99 Nummer der einstellbaren NPV (1=G54) |
| | | | | TAUSENDER : 9 Korrektur aktive einstellbare NPV |

[1] Alle Defaultwerte = 0 oder als default=xx gekennzeichnet

[2] Nachführen Spindel in Schaltrichtung des Messtasters, wenn SD54760 bit17=1

[3] Vor dem Eintragen erfolgt eine Bedienerabfrage mit M0. Erst mit NC-Start werden die Vektoren eingetragen. Wird das Messprogramm mit RESET abgebrochen, werden keine berechneten Vektoren eingetragen. Vektoren werden nur eingetragen, wenn Toleranz der Offsetvektoren bei der Berechnung nicht überschritten ist.

[4] Rundachse 2 nur bei Kinematiken mit 2 Rundachsen

[5] Anzeige abhängig vom allgemeinen SD54760 $SNS_MEA_FUNCTION_MASK_PIECE.

### 3.25.3.14 Messzyklenparameter CYCLE982

```
PROC CYCLE982(INT S_MVAR,INT S_KNUM,INT S_PRNUM,INT S_MA,INT S_MD,REAL S_ID,REAL S_FA,REAL
S_TSA,REAL S_VMS,REAL S_STA1,REAL S_CORA,REAL S_TZL,REAL S_TDIF,INT S_NMSP,INT S_EVNUM,INT
S_MCBIT,INT _DMODE,INT _AMODE)
```

Tabelle 3-21 Aufrufparameter CYCLE982 [1)]

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 1 | | S_MVAR | Messvariante | |
| | | | Werte: | EINER: Kalibrieren / Messen<br>0 = Werkzeugmesstaster kalibrieren<br>1 = Einzelmessung Werkzeug [3)]<br>2 = Mehrfachmessung Werkzeug, Längen und Werkzeugradius (bei Fräswerkzeugen) bestimmen |
| | | | | ZEHNER: Kalibrieren oder Messen im MKS oder WKS<br>0 = Maschinenbezogen [4)]<br>1 = Werkstückbezogen |
| | | | | HUNDERTER: Messen mit oder ohne Umschlag bei Fräswerkzeugen<br>0 = Messen ohne Umschlag<br>1 = Messen mit Umschlag |
| | | | | TAUSENDER: Korrekturziel bei Fräswerkzeugen<br>0 = Länge oder Länge und Radius bestimmen (siehe S_MVAR 1er-Stelle)<br>1 = Radius bestimmen, wenn S_MVAR 1er-Stelle = 1<br>2 = Länge und Radius bestimmen (Stirnseite), wenn S_MVAR 1er-Stelle = 1 oder 2<br>3 = Scheibenfräser obere Schneide (Rückseite) und Länge und Radius bestimmen [5)] |
| | | | | ZEHNTAUSENDER: Stellung der Fräswerkzeuges oder des Bohrers<br>0 = Axiale Stellung des Fräswerkzeuges oder Bohrers, Radius in 2. Achse der Ebene (bei G18 X) [7)]<br>1 = Radiale Stellung des Fräswerkzeuges oder Bohrers, Radius in 1. Achse der Ebene (bei G18 Z) [7)] |
| | | | | HUNDERTTAUSENDER: inkrementelles Kalibrieren oder Messen<br>0 = keine Angabe<br>1 = inkrementelles Kalibrieren oder Messen |
| | | | | EINEMILLION: Spindel auf Startwinkel S_STA1 positionieren (nur bei Messen Fräswerkzeuge)<br>0 = Spindel nicht positionieren<br>1 = Spindel auf Startwinkel S_STA1 positionieren |
| 2 | Auswahl | S_KNUM | Korrekturvariante [2)] | |
| | | | Werte: | EINER: Werkzeugkorrektur<br>0 = keine Angabe (Werkzeugkorrektur in Geometrie)<br>1 = Werkzeugkorrektur in Verschleiß |
| 3 | icon+ Anzahl | S_PRNUM | Nummer des Feldes der Messtasterparameter (nicht Messtasternummer) (default=1) | |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 4 | X0 | S_MA | Messachse | |
| | | | Werte: | 1 = 1. Achse der Ebene (bei G18 Z)<br>2 = 2. Achse der Ebene (bei G18 X) |
| 5 | +- | S_MD | Messrichtung | |
| | | | Werte: | 0 = keine Auswahl (Messrichtung wird aus Istwert ermittelt)<br>1 = positiv<br>2 = negativ |
| 6 | Z2 | S_ID | Versatz | |
| 7 | DFA | S_FA | Messweg | |
| 8 | TSA | S_TSA | Vertrauensbereich | |
| 9 | VMS | S_VMS | Variable Messgeschwindigkeit bei Kalibrieren [2)] | |
| 10 | alpha1 | S_STA1 | Startwinkel bei Messen von Fräswerkzeugen | |
| 11 | alpha2 | S_CORA | Korrekturwinkel bei Messen mit Umschlag von Fräswerkzeugen [8)] | |
| 12 | TZL | S_TZL | Nullkorrektur beim Vermessen von Werkzeugen. Beim Kalibrieren ist S_TZL = 0 | |
| 13 | DIF | S_TDIF | Maßdifferenzkontrolle | |
| 14 | Messungen | S_NMSP | Anzahl Messungen am selben Ort [2)] (default=1) | |
| 15 | EVN | S_EVNUM | Nummer Erfahrungs- Mittelwertspeicher [2), 9)] | |
| 16 | | S_MCBIT | reserviert | |
| 17 | | _DMODE | Displaymode | |
| | | | Werte: | EINER: Bearbeitungsebene G17/G18/G19<br>0 = Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv<br>1 = G17 (nur im Zyklus aktiv)<br>2 = G18 (nur im Zyklus aktiv)<br>3 = G19 nur im Zyklus aktiv) |
| | | | | ZEHNER: Schneidelage bei Dreh- und Fräswerkzeugen<br>(nur für Anzeige in den Eingabemasken 1 bis 9) |
| | | | | HUNDERTER: Werkzeugtyp<br>0 = Drehwerkzeug<br>1 = Fräser<br>2 = Bohrer |
| | | | | TAUSENDER: Anfahrstrategie in Bezug auf den Werkzeugmesstaster<br>0 = PLUS [X/Z]; X wenn Werkzeuglage axial, Z wenn Werkzeuglage radial<br>1 = MINUS [X/Z]; X wenn Werkzeuglage axial, Z wenn Werkzeuglage radial |
| 18 | | _AMODE | Alternativmode | |
| | | | Werte: | EINER: reserviert |
| | | | | ZEHNER: reserviert |
| | | | | HUNDERTER: reserviert |
| | | | | TAUSENDER: Anfahren Startposition nach Messvorgang bei Kalibrieren und Einzelmessung (siehe S_MVAR - EINER)<br>0 = Werkzeug steht um DFA versetzt gegenüber der Messtasterkante<br>1 = Anfahren Startposition |

[1)] alle Defaultwerte = 0 oder als default=x gekennzeichnet

[2)] Anzeige abhängig vom allgemeinen SD54762 _MEA_FUNCTION_MASK_TOOL

[3] Dreh- oder Fräswerkzeug oder Bohrer messen. Messachse in Parameter `S_MA`
Spezifizierung bei Drehwerkzeugen über Schneidenlage 1...8, bei Fräswerkzeugen über HUNDERTER bis TAUSENDER-Stelle im Parameter `S_MVAR`.

[4] Messen und Kalibrieren erfolgt im Basiskoordinatensystem (MKS bei ausgeschalteter kinematischer Transformation).

[5] nicht für inkrementelles Messen

[6] nur für Mehrfachmessung `S_MVAR=x2x02` oder `x3x02` (Beispiel Scheiben- oder Nutenfräser)

[7] wenn das kanalspezifische SD42950 $SC_TOOL_LENGTH_TYPE = 2, dann erfolgt die Zuordnung der Werkzeuglängenkomponenten wie bei Drehwerkzeugen

[8] nur bei Messen mit Umschlag `S_MVAR=xx1x1`

[9] Erfahrungswertbildung
Wertebereich der Erfahrungswertspeicher: 1 bis 20 Nummer (n) des Erfahrungswertspeichers, siehe kanalspezifisches SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1].

### 3.25.3.15 Messzyklenparameter CYCLE971

```
PROC CYCLE971(INT S_MVAR,INT S_KNUM,INT S_PRNUM,INT S_MA,INT S_MD,REAL S_ID,REAL S_FA,REAL
S_TSA,REAL S_VMS,REAL S_TZL,REAL S_TDIF,INT S_NMSP,REAL S_F1,REAL S_S1,REAL S_F2,REAL
S_S2,REAL S_F3,REAL S_S3,INT S_EVNUM,INT S_MCBIT,INT _DMODE,INT _AMODE)
```

Tabelle 3-22 Aufrufparameter CYCLE971 [1)]

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 1 | | S_MVAR | Messvariante | |
| | | | Werte: | EINER:<br>0 = Werkzeugmesstaster kalibrieren<br>1 = Werkzeug messen mit stehender Spindel (Länge bzw. Radius)<br>2 = Werkzeug messen mit drehender Spindel (Länge bzw. Radius), siehe Parameter S_F1 bis S_S4 |
| | | | | ZEHNER: Messen in MKS oder WKS<br>0 = Messen in MKS (maschinenbezogen), Werkzeug messen oder Werkzeugmesstaster kalibrieren<br>1 = Messen in WKS (werkstückbezogen), Werkzeug messen oder Werkzeugmesstaster kalibrieren |
| | | | | HUNDERTER: Zähne einzeln prüfen<br>0 = nein<br>1 = ja |
| | | | | TAUSENDER:<br>0 = 0 |
| | | | | ZEHNTAUSENDER: Inkrementelles Kalibrieren oder Messen<br>0 = keine Angabe<br>1 = inkrementelles Kalibrieren oder Messen |
| | | | | HUNDERTTAUSENDER: Werkzeugmesstaster automatisch kalibrieren<br>0 = Werkzeugmesstaster nicht automatisch kalibrieren<br>1 = Werkzeugmesstaster automatisch kalibrieren |
| | | | | EINEMILLION: Kalibrieren in der Ebene mit Spindelumschlag<br>0 = Kalibrieren in der Ebene ohne Spindelumschlag<br>1 = Kalibrieren in der Ebene mit Spindelumschlag |
| 2 | Auswahl | S_KNUM | Korrekturvariante [2)] | |
| | | | Werte: | EINER: Werkzeugkorrektur<br>0 = keine Angabe (Werkzeugkorrektur in Geometrie)<br>1 = Werkzeugkorrektur in Verschleiß |
| 3 | icon+ Anzahl | S_PRNUM | Nummer des Feldes der Messtasterparameter (nicht Messtasternummer) | |
| 4 | X0 | S_MA | Messachse, Versetzachse [4)] | |
| | | | Werte: | EINER: Nummer der Messachse<br>1 = 1. Achse der Ebene (bei G17 X)<br>2 = 2. Achse der Ebene (bei G17 Y)<br>3 = 3. Achse der Ebene (bei G17 Z) |
| | | | | ZEHNER:<br>0 = 0 |
| | | | | HUNDERTER: Nummer der Versetzachse<br>0 = keine Versetzachse<br>1 = 1. Achse der Ebene (bei G17 X)<br>2 = 2. Achse der Ebene (bei G17 Y) |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 5 | +- | S_MD | Messrichtung | |
| | | | Werte: | 0 = keine Auswahl (Messrichtung wird aus Istwert ermittelt)<br>1 = positiv<br>2 = negativ |
| 6 | V | S_ID | Versatz | |
| | | | Werte: | 0 = Bei Werkzeugen ohne Versatz<br>>0 =<br>• Kalibrieren: Der Versatz wirkt auf die 3. Achse der Ebene (bei G17 Z), wenn der Durchmesser des Kalibrierwerkzeugs größer als der obere Durchmesser des Messtasters ist. Hier wird das Werkzeug um den Werkzeugradius aus der Mitte des Messtasters versetzt, abzüglich des Wertes von S_ID. Die Versetzachse wird zusätzlich in S_MA angegeben.<br>• Messen: Bei Mehrschneidern ist der Versatz von Werkzeuglänge und höchsten Punkt der Schneide beim Radiusmessen oder der Versatz von Werkzeugradius zu höchsten Punkt der Schneide beim Messen der Werkzeuglänge anzugeben. |
| 7 | DFA | S_FA | Messweg | |
| 8 | TSA | S_TSA | Vertrauensbereich | |
| 9 | VMS | S_VMS | Variable Messgeschwindigkeit beim Kalibrieren [2)] | |
| 10 | TZL | S_TZL | Nullkorrektur (nur bei Messen Werkzeug) | |
| 11 | DIF | S_TDIF | Maßdifferenzkontrolle bei Messen Werkzeug (S_MVAR=xx1 oder S_MVAR=xx2) | |
| 12 | Messungen | S_NMSP | Anzahl Messungen am selben Ort [2)] | |
| 13 | F1 | S_F1 | 1. Vorschub für Antasten mit drehender Spindel [2)] | |
| 14 | S1 | S_S1 | 1. Drehzahl für Antasten mit drehender Spindel [2)] | |
| 15 | F2 | S_F2 | 2. Vorschub für Antasten mit drehender Spindel [2)] | |
| 16 | S2 | S_S2 | 2. Drehzahl für Antasten mit drehender Spindel [2)] | |
| 17 | F3 | S_F3 | 3. Vorschub für Antasten mit drehender Spindel [2)] | |
| 18 | S3 | S_S3 | 3. Drehzahl für Antasten mit drehender Spindel [2)] | |
| 19 | EVN | S_EVNUM | Nummer Erfahrungswertspeicher [2)] | |
| 20 | | S_MCBIT | Maske der _CBITs bzw. _CHBITs | |
| 21 | | _DMODE | Displaymode | |
| | | | Werte: | EINER: Bearbeitungsebene G17/G18/G19<br>0 = Kompatibilität, es bleibt die vor Zyklusaufruf wirksame Ebene aktiv<br>1 = G17 (nur im Zyklus aktiv)<br>2 = G18 (nur im Zyklus aktiv)<br>3 = G19 (nur im Zyklus aktiv) |

| Nr. | Maskenparameter | Zyklenparameter | Bedeutung | |
|---|---|---|---|---|
| 22 | | _AMODE | Alternativmode | |
| | | | Werte: | EINER: Werkzeugversatz beim Radius messen<br>1 = nein<br>2 = ja |
| | | | | ZEHNER: Richtung des Werkzeugversatzes beim Radius messen in der 3. Achse der Ebene (bei G17 Z)<br>1 =positiv<br>2 = negativ |
| | | | | HUNDERTER: Werkzeugversatz beim Länge messen bzw. beim Kalibrieren des Messtasters in der 3. Achse<br>0 = Kompatibilität, auto<br>1 = nein<br>2 = ja |
| | | | | TAUSENDER: Richtung des Werkzeugversatzes beim Länge messen in der Versetzachse (siehe S_MA HUNDERTER)<br>1 = positiv<br>2 = negativ |

1) alle Defaultwerte = 0 oder als default=x gekennzeichnet

2) Anzeige abhängig vom allgemeinen SD54762 MEA_FUNCTION_MASK_TOOL

3) nur bei Korrektur in Werkzeug und Maßtoleranz "ja", ansonsten Parameter = 0

4) beim automatischen Messen (S_MVAR=1x00xx) keine Anzeige Messachse, Versetzachse ⇒ S_MA=0.

### 3.25.3.16 Messzyklenparameter CYCLE150

```
PROC CYCLE150(INT S_PICT,INT S_PROT,STRING[160] S_PATH) SAVE
ACTBLOCNO DISPLOF
```

Tabelle 3-23 Aufrufparameter CYCLE150

| Nr. | Masken-parameter | Zyklen-parameter | Bedeutung | |
|---|---|---|---|---|
| 1 | Messergebnisbild | S_PICT | Auswahl Ergebnisanzeige (Default = 0) | |
| | | | Werte: | EINER:<br>0 = Messergebnisbild AUS<br>1 = Messergebnisbild EIN |
| | | | | Zehner: Auswahl Anzeigemodus (Werte wie SD55613)<br>1 = Messergebnisbild anzeigen - Abwahl automatisch nach 8 s<br>3 = Messergebnisbild anzeigen - Quittieren mit NC-Start<br>4 = Messergebnisbild anzeigen - nur bei den Alarmen (61303 ... 61306) |
| 2 | | S_PROT | Auswahl Protokollieren (Default = 0) | |
| | Protokoll | | Werte: | EINER: Auswahl Protokoll aus / ein / letzte Messung<br>0 = Protokoll AUS<br>1 = Protokoll EIN<br>2 = Protokolliere letzte Messung |

| Nr. | Masken-parameter | Zyklen-parameter | Bedeutung | |
|---|---|---|---|---|
| | Protokolltyp | | | ZEHNER: Auswahl Protokolltyp<br>0 = Standardprotokoll<br>1 = Anwenderprotokoll (frei definierbar) |
| | Protokollformat | | | HUNDERTER: Auswahl Protokollformat<br>0 = Textformat<br>1 = Tabellenformat (für Excel Import) |
| | Protokolldaten | | | TAUSENDER: Auswahl neu schreiben oder anhängen<br>0 = neu<br>1 = anhängen |
| | Protokoll ablage | | | ZEHNTAUSENDER: Auswahl Protokollablage<br>0 = wie Teileprogramm<br>1 = Verzeichnis |
| 3 | | S_PATH | Pfad für Protokolldatei entsprechend Auswahl Protokollablage<br>(komplette Pfadangabe oder nur Dateiname, z. B.:<br>"//NC:/WKS.DIR/NAME.WPD oder "MESSPROTOKOLL.TXT" | |

# Tabellen

# 4

## 4.1 Anweisungen

**Hinweis**

**Zyklen**

Die Liste der Anweisungen enthält alle Zyklen, die im NC-Programm (G-Code) vorkommen können, d. h. im Programmeditor über Masken programmiert werden können bzw. beim Schleifen ohne Programmierunterstützung programmiert werden müssen. Nicht berücksichtigt wurden Zyklen, die aus Kompatibilitätsgründen in der Steuerung noch vorhanden sind, aber über den Programmeditor von SINUMERIK Operate nicht mehr editiert werden können ("Kompatibilitätszyklen").

### Anweisungen A ... C

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| : | O | NC-Hauptsatznummer, Sprungmarkenabschluss, Kettungsoperator | | + | | PM-NC |
| * | O | Operator für Multiplikation | | + | | PM-NC |
| + | O | Operator für Addition | | + | | PM-NC |
| - | O | Operator für Subtraktion | | + | | PM-NC |
| < | O | Vergleichsoperator, kleiner | | + | | PM-NC |
| << | O | Verkettungsoperator für Strings | | + | | PM-NC |
| <= | O | Vergleichsoperator, kleiner gleich | | + | | PM-NC |
| = | O | Zuweisungsoperator | | + | | PM-NC |
| >= | O | Vergleichsoperator, größer gleich | | + | | PM-NC |
| / | O | Operator für Division | | + | | PM-NC |
| /0<br>…<br>…<br>/7 | | Satz wird ausgeblendet (1. Ausblendebene)°<br>...<br>...<br>Satz wird ausgeblendet (8. Ausblendebene) | | + | | PM-NC |
| A | A | Achsname | m/s | + | | PM-NC |
| A2 | A | Werkzeugorientierung: RPY- oder Eulerwinkel | s | + | | PM-NC |
| A3 | A | Werkzeugorientierung: 1. Komponente des Richtungsvektors | s | + | | PM-NC |
| A4 | A | Werkzeugorientierung: 1. Komponente des Flächennormalenvektors am Satzanfang | s | + | | PM-NC |
| A5 | A | Werkzeugorientierung: 1. Komponente des Flächennormalenvektors am Satzende | s | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| A6 | A | Werkzeugorientierung: 1. Komponente des Richtungsvektors für die Drehachse des Kegels | s | + | | PM-NC |
| A7 | A | Werkzeugorientierung: 1. Vektorkomponente für die Zwischenorientierung auf der Kegelmantelfläche | s | + | | PM-NC |
| ABS | F | Absolutwert (Betrag) | | + | + | PM-NC |
| AC | K | absolute Maßangabe von Koordinaten/Positionen | s | + | | PM-NC |
| ACC | K | Beeinflussung der aktuellen axialen Beschleunigung | m | + | + | PM-NC |
| ACCLIMA | K | Beeinflussung der aktuellen maximalen axialen Beschleunigung | m | + | + | PM-NC |
| ACN | K | absolute Maßangabe für Rundachsen, Position in negativer Richtung anfahren | s | + | | PM-NC |
| ACOS | F | Arcus-Cosinus<br>(Trigon. Funktion) | | + | + | PM-NC |
| ACP | K | absolute Maßangabe für Rundachsen, Position in positiver Richtung anfahren | s | + | | PM-NC |
| ACTBLOCNO | P | Ausgabe der aktuellen Satznummer eines Alarmsatzes, auch wenn "aktuelle Satzanzeige unterdrückt" (DISPLOF) aktiv ist! | | + | | PM-NC |
| ADDFRAME | F | Einrechnung und evtl. Aktivierung eines gemessenen Frames | | + | - | PM-NC, FM-B |
| ADIS | A | Überschleifabstand für Bahnfunktionen G1, G2, G3, ... | m | + | | PM-NC |
| ADISPOS | A | Überschleifabstand für Eilgang G0 | m | + | | PM-NC |
| ADISPOSA | P | Größe des Toleranzfenster für IPOBRKA | m | + | + | PM-NC |
| ALF | A | Schnellabhebewinkel | m | + | | PM-NC |
| AMIRROR | G | Programmierbare Spiegelung | s | + | | PM-NC |
| AND | K | Logisches UND | | + | | PM-NC |
| ANG | A | Konturzug-Winkel | s | + | | PM-NC |
| AP | A | Polarwinkel | m/s | + | | PM-NC |
| APR | K | Zugriffsschutz lesen / anzeigen | | + | | PM-NC |
| APRB | K | Zugriffsrecht lesen, BTSS | | + | | PM-NC |
| APRP | K | Zugriffsrecht lesen, Teileprogramm | | + | | PM-NC |
| APW | K | Zugriffsschutz schreiben | | + | | PM-NC |
| APWB | K | Zugriffsrecht schreiben, BTSS | | + | | PM-NC |
| APWP | K | Zugriffsrecht schreiben, Teileprogramm | | + | | PM-NC |
| APX | K | Definition des Zugriffsschutzes für die Ausführung des angegebenen Sprachelements | | + | | PM-NC |
| AR | A | Öffnungswinkel | m/s | + | | PM-NC |
| AROT | G | Programmierbare Drehung | s | + | | PM-NC |
| AROTS | G | Programmierbare Framedrehungen mit Raumwinkeln | s | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| AS | K | Makro-Definition | | + | | PM-NC |
| ASCALE | G | Programmierbare Skalierung | s | + | | PM-NC |
| ASIN | F | Rechenfunktion, Arcussinus | | + | + | PM-NC |
| ASPLINE | G | Akima-Spline | m | + | | PM-NC |
| ATAN2 | F | Arcus-Tangens2 | | + | + | PM-NC |
| ATOL | A | Achsspezifische Toleranz für Kompressor-Funktionen, Orientierungsglättung und Überschleifarten | m | + | | PM-NC |
| ATRANS | G | Additive programmierbare Nullpunktverschiebung | s | + | | PM-NC |
| AUXFUDEL | P | Hilfsfunktion kanalspezifisch aus der globalen Liste löschen | | + | - | FM-B |
| AUXFUDELG | P | Alle Hilfsfunktionen einer Hilfsfunktionsgruppe kanalspezifisch aus der globalen Liste löschen | | + | - | FM-B |
| AUXFUMSEQ | P | Ausgabe-Reihenfolge für M-Hilfsfunktionen ermitteln | | + | - | FM-B |
| AUXFUSYNC | P | Aus der globalen Liste der Hilfsfunktionen einen vollständigen Teileprogrammsatz für das kanalspezifische SERUPRO-Ende-ASUP als String generieren | | + | - | FM-B |
| AX | K | Variabler Achsbezeichner | m/s | + | | PM-NC |
| AXCTSWE | P | Achscontainer drehen | | + | - | PM-NC |
| AXCTSWEC | P | Freigabe zur Achscontainer-Drehung zurücknehmen | | + | + | PM-NC |
| AXCTSWED | P | Achscontainer drehen (Befehlsvariante für die Inbetriebnahme!) | | + | - | PM-NC |
| AXIS | K | Achsbezeichner, Achsadresse | | + | | PM-NC |
| AXNAME | F | Konvertiert Eingangsstring in Achsbezeichner | | + | - | PM-NC |
| AXSTRING | F | Konvertiert den String Spindelnummer | | + | - | PM-NC |
| AXTOCHAN | P | Achse für einen bestimmten Kanal anfordern. Ist vom NC-Programm und aus Synchronaktion möglich. | | + | + | PM-NC |
| AXTOSPI | F | konvertiert Achsbezeichner in einen Spindelindex um | | + | - | PM-NC |
| B | A | Achsname | m/s | + | | PM-NC |
| B2 | A | Werkzeugorientierung: RPY- oder Eulerwinkel | s | + | | PM-NC |
| B3 | A | Werkzeugorientierung: . Komponente des Richtungsvektors | s | + | | PM-NC |
| B4 | A | Werkzeugorientierung: 2. Komponente des Flächennormalenvektors am Satzanfang | s | + | | PM-NC |
| B5 | A | Werkzeugorientierung: 2. Komponente des Flächennormalenvektors am Satzende | s | + | | PM-NC |
| B6 | A | Werkzeugorientierung: 2. Komponente des Richtungsvektors für die Drehachse des Kegels | s | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| B7 | A | Werkzeugorientierung: 2. Vektorkomponente für die Zwischenorientierung auf der Kegelmantelfläche | s | + | | PM-NC |
| B_AND | O | Bitweises UND | | + | | PM-NC |
| B_OR | O | Bitweises ODER | | + | | PM-NC |
| B_NOT | O | Bitweise Negierung | | + | | PM-NC |
| B_XOR | O | Bitweises Exklusiv-ODER | | + | | PM-NC |
| BAUTO | G | Definieren des ersten Spline-Abschnitts durch die nachfolgenden 3 Punkte | m | + | | PM-NC |
| BLOCK | K | Definiert zusammen mit dem Schlüsselwort TO den abzuarbeitenden Programmteil in einem indirekten Unterprogrammlauf | | + | | PM-NC |
| BLSYNC | K | Bearbeitung der Interruptroutine soll erst mit dem nächsten Satzwechsel beginnen | | + | | PM-NC |
| BNAT [6] | G | Natürlicher Übergang zum ersten Spline-Satz | m | + | | PM-NC |
| BOOL | K | Datentyp: Wahrheitswerte TRUE/FALSE bzw. 1/0 | | + | | PM-NC |
| BOUND | F | Prüft, ob Wert innerhalb des definierten Wertebereichs liegt. Gleichheit gibt Prüfwert zurück. | | + | + | PM-NC |
| BRISK [6] | G | Sprungförmige Bahnbeschleunigung | m | + | | PM-NC |
| BRISKA | P | Sprungförmige Bahnbeschleunigung für die programmierten Achsen einschalten | | + | - | PM-NC |
| BSPLINE | G | B-Spline | m | + | | PM-NC |
| BTAN | G | Tangentialer Übergang zum ersten Spline-Satz | m | + | | PM-NC |
| C | A | Achsname | m/s | + | | PM-NC |
| C2 | A | Werkzeugorientierung: RPY- oder Eulerwinkel | s | + | | PM-NC |
| C3 | A | Werkzeugorientierung: 3. Komponente des Richtungsvektors | s | + | | PM-NC |
| C4 | A | Werkzeugorientierung: 3. Komponente des Flächennormalenvektors am Satzanfang | s | + | | PM-NC |
| C5 | A | Werkzeugorientierung: 3. Komponente des Flächennormalenvektors am Satzende | s | + | | PM-NC |
| C6 | A | Werkzeugorientierung: 3. Komponente des Richtungsvektors für die Drehachse des Kegels | s | + | | PM-NC |
| C7 | A | Werkzeugorientierung: 3. Vektorkomponente für die Zwischenorientierung auf der Kegelmantelfläche | s | + | | PM-NC |
| CAC | K | Absolutes Anfahren einer Position | | + | | PM-NC |
| CACN | K | In Tabelle abgelegter Wert wird absolut in negativer Richtung angefahren | | + | | PM-NC |
| CACP | K | In Tabelle abgelegter Wert wird absolut in positiver Richtung angefahren | | + | | PM-NC |
| CADAPTOF | P | Lastanpassung ausschalten | | + | - | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| CADAPTON | P | Lastanpassung einschalten | | + | - | PM-NC |
| CALCDAT | F | Berechnet Radius und Mittelpunkt eines Kreises aus 3 oder 4 Punkten | | + | - | PM-NC |
| CALCFIR | P | FIR-Ruckfilter an Dynamikmodus anpassen | | + | - | PM-NC |
| CALCPOSI | F | Überprüfung auf Schutzbereichsverletzung, Arbeitsfeldbegrenzung und Softwarelimits | | + | - | PM-NC |
| CALL | K | Indirekter Unterprogrammaufruf | | + | | PM-NC |
| CALLPATH | P | Programmierbarer Suchpfad bei Unterprogrammaufrufen | | + | - | PM-NC |
| CANCEL | P | Modale Synchronaktion abbrechen | | + | - | FM-SA |
| CANCELSUB | P | Aktuelle Unterprogrammebene abbrechen | | + | + | FM-SA |
| CASE | K | Bedingte Programmverzweigung | | + | | PM-NC |
| CDC | K | Direktes Anfahren einer Position | | + | | PM-NC |
| CDOF [6] | G | Kollisionsüberwachung ausschalten | m | + | | PM-NC |
| CDOF2 | G | Kollisionsüberwachung ausschalten, bei 3D-Umfangsfräsen | m | + | | PM-NC |
| CDON | G | Kollisionsüberwachung einschalten | m | + | | PM-NC |
| CFC [6] | G | Konstanter Vorschub an der Kontur | m | + | | PM-NC |
| CFIN | G | Konstanter Vorschub nur bei Innenkrümmung, nicht bei Außenkrümmung | m | + | | PM-NC |
| CFINE | F | Zuweisung der Fein-Verschiebung an eine FRAME-Variable | | + | - | PM-NC |
| CFTCP | G | Konstanter Vorschub im Werkzeugschneiden-Be-zugspunkt, Mittelpunkts-bahn | m | + | | PM-NC |
| CHAN | K | Spezifizierung des Gültigkeitsbereichs von Daten | | + | | PM-NC |
| CHANDATA | P | Kanalnummer für Kanaldatenzugriffe einstellen | | + | - | PM-NC |
| CHAR | K | Datentyp: ASCII-Zeichen | | + | | PM-NC |
| CHF | A | Fase;<br>Wert = Länge der Fase | s | + | | PM-NC |
| CHKDM | F | Prüfung der Eindeutigkeit innerhalb eines Magazins | | + | - | FM-TM |
| CHKDNO | F | Eindeutigkeitsprüfung der D-Nummern | | + | - | PM-NC |
| CHR | A | Fase;<br>Wert = Länge der Fase in Bewegungsrichtung | | + | | PM-NC |
| CIC | K | Inkrementelles Anfahren einer Position | | + | | PM-NC |
| CIP | G | Kreisinterpolation über Zwischenpunkt | m | + | | PM-NC |
| CLEARM | P | Rücksetzen einer/mehrerer Marken für Kanalkoordinierung | | + | + | PM-NC |
| CLRINT | P | Interrupt abwählen | | + | - | PM-NC |
| CMIRROR | F | Spiegeln an einer Koordinatenachse | | + | - | PM-NC |
| COARSEA | K | Bewegungsende beim Erreichen von "Genauhalt Grob" | m | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| COLLPAIR | F | Prüfen auf Zugehörigkeit zu einem Kollisionspaar | | + | | PM-NC |
| COMPCAD | G | Kompressor-Funktion COMPCAD einschalten | m | + | | PM-NC |
| COMPCURV | G | Kompressor-Funktion COMPCURV einschalten | m | + | | PM-NC |
| COMPLETE | | Steueranweisung für das Aus- und Einlesen von Daten | | + | | PM-NC |
| COMPOF [6] | G | NC-Satz-Kompression ausschalten | m | + | | PM-NC |
| COMPON | G | Kompressor-Funktion COMPON einschalten | m | + | | PM-NC |
| COMPSURF | G | Kompressor-Funktion COMPSURF einschalten | m | + | | PM-NC |
| CONTDCON | P | Konturdecodierung in Tabellenform einschalten | | + | - | PM-NC |
| CONTPRON | P | Referenzaufbereitung einschalten | | + | - | PM-NC |
| CORROF | P | Alle aktiven überlagerten Bewegungen werden abgewählt. | | + | - | PM-NC |
| CORRTC | F | Offsetvektoren oder Richtungsvektoren orientierbarer Werkzeugträger nach Maschinenvermessung modifizieren. | | + | - | PM-NC |
| CORRTRAFO | F | Offsetvektoren oder Richtungsvektoren der Orientierungsachsen im kinematischen Modell der Maschine modifizieren | | + | - | PM-NC |
| COS | F | Cosinus<br>(Trigon. Funktion) | | + | + | PM-NC |
| COUPDEF | P | Definition ELG-Verband / Synchronspindel-Verband | | + | - | PM-NC |
| COUPDEL | P | ELG-Verband löschen | | + | - | PM-NC |
| COUPOF | P | ELG-Verband / Synchronspindelpaar ausschalten | | + | - | PM-NC |
| COUPOFS | P | Ausschalten ELG-Verband / Synchronspindelpaar mit Stopp der Folgespindel | | + | - | PM-NC |
| COUPON | P | ELG-Verband / Synchronspindelpaar einschalten | | + | - | PM-NC |
| COUPONC | P | Einschalten ELG-Verband / Synchronspindelpaar mit vorhergehender Programmierung übernehmen | | + | - | PM-NC |
| COUPRES | P | ELG-Verband rücksetzen | | + | - | PM-NC |
| CP [6] | G | Bahnbewegung | m | + | | PM-NC |
| CPBC | K | Generische Kopplung: Satzwechselkriterium | | + | + | FM-A |
| CPDEF | K | Generische Kopplung: Anlegen eines Koppelmoduls | | + | + | FM-A |
| CPDEL | K | Generische Kopplung: Löschen eines Koppelmoduls | | + | + | FM-A |
| CPFMOF | K | Generische Kopplung: Verhalten der Folgeachse beim vollständigen Ausschalten | | + | + | FM-A |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| CPFMON | K | Generische Kopplung: Verhalten der Folgeachse beim Einschalten | | + | + | FM-A |
| CPFMSON | K | Generische Kopplung: Synchronisationsmodus | | + | + | FM-A |
| CPFPOS | K | Generische Kopplung: Synchronposition der Folgeachse | | + | + | FM-A |
| CPFRS | K | Generische Kopplung: Koordinatenbezugssystem | | + | + | FM-A |
| CPLA | K | Generische Kopplung: Definition einer Leitachse | | + | - | FM-A |
| CPLCTID | K | Generische Kopplung: Nummer der Kurventabelle | | + | + | FM-A |
| CPLDEF | K | Generische Kopplung: Definition einer Leitachse und Anlegen eines Koppelmoduls | | + | + | FM-A |
| CPLDEL | K | Generische Kopplung: Löschen einer Leitachse eines Koppelmoduls | | + | + | FM-A |
| CPLDEN | K | Generische Kopplung: Nenner des Koppelfaktors | | + | + | FM-A |
| CPLINSC | K | Generische Kopplung: Skalierfaktor für den Eingangswert einer Leitachse | | + | + | FM-A |
| CPLINTR | K | Generische Kopplung: Verschiebewert für den Eingangswert einer Leitachse | | + | + | FM-A |
| CPLNUM | K | Generische Kopplung: Zähler des Koppelfaktors | | + | + | FM-A |
| CPLOF | K | Generische Kopplung: Ausschalten einer Leitachse eines Koppelmoduls | | + | + | FM-A |
| CPLON | K | Generische Kopplung: Einschalten einer Leitachse eines Koppelmoduls | | + | + | FM-A |
| CPLOUTSC | K | Generische Kopplung: Skalierfaktor für den Ausgangswert einer Kopplung | | + | + | FM-A |
| CPLOUTTR | K | Generische Kopplung: Verschiebewert für den Ausgangswert einer Kopplung | | + | + | FM-A |
| CPLPOS | K | Generische Kopplung: Synchronposition der Leitachse | | + | + | FM-A |
| CPLSETVAL | K | Generische Kopplung: Kopplungsbezug | | + | + | FM-A |
| CPMALARM | K | Generische Kopplung: Unterdrückung spezieller kopplungsbezogener Alarmausgaben | | + | + | FM-A |
| CPMBRAKE | K | Generische Kopplung: Verhalten der Folgeachse bei bestimmten Stoppsignalen und -kommandos | | + | - | FM-A |
| CPMPRT | K | Generische Kopplung: Kopplungsverhalten beim Teileprogrammstart unter Satzsuchlauf via Programmtest | | + | + | FM-A |
| CPMRESET | K | Generische Kopplung: Kopplungsverhalten bei RESET | | + | + | FM-A |
| CPMSTART | K | Generische Kopplung: Kopplungsverhalten beim Teileprogrammstart | | + | + | FM-A |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| CPMVDI | K | Generische Kopplung: Verhalten der Folgeachse auf bestimmte NC/PLC-Nahtstellensignale | | + | + | FM-A |
| CPOF | K | Generische Kopplung: Ausschalten eines Koppelmoduls | | + | + | FM-A |
| CPON | K | Generische Kopplung: Einschalten eines Koppelmoduls | | + | + | FM-A |
| CPRECOF [6] | G | Programmierbare Konturgenauigkeit ausschalten | m | + | | PM-NC |
| CPRECON | G | Programmierbare Konturgenauigkeit einschalten | m | + | | PM-NC |
| CPRES | K | Generische Kopplung: Aktiviert die projektierten Daten der Synchronspindelkopplung | | + | - | FM-A |
| CPROT | P | Kanalspezifischer Schutzbereich ein-/ausschalten | | + | - | PM-NC |
| CPROTDEF | P | Definition eines kanalspezifischen Schutzbereichs | | + | - | PM-NC |
| CPSETTYPE | K | Generische Kopplung: Kopplungstyp | | + | + | FM-A |
| CPSYNCOP | K | Generische Kopplung: Schwellwert für den Positionssynchronlauf "Grob" | | + | + | FM-A |
| CPSYNCOP2 | K | Generische Kopplung: Schwellwert für den Positionssynchronlauf "Grob" 2 | | + | + | FM-A |
| CPSYNCOV | K | Generische Kopplung: Schwellwert für den Geschwindigkeitssynchronlauf "Grob" | | + | + | FM-A |
| CPSYNFIP | K | Generische Kopplung: Schwellwert für den Positionssynchronlauf "Fein" | | + | + | FM-A |
| CPSYNFIP2 | K | Generische Kopplung: Schwellwert für den Positionssynchronlauf "Fein" 2 | | + | + | FM-A |
| CPSYNFIV | K | Generische Kopplung: Schwellwert für den Geschwindigkeitssynchronlauf "Fein" | | + | + | FM-A |
| CR | A | Kreisradius | s | + | | PM-NC |
| CROT | F | Drehung des aktuellen Koordinatensystems | | + | - | PM-NC |
| CROTS | F | Programmierbare Framedrehungen mit Raumwinkeln (Drehung in den angegebenen Achsen) | s | + | - | PM-NC |
| CRPL | F | Frame-Drehung in einer beliebigen Ebene | | + | - | FM-B |
| CSCALE | F | Maßstabsfaktor für mehrere Achsen | | + | - | PM-NC |
| CSPLINE | F | Kubischer Spline | m | + | | PM-NC |
| CT | G | Kreis mit tangentialem Übergang | m | + | | PM-NC |
| CTAB | F | Ermittle Folgeachsposition anhand der Leitachsposition aus Kurventabelle | | + | + | PM-NC |
| CTABDEF | P | Tabellendefinition einschalten | | + | - | PM-NC |
| CTABDEL | P | Kurventabelle löschen | | + | - | PM-NC |
| CTABEND | P | Tabellendefinition ausschalten | | + | - | PM-NC |
| CTABEXISTS | F | Prüft die Kurventabelle mit der Nummer n | | + | + | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| CTABFNO | F | Anzahl der noch möglichen Kurventabellen im Speicher | | + | + | PM-NC |
| CTABFPOL | F | Anzahl der noch möglichen Polynome im Speicher | | + | + | PM-NC |
| CTABFSEG | F | Anzahl der noch möglichen Kurvensegmente im Speicher | | + | + | PM-NC |
| CTABID | F | Liefert Tabellen-Nummer der n-ten Kurventabelle | | + | + | PM-NC |
| CTABINV | F | Ermittle Leitachsposition anhand der Folgeachsposition aus Kurventabelle | | + | + | PM-NC |
| CTABISLOCK | F | Gibt den Sperrzustand der Kurventabelle mit der Nummer n zurück | | + | + | PM-NC |
| CTABLOCK | P | Löschen und Über-schreiben, sperren | | + | + | PM-NC |
| CTABMEMTYP | F | Gibt den Speicher zurück, in dem die Kurventabelle mit der Nummer n angelegt ist. | | + | + | PM-NC |
| CTABMPOL | F | Anzahl der maximal möglichen Polynome im Speicher | | + | + | PM-NC |
| CTABMSEG | F | Anzahl der maximal möglichen Kurvensegmente im Speicher | | + | + | PM-NC |
| CTABNO | F | Anzahl der definierten Kurventabellen im SRAM oder DRAM | | + | + | FM-A |
| CTABNOMEM | F | Anzahl der definierten Kurventabellen im SRAM oder DRAM | | + | + | PM-NC |
| CTABPERIOD | F | Gibt die Tabellenperiodizität der Kurventabelle mit der Nummer n zurück | | + | + | PM-NC |
| CTABPOL | F | Anzahl der bereits verwendeten Polynome im Speicher | | + | + | PM-NC |
| CTABPOLID | F | Anzahl der von der Kurventabelle mit der Nummer n verwendeten Kurvenpolynome | | + | + | PM-NC |
| CTABSEG | F | Anzahl der bereits verwendeten Kurvensegmente im Speicher | | + | + | PM-NC |
| CTABSEGID | F | Anzahl der von der Kurventabelle mit der Nummer n verwendeten Kurvensegmente | | + | + | PM-NC |
| CTABSEV | F | Liefert den Endwert der Folgeachse eines Segments der Kurventabelle | | + | + | PM-NC |
| CTABSSV | F | Liefert den Startwert der Folgeachse eines Segments der Kurventabelle | | + | + | PM-NC |
| CTABTEP | F | Liefert den Wert der Leitachse am Kurventabellen-Ende | | + | + | PM-NC |
| CTABTEV | F | Liefert den Wert der Folgeachse am Kurventabellen-Ende | | + | + | PM-NC |
| CTABTMAX | F | Liefert Maximalwert der Folgeachse der Kurventabelle | | + | + | PM-NC |
| CTABTMIN | F | Liefert Minimalwert der Folgeachse der Kurventabelle | | + | + | PM-NC |
| CTABTSP | F | Liefert den Wert der Leitachse am Kurventabellen-Anfang | | + | + | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| CTABTSV | F | Liefert den Wert der Folgeachse am Kurventabellen-Anfang | | + | + | PM-NC |
| CTABUNLOCK | P | Aufheben der Lösch- und Überschreibsperre | | + | + | PM-NC |
| CTOL | A | Konturtoleranz für Kompressor-Funktionen, Orientierungsglättung und Überschleifarten | m | + | - | PM-NC |
| CTOLG0 | A | Konturtoleranz bei Eilgangbewegungen | m | + | - | PM-NC |
| CTRANS | F | Nullpunktverschiebung für mehrere Achsen | | + | - | PM-NC |
| CUT2D [6] | G | 2D-WRK | m | + | | PM-NC |
| CUT2DD | G | Auf ein Differenzwerkzeug bezogene 2½-D-WRK | m | + | | PM-NC |
| CUT2DF | G | 2D-WRK, relativ zum aktuellen Frame (schräge Ebene) | m | + | | PM-NC |
| CUT2DFD | G | Auf ein Differenzwerkzeug bezogene 2½-D-WRK, relativ zum aktuellen Frame (schräge Ebene) | m | + | | PM-NC |
| CUT3DC | G | 3D-WRK für das Umfangsfräsen | m | + | | PM-NC |
| CUT3DCC | G | 3D-WRK für das Umfangsfräsen unter Berücksichtigung einer Begrenzungsfläche mit 3D-Radiuskorrektur: Kontur an der Bearbeitungsfläche | m | + | | PM-NC |
| CUT3DCCD | G | 3D-WRK für das Umfangsfräsen unter Berücksichtigung einer Begrenzungsfläche mit Differenzwerkzeug auf der Werkzeugmittelpunktbahn: Zustellung zur Begrenzungsfläche | m | + | | PM-NC |
| CUT3DCD | G | Auf ein Differenzwerkzeug bezogene 3D-WRK für das Umfangsfräsen | m | + | | PM-NC |
| CUT3DF | G | 3D-WRK für das Stirnfräsen mit Orientierungsänderung | m | + | | PM-NC |
| CUT3DFD | G | Auf ein Differenzwerkzeug bezogene 3D-WRK für das Stirnfräsen mit Orientierungsänderung | m | + | | PM-NC |
| CUT3DFF | G | 3D-WRK für das Stirnfräsen mit konstanter Orientierung. Die Werkzeugorientierung ist die durch G17 - G19 festgelegte und gegebenenfalls durch einen Frame gedrehte Richtung. | m | + | | PM-NC |
| CUT3DFS | G | 3D-WRK für das Stirnfräsen mit konstanter Orientierung. Die Werkzeugorientierung ist durch G17 - G19 festgelegt und wird durch Frames nicht beeinflusst. | m | + | | PM-NC |
| CUTCONOF [6] | G | Konstante Radiuskorrektur ausschalten | m | + | | PM-NC |
| CUTCONON | G | Konstante Radiuskorrektur einschalten | m | + | | PM-NC |
| CUTMOD | A | Modifikation der Korrekturdaten bei drehbaren Werkzeugen einschalten (in Verbindung mit orientierbaren Werkzeugträgern) | m | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| CUTMODK | A | Modifikation der Korrekturdaten bei drehbaren Werkzeugen einschalten (in Verbindung mit Orientierungstransformationen, die mittels kinematischer Ketten definiert wurden) | m | + | | PM-NC |
| CYCLE60 | C (T) | Gravurzyklus | | + | | PM-NC |
| CYCLE61 | C (T) | Planfräsen | | + | | PM-NC |
| CYCLE62 | C (T) | Konturaufruf | | + | | PM-NC |
| CYCLE63 | C (T) | Konturtasche fräsen | | + | | PM-NC |
| CYCLE64 | C (T) | Konturtasche vorbohren | | + | | PM-NC |
| CYCLE70 | C (T) | Gewindefräsen | | + | | PM-NC |
| CYCLE72 | C (T) | Bahnfräsen | | + | | PM-NC |
| CYCLE76 | C (T) | Rechteckzapfen fräsen | | + | | PM-NC |
| CYCLE77 | C (T) | Kreiszapfen fräsen | | + | | PM-NC |
| CYCLE78 | C (T) | Bohrgewinde fräsen | | + | | PM-NC |
| CYCLE79 | C (T) | Mehrkant | | + | | PM-NC |
| CYCLE81 | C (T) | Bohren, Zentrieren | | + | | PM-NC |
| CYCLE82 | C (T) | Bohren, Plansenken | | + | | PM-NC |
| CYCLE83 | C (T) | Tieflochbohren | | + | | PM-NC |
| CYCLE84 | C (T) | Gewindebohren ohne Ausgleichsfutter | | + | | PM-NC |
| CYCLE85 | C (T) | Reiben | | + | | PM-NC |
| CYCLE86 | C (T) | Ausdrehen | | + | | PM-NC |
| CYCLE92 | C (T) | Abstich | | + | | PM-NC |
| CYCLE95 | C (T) | Konturabspanen | | + | | PM-NC |
| CYCLE98 | C (T) | Gewindekette | | + | | PM-NC |
| CYCLE99 | C (T) | Gewindedrehen | | + | | PM-NC |
| CYCLE116 | C (M) | Berechnung von Mittelpunkt und Radius eines Kreises | | + | | PM-MC |
| CYCLE119 | C (M) | Bestimmung der räumlichen Lage | | + | | PM-MC |
| CYCLE150 | C (M) | Messergebnisse anzeigen / protokollieren | | + | | PM-MC |
| CYCLE435 | C (T) | Abrichterposition berechnen | | + | | PM-NC |
| CYCLE495 | C (T) | Profilieren | | + | | PM-NC |
| CYCLE750 | C (A) | Interner Arbeitszyklus für CYCLE751 ... CYCLE759 (enthält das MMC-Kommando für den eigentlichen Funktionsaufruf) | | - | | FM-A |
| CYCLE751 | C (A) | Optimierungssitzung öffnen / ausführen / schließen | | M | | FM-A |
| CYCLE752 | C (A) | Achse zu einer Optimierungssitzung hinzufügen | | M | | FM-A |
| CYCLE753 | C (A) | Optimierungsmodus auswählen | | M | | FM-A |
| CYCLE754 | C (A) | Datensatz hinzufügen / entfernen | | M | | FM-A |
| CYCLE755 | C (A) | Datensatz sichern / wiederherstellen | | M | | FM-A |
| CYCLE756 | C (A) | Optimierungsergebnisse aktivieren | | M | | FM-A |
| CYCLE757 | C (A) | Optimierungsdaten speichern | | M | | FM-A |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| CYCLE758 | C (A) | Parameterwert ändern | | M | | FM-A |
| CYCLE759 | C (A) | Parameterwert lesen | | M | | FM-A |
| CYCLE782 | C (T) | An Beladung anpassen | | + | | PM-NC |
| CYCLE800 | C (T) | Schwenken | | + | | PM-NC |
| CYCLE801 | C (T) | Gitter oder Rahmen | | + | | PM-NC |
| CYCLE802 | C (T) | Beliebige Positionen | | + | | PM-NC |
| CYCLE830 | C (T) | Tieflochbohren 2 | | + | | PM-NC |
| CYCLE832 | C (T) | High Speed Settings | | + | | PM-NC |
| CYCLE840 | C (T) | Gewindebohren mit Ausgleichsfutter | | + | | PM-NC |
| CYCLE899 | C (T) | Offene Nut fräsen | | + | | PM-NC |
| CYCLE930 | C (T) | Einstich | | + | | PM-NC |
| CYCLE940 | C (T) | Freistich Formen | | + | | PM-NC |
| CYCLE951 | C (T) | Abspanen | | + | | PM-NC |
| CYCLE952 | C (T) | Konturstechen | | + | | PM-NC |
| CYCLE961 | C (M) | Lage einer Werkstückecke (innen oder außen) bestimmen und als Nullpunktverschiebung einsetzen | | + | | PM-MC |
| CYCLE971 | C (M) | Werkzeugmesstaster kalibrieren, Werkzeuglänge und/oder Werkzeugradius messen (nur für Technologie Fräsen) | | + | | PM-MC |
| CYCLE973 | C (M) | Werkstückmesstaster an einer Fläche am Werkstück oder in einer Nut kalibrieren (nur für Technologie Drehen) | | + | | PM-MC |
| CYCLE974 | C (M) | Werkstücknullpunkt in der gewählten Messachse bestimmen, Werkzeugkorrektur mit 1-Punkt-Messung bestimmen (nur für Technologie Drehen) | | + | | PM-MC |
| CYCLE976 | C (M) | Werkstückmesstaster in einem Kalibrierring bzw. an einer Kalibrierkugel vollständig in der Arbeitsebene oder an einer Kante für eine bestimmte Achse und Richtung kalibrieren | | + | | PM-MC |
| CYCLE977 | C (M) | Mittelpunkt in der Ebene sowie Breite bzw. Durchmesser bestimmen | | + | | PM-MC |
| CYCLE978 | C (M) | Position einer Kante im Werkstückkoordinatensystem messen | | + | | PM-MC |
| CYCLE979 | C (M) | Mittelpunkt in der Ebene bestimmen, Radius von Kreissegmenten messen | | + | | PM-MC |
| CYCLE982 | C (M) | Werkzeugmesstaster kalibrieren, Dreh-, Bohr- und Fräswerkzeuge vermessen (nur für Technologie Drehen) | | + | | PM-MC |
| CYCLE994 | C (M) | Werkstücknullpunk in der gewählten Messachse mit 2-Punkt-Messung bestimmen (nur für Technologie Drehen) | | + | | PM-MC |
| CYCLE995 | C (M) | Winkligkeit der Spindel an einer Werkzeugmaschine messen | | + | | PM-MC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| CYCLE996 | C (M) | Transformationsrelevante Daten für kinematische Transformationen mit Rundachsen bestimmen | | + | | PM-MC |
| CYCLE997 | C (M) | Mittelpunkt und Durchmesser einer Kugel bestimmen, Mittelpunkte von drei verteilten Kugeln messen | | + | | PM-MC |
| CYCLE998 | C (M) | Winkellage einer Fläche (Ebene) bezogen auf die Arbeitsebene bestimmen, Winkel von Kanten im Werkstückkoordinatensystem bestimmen | | + | | PM-MC |
| CYCLE4071 | C (T) | Längschleifen mit Zustellung am Umkehrpunkt | | + | | PM-NC |
| CYCLE4072 | C (T) | Längschleifen mit Zustellung am Umkehrpunkt und Abbruchsignal | | + | | PM-NC |
| CYCLE4073 | C (T) | Längschleifen mit kontinuierlicher Zustellung | | + | | PM-NC |
| CYCLE4074 | C (T) | Längschleifen mit kontinuierlicher Zustellung und Abbruchsignal | | + | | PM-NC |
| CYCLE4075 | C (T) | Flachschleifen mit Zustellung am Umkehrpunkt | | + | | PM-NC |
| CYCLE4077 | C (T) | Flachschleifen mit Zustellung am Umkehrpunkt und Abbruchsignal | | + | | PM-NC |
| CYCLE4078 | C (T) | Flachschleifen mit kontinuierlicher Zustellung | | + | | PM-NC |
| CYCLE4079 | C (T) | Flachschleifen mit intermittierender Zustellung | | + | | PM-NC |
| CYCLE9960 | C (M) | Kinematik komplett vermessen | | + | | PM-MC |

## Anweisungen D ... F

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| D | A | Werkzeugkorrekturnummer | | + | | PM-NC |
| D0 | A | Bei D0 sind die Korrekturen für das Werkzeug unwirksam | | + | | PM-NC |
| DAC | K | Absolut satzweise achsspezifische Durchmesserprogrammierung | s | + | | PM-NC |
| DC | K | Absolute Maßangabe für Rundachsen, Position direkt anfahren | s | + | | PM-NC |
| DCI | K | Datenklasse I (= Individual) zuordnen (nur SINUMERIK 828D!) | | + | | PM-NC |
| DCM | K | Datenklasse M (= Manufacturer) zuordnen (nur SINUMERIK 828D!) | | + | | PM-NC |
| DCU | K | Datenklasse U (= Anwender) zuordnen (nur SINUMERIK 828D!) | | + | | PM-NC |
| DEF | K | Variablendefinition | | + | | PM-NC |
| DEFAULT | K | Zweig in der CASE-Verzweigung | | + | | PM-NC |
| DEFINE | K | Schlüsselwort für Makrodefinitionen | | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1][2][3][4][5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| DELAYFSTOF | P | Ende eines Stopp-Delay-Bereichs definieren | m | + | - | PM-NC |
| DELAYFSTON | P | Beginn eines Stopp-Delay-Bereichs definieren | m | + | - | PM-NC |
| DELDL | F | Additive Korrekturen löschen | | + | - | PM-NC |
| DELDTG | P | Restweglöschen | | - | + | FM-SA |
| DELETE | P | Die angegebene Datei löschen. Der Dateiname kann mit Pfad und Datei-Kennung angegeben werden. | | + | - | PM-NC |
| DELMLOWNER | F | Eigentümermagazinplatz des Werkzeugs löschen | | + | - | FM-TM |
| DELMLRES | F | Magazinplatz-Reservierung löschen | | + | - | FM-TM |
| DELMT | P | Multitool löschen | | + | - | FM-TM |
| DELOBJ | F | Löschen von Elementen von kinematischen Ketten, Schutzbereichen, Schutzbereichselementen, Kollisionspaaren und Transformationsdaten | | + | | PM-NC |
| DELT | P | Werkzeug löschen | | + | - | FM-TM |
| DELTC | P | Werkzeugträgerdatensatz löschen | | + | - | FM-TM |
| DELTOOLENV | F | Datensätze zur Beschreibung von Werkzeugumgebungen löschen | | + | - | PM-NC |
| DIACYCOFA | K | Achsspezifische modale Durchmesser-programmierung: AUS in Zyklen | m | + | | FM-A |
| DIAM90 | G | Durchmesserprogrammierung für G90, Radiusprogrammierung für G91 | m | + | | PM-NC |
| DIAM90A | K | Achsspezifische modale Durchmesserprogrammierung für G90 und AC, Radiusprogrammierung für G91 und IC | m | + | | PM-NC |
| DIAMCHAN | K | Übernahme aller Achsen aus MD Achsfunktionen in den Kanalzustand der Durchmesserprogrammierung | | + | | PM-NC |
| DIAMCHANA | K | Übernahme Kanalzustand der Durchmesserprogrammierung | | + | | PM-NC |
| DIAMCYCOF | G | Kanalspezifische Durchmesserprogrammierung: AUS in Zyklen | m | + | | FM-A |
| DIAMOF [6] | G | Durchmesserprogrammierung: AUS<br>Grundstellung siehe Maschinenhersteller | m | + | | PM-NC |
| DIAMOFA | K | Achsspezifische modale Durchmesserprogrammierung: AUS<br>Grundstellung siehe Maschinenhersteller | m | + | | PM-NC |
| DIAMON | G | Durchmesserprogrammierung: EIN | m | + | | PM-NC |
| DIAMONA | K | Achsspezifische modale Durchmesserprogrammierung: EIN<br>Freischaltung siehe Maschinenhersteller | m | + | | PM-NC |
| DIC | K | Relativ satzweise achsspezifische Durchmesserprogrammierung | s | + | | PM-NC |
| DILF | A | Rückzugsweg (Länge) | m | + | | PM-NC |
| DISABLE | P | Interrupt AUS | | + | - | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1][2][3][4][5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| DISC | A | Überhöhung Übergangskreis Werkzeug-Radiuskorrektur | m | + | | PM-NC |
| DISCL | A | Abstand des Endpunkts der schnellen Zustellbe-wegung, von der Bearbeitungsebene | | + | | PM-NC |
| DISPLOF | PA | Aktuelle Satzanzeige unterdrücken | | + | | PM-NC |
| DISPLON | PA | Unterdrückung der aktuellen Satzanzeige aufheben | | + | | PM-NC |
| DISPR | A | Repos-Bahndifferenz | s | + | | PM-NC |
| DISR | A | Repos-Abstand | s | + | | PM-NC |
| DISRP | A | Abstand der Rückzugsebene von der Bearbeitungsebene beim weichen An- und Abfahren | | + | | PM-NC |
| DITE | A | Gewindeauslaufweg | m | + | | PM-NC |
| DITS | A | Gewindeeinlaufweg | m | + | | PM-NC |
| DIV | K | Integer-Division | | + | | PM-NC |
| DL | A | Ortsabhängige additive Werkzeugkorrektur an-wählen (DL, Summen- Einrichtekorrektur) | m | + | | PM-NC |
| DO | K | Synchronaktion: Auslösen von Aktionen bei erfüllter Bedingung | | - | + | FM-SA |
| DRFOF | P | Ausschalten der Handradverschiebungen (DRF) | m | + | - | PM-NC |
| DRIVE | G | Geschwindigkeitsabhängige Bahnbeschleunigung | m | + | | PM-NC |
| DRIVEA | P | Geknickte Beschleunigungskennlinie für die programmierten Achsen einschalten | | + | - | PM-NC |
| DYNFINISH | G | Dynamik für Schlichten | m | + | | PM-NC |
| DYNNORM [6] | G | Normale Dynamik | m | + | | PM-NC |
| DYNPOS | G | Dynamik für Positionierbetrieb, Gewindebohren | m | + | | PM-NC |
| DYNPREC | G | Dynamik für Feinschlichten | m | + | | PM-NC |
| DYNROUGH | G | Dynamik für Schruppen | m | + | | PM-NC |
| DYNSEMIFIN | G | Dynamik für Vorschlichten | m | + | | PM-NC |
| DZERO | P | Kennzeichnet alle D-Nummern der TO-Einheit als ungültig | | + | - | PM-NC |
| EAUTO | G | Festlegung des letzten Spline-Abschnitts durch die letzten 3 Punkte | m | + | | PM-NC |
| EGDEF | P | Definition eines elektronischen Getriebes | | + | - | PM-NC |
| EGDEL | P | Kopplungsdefinition für die Folgeachse löschen | | + | - | PM-NC |
| EGOFC | P | Elektronisches Getriebe kontinuierlich ausschalten | | + | - | PM-NC |
| EGOFS | P | Elektronisches Getriebe selektiv ausschalten | | + | - | PM-NC |
| EGON | P | Elektronisches Getriebe einschalten | | + | - | PM-NC |
| EGONSYN | P | Elektronisches Getriebe einschalten | | + | - | PM-NC |
| EGONSYNE | P | Elektronisches Getriebe einschalten, mit Vorgabe von Anfahrmodus | | + | - | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1][2][3][4][5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| ELSE | K | NC-Programm: Programmverzweigung bei nicht erfüllter IF-Bedingung | | + | - | PM-NC |
| ELSE | K | Synchronaktion: Auslösen von Aktionen bei nicht erfüllter Bedingung | | - | + | FM-SA |
| ENABLE | P | Interrupt EIN | | + | - | PM-NC |
| ENAT [6] | G | Natürlicher Kurvenübergang zum nächsten Verfahrsatz | m | + | | PM-NC |
| ENDFOR | K | Endezeile der FOR-Zählschleife | | + | | PM-NC |
| ENDIF | K | Endezeile der IF-Verzweigung | | + | | PM-NC |
| ENDLABEL | K | Endmarke für Teilprogrammwiederholungen über REPEAT | | + | | PM-NC, FM-B |
| ENDLOOP | K | Endezeile der Endlos-Programmschleife LOOP | | + | | PM-NC |
| ENDPROC | K | Endezeile eines Programms mit der Anfangszeile PROC | | + | | |
| ENDWHILE | K | Endezeile der WHILE-Schleife | | + | | PM-NC |
| ESRR | P | Antriebsautarkes ESR-Rückziehen im Antrieb parametrieren | | + | | PM-NC |
| ESRS | P | Antriebsautarkes ESR-Stillsetzen im Antrieb parametrieren | | + | | PM-NC |
| ETAN | G | Tangentialer Kurvenübergang zum nächsten Verfahrsatz bei Spline-Beginn | m | + | | PM-NC |
| EVERY | K | Synchronaktion ausführen bei Übergang der Bedingung von FALSE zu TRUE | | - | + | FM-SA |
| EX | K | Schlüsselwert für die Wertzuweisung in exponentieller Schreibweise | | + | | PM-NC |
| EXECSTRING | P | Übergabe einer String-Variablen mit der auszuführenden Teileprogrammzeile | | + | - | PM-NC |
| EXECTAB | P | Ein Element aus einer Bewegungstabelle abarbeiten | | + | - | PM-NC |
| EXECUTE | P | Programmausführung EIN | | + | - | PM-NC |
| EXP | F | Exponentialfunktion ex | | + | + | PM-NC |
| EXTCALL | A | Externes Unterprogramm abarbeiten | | + | + | PM-NC |
| EXTCLOSE | P | Zum Schreiben geöffnetes externes Gerät/ Datei schließen | | + | - | PM-NC |
| EXTERN | K | Bekanntmachung eines Unterprogramms mit Parameterübergabe | | + | | PM-NC |
| EXTOPEN | P | Externes Gerät/Datei für den Kanal zum Schreiben öffnen | | + | - | PM-NC |
| F | A | Vorschubwert<br>(in Verbindung mit G4 wird mit F auch die Verweilzeit programmiert) | | + | + | PM-NC |
| FA | K | Axialer Vorschub | m | + | + | PM-NC |
| FAD | A | Zustell-Vorschub für Weiches An- und Abfahren | | + | | PM-NC |
| FALSE | K | Logische Konstante: falsch | | + | + | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1][2][3][4][5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| FB | A | Satzweiser Vorschub | | + | | PM-NC |
| FCTDEF | P | Polynomfunktion definieren | | + | - | PM-NC |
| FCUB | G | Vorschub nach kubischem Spline veränderlich | m | + | | PM-NC |
| FD | A | Bahnvorschub für Handradüberlagerung | s | + | | PM-NC |
| FDA | K | Axialer Vorschub für Handradüberlagerung | s | + | | PM-NC |
| FENDNORM [6] | G | Eckenverzögerung AUS | m | + | | PM-NC |
| FFWOF [6] | G | Vorsteuerung AUS | m | + | | PM-NC |
| FFWON | G | Vorsteuerung Ein | m | + | | PM-NC |
| FGREF | K | Bezugsradius bei Rundachsen oder Bahnbezugsfaktoren bei Orientierungsachsen (Vektorinterpolation) | m | + | | PM-NC |
| FGROUP | P | Festlegung der Achse(n) mit Bahnvorschub | | + | - | PM-NC |
| FI | K | Parameter für Zugriff auf Framedaten: Feinverschiebung | | + | | PM-NC |
| FIFOCTRL | G | Steuerung des Vorlaufpuffers | m | + | | PM-NC |
| FILEDATE | P | Liefert Datum des zuletzt schreibenden Zugriffs auf die Datei | | + | - | PM-NC |
| FILEINFO | P | Liefert Summe von FILEDATE, FILESIZE, FILESTAT und FILETIME zusammen | | + | - | PM-NC |
| FILESIZE | P | Liefert aktuelle Größe der Datei | | + | - | PM-NC |
| FILESTAT | P | Liefert Filestatus der Rechte Lesen, Schreiben, Execute, Anzeigen , Löschen (rwxsd) | | + | - | PM-NC |
| FILETIME | P | Liefert Uhrzeit des zuletzt schreibenden Zugriffs auf die Datei | | + | - | PM-NC |
| FINEA | K | Bewegungsende beim Erreichen von "Genauhalt Fein" | m | + | | PM-NC |
| FL | K | Grenzgeschwindigkeit für Synchronachsen | m | + | | PM-NC |
| FLIN | G | Vorschub linear veränderlich | m | + | | PM-NC |
| FMA | K | Mehrere Vorschübe axial | m | + | | PM-NC |
| FNORM [6] | G | Vorschub normal nach DIN66025 | m | + | | PM-NC |
| FOC | K | Satzweise wirksame Moment/Kraft-Begrenzung | s | - | + | FM-SA |
| FOCOF | K | Modale Moment/Kraft-Begrenzung ausschalten | m | - | + | FM-SA |
| FOCON | K | Modale Moment/Kraft-Begrenzung einschalten | m | - | + | FM-SA |
| FOR | K | Zählschleife mit fester Anzahl von Durchläufen | | + | | PM-NC |
| FP | A | Festpunkt: Nummer des anzufahrenden Festpunkts | s | + | | PM-NC |
| FPO | K | Über ein Polynom programmierter Vorschubverlauf | | + | | PM-NC |
| FPR | P | Kennzeichnung Rundachse | | + | - | PM-NC |
| FPRAOF | P | Umdrehungsvorschub ausschalten | | + | - | PM-NC |
| FPRAON | P | Umdrehungsvorschub einschalten | | + | - | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1][2][3][4][5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| FRAME | K | Datentyp zur Festlegung von Koordinatensystemen | | + | | PM-NC |
| FRC | A | Vorschub für Radius und Fase | s | + | | PM-NC |
| FRCM | A | Vorschub für Radius und Fase modal | m | + | | PM-NC |
| FROM | K | Die Aktion wird ausgeführt, wenn die Bedingung einmal erfüllt ist und solange die Synchronaktion aktiv ist | | - | + | FM-SA |
| FTOC | P | Werkzeugfeinkorrektur ändern | | - | + | FM-SA |
| FTOCOF [6] | G | Online wirksame Werkzeugfeinkorrektur AUS | m | + | | PM-NC |
| FTOCON | G | Online wirksame Werkzeugfeinkorrektur EIN | m | + | | PM-NC |
| FXS | K | Fahren auf Festanschlag ein | m | + | + | PM-NC |
| FXST | K | Momentgrenze für Fahren auf Festanschlag | m | + | + | PM-NC |
| FXSW | K | Überwachungsfenster für Fahren auf Festanschlag | | + | + | PM-NC |
| FZ | K | Zahnvorschub | m | + | | PM-NC |

## Anweisungen G ... L

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| G0 | G | Linearinterpolation mit Eilgang (Eilgangsbewegung) | m | + | | PM-NC |
| G1 [6] | G | Linearinterpolation mit Vorschub (Geradeninterpolation) | m | + | | PM-NC |
| G2 | G | Kreisinterpolation im Uhrzeigersinn | m | + | | PM-NC |
| G3 | G | Kreisinterpolation gegen Uhrzeigersinn | m | + | | PM-NC |
| G4 | G | Verweilzeit, zeitlich vorbestimmt | s | + | | PM-NC |
| G5 | G | Schrägeinstechschleifen | s | + | | PM-NC |
| G7 | G | Ausgleichsbewegung beim Schrägeinstechschleifen | s | + | | PM-NC |
| G9 | G | Genauhalt - Geschwindigkeitsabnahme | s | + | | PM-NC |
| G17 [6] | G | Wahl der Arbeitsebene X/Y | m | + | | PM-NC |
| G18 | G | Wahl der Arbeitsebene Z/X | m | + | | PM-NC |
| G19 | G | Wahl der Arbeitsebene Y/Z | m | + | | PM-NC |
| G25 | G | Untere Arbeitsfeldbegrenzung | s | + | | PM-NC |
| G26 | G | Obere Arbeitsfeldbegrenzung | s | + | | PM-NC |
| G33 | G | Gewindeschneiden mit konstanter Steigung | m | + | | PM-NC |
| G34 | G | Gewindeschneiden mit linear zunehmender Steigung | m | + | | PM-NC |
| G35 | G | Gewindeschneiden mit linear abnehmender Steigung | m | + | | PM-NC |
| G40 [6] | G | Werkzeugradiuskorrektur AUS | m | + | | PM-NC |
| G41 | G | Werkzeugradiuskorrektur links von der Kontur | m | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| G42 | G | Werkzeugradiuskorrektur rechts von der Kontur | m | + | | PM-NC |
| G53 | G | Unterdrückung der aktuellen Nullpunktverschiebung (satzweise) | s | + | | PM-NC |
| G54 | G | 1.Einstellbare Nullpunktverschiebung | m | + | | PM-NC |
| G55 | G | 2. Einstellbare Nullpunktverschiebung | m | + | | PM-NC |
| G56 | G | 3. Einstellbare Nullpunktverschiebung | m | + | | PM-NC |
| G57 | G | 4. Einstellbare Nullpunktverschiebung | m | + | | PM-NC |
| G58 (840D sl) | G | Absolute programmierbare Nullpunktverschiebung (Grobverschiebung) | s | + | | PM-NC |
| G58 (828D) | G | 5. Einstellbare Nullpunktverschiebung | m | + | | PM-NC |
| G59 (840D sl) | G | Additive programmierbare Nullpunktverschiebung (Feinverschiebung) | s | + | | PM-NC |
| G59 (828D) | G | 6. Einstellbare Nullpunktverschiebung | m | + | | PM-NC |
| G60 [6] | G | Genauhalt - Geschwindigkeitsabnahme | m | + | | PM-NC |
| G62 | G | Eckenverzögerung an Innenecken bei aktiver Werkzeuradiuskorrektur (G41, G42) | m | + | | PM-NC |
| G63 | G | Gewindebohren mit Ausgleichsfutter | s | + | | PM-NC |
| G64 | G | Bahnsteuerbetrieb | m | + | | PM-NC |
| G70 | G | Inch-Maßangabe für geometrische Angaben (Längen) | m | + | + | PM-NC |
| G71 [6] | G | Metrische Maßangabe für geometrische Angaben (Längen) | m | + | + | PM-NC |
| G74 | G | Referenzpunktanfahren | s | + | | PM-NC |
| G75 | G | Festpunktanfahren | s | + | | PM-NC |
| G90 [6] | G | Maßangabe absolut | m/s | + | | PM-NC |
| G91 | G | Kettenmaßangabe | m/s | + | | PM-NC |
| G93 | G | Zeitreziproker Vorschub 1/min | m | + | | PM-NC |
| G94 [6] | G | Linearvorschub F in mm/min oder inch/min und Grad/min | m | + | | PM-NC |
| G95 | G | Umdrehungsvorschub F in mm/U oder inch/U | m | + | | PM-NC |
| G96 | G | Umdrehungsvorschub (wie bei G95) und konstante Schnittgeschwindigkeit | m | + | | PM-NC |
| G97 | G | Umdrehungsvorschub und konstante Spindeldrehzahl (konstante Schnittgeschwindigkeit AUS) | m | + | | PM-NC |
| G110 | G | Polprogrammierung relativ zur letzten programmierten Sollposition | s | + | | PM-NC |
| G111 | G | Polprogrammierung relativ zum Nullpunkt des aktuellen Werkstück-Koordinatensystems | s | + | | PM-NC |
| G112 | G | Polprogrammierung relativ zum letzten gültigen Pol | s | + | | PM-NC |
| G140 [6] | G | Anfahrrichtung WAB festgelegt durch G41/G42 | m | + | | PM-NC |
| G141 | G | Anfahrrichtung WAB links der Kontur | m | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| G142 | G | Anfahrrichtung WAB rechts der Kontur | m | + | | PM-NC |
| G143 | G | Anfahrrichtung WAB tangentenabhängig | m | + | | PM-NC |
| G147 | G | Weiches Anfahren mit Gerade | s | + | | PM-NC |
| G148 | G | Weiches Abfahren mit Gerade | s | + | | PM-NC |
| G153 | G | Unterdrückung aktueller Frames inklusive Basisframe | s | + | | PM-NC |
| G247 | G | Weiches Anfahren mit Viertelkreis | s | + | | PM-NC |
| G248 | G | Weiches Abfahren mit Viertelkreis | s | + | | PM-NC |
| G290 [6] | G | Umschalten auf SINUMERIK-Mode EIN | m | + | | FM-TM |
| G291 | G | Umschalten auf ISO2/3-Mode EIN | m | + | | FM-TM |
| G331 | G | Gewindebohren ohne Ausgleichsfutter, positive Steigung, Rechtslauf | m | + | | PM-NC |
| G332 | G | Gewindebohren ohne Ausgleichsfutter, negative Steigung, Linkslauf | m | + | | PM-NC |
| G335 | G | Drehen eines balligen Gewindes im Uhrzeigersinn | m | + | | PM-NC |
| G336 | G | Drehen eines balligen Gewindes gegen den Uhrzeigersinn | m | + | | PM-NC |
| G340 [6] | G | Anfahrsatz räumlich (Tiefe und in der Ebene zugleich (Helix)) | m | + | | PM-NC |
| G341 | G | Zuerst in der senkrechten Achse zustellen (z), dann Anfahren in der Ebene | m | + | | PM-NC |
| G347 | G | Weiches Anfahren mit Halbkreis | s | + | | PM-NC |
| G348 | G | Weiches Abfahren mit Halbkreis | s | + | | PM-NC |
| G450 [6] | G | Übergangskreis | m | + | | PM-NC |
| G451 | G | Schnittpunkt der Äquidistanten | m | + | | PM-NC |
| G460 [6] | G | Einschalten der Kollisionsüberwachung für An- und Abfahrsatz | m | + | | PM-NC |
| G461 | G | Einfügen eines Kreises im WRK-Satz | m | + | | PM-NC |
| G462 | G | Einfügen einer Geraden im WRK-Satz | m | + | | PM-NC |
| G500 [6] | G | Ausschalten aller einstellbaren Frames, Basisframes sind aktiv | m | + | | PM-NC |
| G505 ... G599 | G | 5. ... 99. Einstellbare Nullpunktverschiebung | m | + | | PM-NC |
| G601 [6] | G | Satzwechsel bei Genauhalt fein | m | + | | PM-NC |
| G602 | G | Satzwechsel bei Genauhalt grob | m | + | | PM-NC |
| G603 | G | Satzwechsel bei IPO-Satzende | m | + | | PM-NC |
| G621 | G | Eckenverzögerung an allen Ecken | m | + | | PM-NC |
| G641 | G | Bahnsteuerbetrieb mit Überschleifen nach Wegkriterium (= programmierbarer Überschleifabstand) | m | + | | PM-NC |
| G642 | G | Bahnsteuerbetrieb mit Überschleifen unter Einhaltung definierter Toleranzen | m | + | | PM-NC |
| G643 | G | Bahnsteuerbetrieb mit Überschleifen unter Einhaltung definierter Toleranzen (satzintern) | m | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| G644 | G | Bahnsteuerbetrieb mit Überschleifen mit maximal möglicher Dynamik | m | + | | PM-NC |
| G645 | G | Bahnsteuerbetrieb mit Überschleifen von Ecken und tangentialer Satzübergänge unter Einhaltung definierter Toleranzen | m | + | | PM-NC |
| G700 | G | Inch-Maßangabe für geometrische und technologische Angaben (Längen, Vorschub) | m | + | + | PM-NC |
| G710 [6] | G | Metrische Maßangabe für geometrische und technologische Angaben (Längen, Vorschub) | m | + | + | PM-NC |
| G810 [6], ..., G819 | G | Für den OEM-Anwender reservierte G-Gruppe | | + | | PM-NC |
| G820 [6], ..., G829 | G | Für den OEM-Anwender reservierte G-Gruppe | | + | | PM-NC |
| G931 | G | Vorschubvorgabe durch Verfahrzeit, konstante Bahngeschwindigkeit ausschalten | m | + | | |
| G942 | G | Linearvorschub und konstante Schnittgeschwindigkeit oder Spindeldrehzahl einfrieren | m | + | | |
| G952 | G | Umdrehungsvorschub und konstante Schnittgeschwindigkeit oder Spindeldrehzahl einfrieren | m | + | | |
| G961 | G | Linearvorschub (wie bei G94) und konstante Schnittgeschwindigkeit | m | + | | PM-NC |
| G962 | G | Linearvorschub oder Umdrehungsvorschub und konstante Schnittgeschwindigkeit | m | + | | PM-NC |
| G971 | G | Linearvorschub und konstante Spindeldrehzahl (konstante Schnittgeschwindigkeit AUS) | m | + | | PM-NC |
| G972 | G | Linearvorschub oder Umdrehungsvorschub und konstante Spindeldrehzahl (konstante Schnittgeschwindigkeit AUS) | m | + | | PM-NC |
| G973 | G | Umdrehungsvorschub ohne Spindeldrehzahlbegrenzung und konstante Spindeldrehzahl (G97 ohne LIMS für ISO-Modus) | m | + | | PM-NC |
| GEOAX | P | Den Geometrieachsen 1 - 3 neue Kanalachsen zuordnen | | + | - | PM-NC |
| GET | P | Freigegebene Achse zwischen Kanälen tauschen | | + | + | PM-NC |
| GETACTT | F | Bestimmt das aktive Werkzeug aus einer Gruppe von gleichnamigen Werkzeugen | | + | - | FM-TM |
| GETACTTD | F | Bestimmt zu einer absoluten D-Nummer die zugehörige T-Nummer | | + | - | PM-NC |
| GETD | P | Achse direkt zwischen Kanälen tauschen | | + | - | PM-NC |
| GETDNO | F | Liefert D-Nummer einer Schneide (CE) eines Werkzeugs (T) | | + | - | PM-NC |
| GETEXET | P | Lesen der eingewechselten T-Nummer | | + | - | FM-TM |
| GETFREELOC | P | Für ein gegebenes Werkzeug einen Leerplatz in den Magazinen suchen | | + | - | FM-TM |
| GETSELT | P | Vorgewählte T-Nummer liefern | | + | - | FM-TM |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| GETT | F | T-Nummer zu Werkzeugnamen bestimmen | | + | - | FM-TM |
| GETTCOR | F | Werkzeuglängen bzw. Werkzeuglängenkomponenten auslesen | | + | - | PM-NC |
| GETTENV | F | T-, D-, und DL-Nummern lesen | | + | - | PM-NC |
| GETVARAP | F | Zugriffsrecht auf eine System-/Anwendervariable lesen | | + | - | PM-NC |
| GETVARDFT | F | Standardwert einer System-/Anwendervariablen lesen | | + | - | PM-NC |
| GETVARLIM | F | Grenzwerte einer System-/Anwendervariablen lesen | | + | - | PM-NC |
| GETVARPHU | F | Physikalische Einheit einer System-/Anwendervariablen lesen | | + | - | PM-NC |
| GETVARTYP | F | Datentyp einer System-/Anwendervariablen lesen | | + | - | PM-NC |
| GFRAME0 ... GFRAME100 | G | Aktivierung des Schleifframes <n> der Datenhaltung im Kanal | m | + | | PM-NC |
| GOTO | K | Sprunganweisung erst vorwärts dann rückwärts (Richtung erst zum Programm-Ende und dann zum Programm-Anfang) | | + | | PM-NC |
| GOTOB | K | Sprunganweisung rückwärts (Richtung Programm-Anfang) | | + | | PM-NC |
| GOTOC | K | Wie GOTO, aber Alarm 14080 "Sprungziel nicht gefunden" unterdrücken | | + | | PM-NC |
| GOTOF | K | Sprunganweisung vorwärts (Richtung Programm-Ende) | | + | | PM-NC |
| GOTOS | K | Rücksprung auf Programmanfang | | + | | PM-NC |
| GP | K | Schlüsselwort zur indirekten Programmierung von Positionsattributen | | + | | PM-NC |
| GROUP_ ADDEND | C (T) | Ende Einfahrzusatz | | + | | PM-NC |
| GROUP_BEGIN | C (T) | Anfang Programmblock | | + | | PM-NC |
| GROUP_END | C (T) | Ende Programmblock | | + | | PM-NC |
| GWPSOF | P | Konstante Scheibenumfangsgeschwindigkeit (SUG) abwählen | s | + | - | PM-NC |
| GWPSON | P | Konstante Scheibenumfangsgeschwindigkeit (SUG) anwählen | s | + | - | PM-NC |
| H... | A | Hilfsfunktionsausgabe an die PLC | | + | + | PM-NC, FM-B |
| HOLES1 | C (T) | Lochreihe | | + | | PM-NC |
| HOLES2 | C (T) | Lochkreis | | + | | PM-NC |
| I | A | Interpolationsparameter | s | + | | PM-NC |
| I1 | A | Zwischenpunktkoordinate | s | + | | PM-NC |
| IC | K | Kettenmaßeingabe | s | + | | PM-NC |
| ICYCOF | P | Alle Sätze eines Technologiezyklus nach ICYCOF in einem IPO-Takt abarbeiten | | + | + | FM-SA |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| ICYCON | P | Jeden Satz eines Technologiezyklus nach ICYCON in einem separaten IPO-Takt abarbeiten | | + | + | FM-SA |
| ID | K | Kennzeichnung für modale Synchronaktionen | m | - | + | FM-SA |
| IDS | K | Kennzeichnung für modale statische Synchronaktionen | | - | + | FM-SA |
| IF | K | Einleitung eines bedingten Sprungs im Teileprogramm / Technologiezyklus | | + | + | PM-NC |
| INDEX | F | Index eines Zeichens im Eingangsstring bestimmen | | + | - | PM-NC |
| INICF | K | Initialisierung der Variablen bei NEWCONF | | + | | PM-NC |
| INIPO | K | Initialisierung der Variablen bei PowerOn | | + | | PM-NC |
| INIRE | K | Initialisierung der Variablen bei Reset | | + | | PM-NC |
| INIT | P | Anwahl eines bestimmten NC-Programms zur Abarbeitung in einem bestimmten Kanal | | + | - | PM-NC |
| INITIAL | | Erzeugen eines INI-Files über alle Bereiche | | + | | PM-NC |
| INT | K | Datentyp: Ganzzahliger Wert mit Vorzeichen | | + | | PM-NC |
| INTERSEC | F | Schnittpunkt zwischen zwei Konturelementen berechnen | | + | - | PM-NC |
| INVCCW | G | Evolvente fahren, gegen den Uhrzeigersinn | m | + | | PM-NC |
| INVCW | G | Evolvente fahren, im Uhrzeigersinn | m | + | | PM-NC |
| INVFRAME | F | Aus einem Frame den inversen Frame berechnen | | + | - | FM-B |
| IP | K | Variabler Interpolationsparameter | | + | | PM-NC |
| IPOBRKA | P | Bewegungskriterium ab Einsatzpunkt der Bremsrampe | m | + | + | |
| IPOENDA | K | Bewegungsende beim Erreichen von "IPO-Stopp" | m | + | | PM-NC |
| IPTRLOCK | P | Beginn des suchunfähigen Programmabschnitts auf nächsten Maschinenfunktionssatz einfrieren. | m | + | - | PM-NC |
| IPTRUNLOCK | P | Ende des suchunfähigen Programmabschnitts auf aktuellen Satz zum Unterbrechungszeitpunkt setzen. | m | + | - | PM-NC |
| IR | A | Kreismittelpunktkoordinate (X-Richtung) beim Drehen balliger Gewinde | | + | | PM-NC |
| ISAXIS | F | Prüfen, ob die als Parameter angegebene Geometrieachse 1 ist | | + | - | PM-NC |
| ISD | A | Eintauchtiefe | m | + | | PM-NC |
| ISFILE | F | Prüfen, ob eine Datei im NC-Anwendungsspeicher vorhanden ist | | + | - | PM-NC |
| ISNUMBER | F | Prüfen, ob Eingangsstring in Zahl umgewandelt werden kann | | + | - | PM-NC |
| ISOCALL | K | Indirekter Aufruf eines in ISO-Sprache programmierten Programms | | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| ISVAR | F | Prüfen, ob der Übergabeparameter eine in der NC bekannte Variable enthält | | + | - | PM-NC |
| J | A | Interpolationsparameter | s | + | | PM-NC |
| J1 | A | Zwischenpunktkoordinate | s | + | | PM-NC |
| JERKA | P | Über MD eingestelltes Beschleunigungsverhalten für die programmierten Achsen aktivieren | | + | - | |
| JERKLIM | K | Maximalen Achsruck anpassen | m | + | | PM-NC |
| JERKLIMA | K | Reduktion oder Überhöhung des maximalen Rucks einer Folgeachse | m | + | + | PM-NC |
| JR | A | Kreismittelpunktkoordinate (Y-Richtung) beim Drehen balliger Gewinde | | + | | PM-NC |
| K | A | Interpolationsparameter | s | + | | PM-NC |
| K1 | A | Zwischenpunktkoordinate | s | + | | PM-NC |
| KONT | G | Kontur umfahren bei der Werkzeugkorrektur | m | + | | PM-NC |
| KONTC | G | Mit krümmungsstetigem Polynom an-/abfahren | m | + | | PM-NC |
| KONTT | G | Mit tangentenstetigem Polynom an-/abfahren | m | + | | PM-NC |
| KR | A | Kreismittelpunktkoordinate (Z-Richtung) beim Drehen balliger Gewinde | | + | | PM-NC |
| L | A | Unterprogramm-Nummer | s | + | + | PM-NC |
| LEAD | A | Voreilwinkel<br>1. Werkzeugorientierung<br>2. Orientierungpolynome | m | + | | PM-NC |
| LEADOF | P | Axiale Leitwerkkopplung AUS | | + | + | PM-NC |
| LEADON | P | Axiale Leitwerkkopplung EIN | | + | + | PM-NC |
| LENTOAX | F | Liefert Informationen über die Zuordnung der Werkzeuglängen L1, L2 und L3 des aktiven Werkzeugs zur Abszisse, Ordinate und Applikate | | + | - | PM-NC |
| LFOF [6] | G | Schnellrückzug für Gewindeschneiden AUS | m | + | | PM-NC |
| LFON | G | Schnellrückzug für Gewindeschneiden EIN | m | + | | PM-NC |
| LFPOS | G | Rückzug der mit POLFMASK oder POLFMLIN bekannt gemachten Achse auf die mit POLF programmierte absolute Achsposition | m | + | | PM-NC |
| LFTXT [6] | G | Ebene der Rückzugsbewegung beim Schnellabheben wird bestimmt aus der Bahntangente und der aktuellen Werkzeugrichtung | m | + | | PM-NC |
| LFWP | G | Ebene der Rückzugsbewegung beim Schnellabheben wird bestimmt durch die aktuelle Arbeitsebene (G17/G18/G19) | m | + | | PM-NC |
| LIFTFAST | K | Schnellabheben | | + | | PM-NC |
| LIMS | K | Drehzahlbegrenzung<br>bei G96/G961 und G97 | m | + | | PM-NC |
| LLI | K | Unterer Grenzwert von Variablen | | + | | PM-NC |
| LN | F | Natürlicher Logarithmus | | + | + | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| LOCK | P | Synchronaktion mit ID sperren (Technologiezyklus stoppen) | | - | + | FM-SA |
| LONGHOLE | C (T) | Langloch | | + | | PM-NC |
| LOOP | K | Einleitung einer Endlosschleife | | + | | PM-NC |

## Anweisungen M ... R

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| M0 | | Programmierter Halt | | + | + | PM-NC |
| M1 | | Wahlweiser Halt | | + | + | PM-NC |
| M2 | | Programmende Hauptprogramm (wie M30) | | + | + | PM-NC |
| M3 | | Spindeldrehrichtung rechts | | + | + | PM-NC |
| M4 | | Spindeldrehrichtung links | | + | + | PM-NC |
| M5 | | Spindel Halt | | + | + | PM-NC |
| M6 | | Werkzeugwechsel | | + | + | PM-NC |
| M17 | | Programmende Unterprogramm | | + | + | PM-NC |
| M19 | | Spindelpositionierung auf die im SD43240 eingetragene Position | | + | + | PM-NC |
| M30 | | Programmende Hauptprogramm (wie M2) | | + | + | PM-NC |
| M40 | | Automatische Getriebeschaltung | | + | + | PM-NC |
| M41 ... M45 | | Getriebestufe 1 ... 5 | | + | + | PM-NC |
| M70 | | Übergang in Achsbetrieb | | + | + | PM-NC |
| MASLDEF | P | Master/Slave-Achsverband definieren | | + | + | PM-NC |
| MASLDEL | P | Master/Slave-Achsverband trennen und Definition des Verbandes löschen | | + | + | PM-NC |
| MASLOF | P | Ausschalten einer temporären Kopplung | | + | + | PM-NC |
| MASLOFS | P | Ausschalten einer temporären Kopplung mit automatischem Stillsetzen der Slave-Achse | | + | + | PM-NC |
| MASLON | P | Einschalten einer temporären Kopplung | | + | + | PM-NC |
| MATCH | F | Suchen eines String im String | | + | - | PM-NC |
| MAXVAL | F | Größerer Wert zweier Variablen (arithm. Funktion) | | + | + | PM-NC |
| MCALL | K | Modaler Unterprogrammaufruf | | + | | PM-NC |
| MEAC | K | Axiales kontinuierliches Messen ohne Restweglöschen | s | + | + | PM-NC |
| MEAFRAME | F | Frame-Berechnung aus Messpunkten | | + | - | PM-NC |
| MEAS | A | Messen mit Restweglöschen | s | + | | PM-NC |
| MEASA | K | Axiales Messen mit Restweglöschen | s | + | + | PM-NC |
| MEASURE | F | Berechnungsmethode für die Werkstück- und Werkzeugvermessung | | + | - | FM-TE |
| MEAW | A | Messen ohne Restweglöschen | s | + | | PM-NC |
| MEAWA | K | Axiales Messen ohne Restweglöschen | s | + | + | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| MI | K | Zugriff auf Frame-Daten: Spiegelung | | + | | PM-NC |
| MINDEX | F | Index eines Zeichens im Eingangsstring bestimmen | | + | - | PM-NC |
| MINVAL | F | Kleinerer Wert zweier Variablen (arithm. Funktion) | | + | + | PM-NC |
| MIRROR | G | Programmierbare Spiegelung | s | + | | PM-NC |
| MMC | P | Aus dem Teileprogramm interaktiv Dialogfenster am HMI aufrufen | | + | - | PM-NC |
| MOD | K | Modulo-Division | | + | | PM-NC |
| MODAXVAL | F | Modulo-Position einer Modulo-Rundachse ermitteln | | + | - | PM-NC |
| MOV | K | Positionierachse starten | | - | + | FM-SA |
| MOVT | A | Endpunkt einer Verfahrbewegung in Werkzeugrichtung angeben | | | | FM-B |
| MSG | P | Programmierbare Meldungen | m | + | - | PM-NC |
| MVTOOL | P | Sprachbefehl zum Bewegen eines Werkzeugs | | + | - | FM-TM |
| N | A | NC-Nebensatznummer | | + | | PM-NC |
| NAMETOINT | F | Systemvariablenindex ermitteln | | + | | PM-NC |
| NC | K | Spezifizierung des Gültigkeitsbereichs von Daten | | + | | PM-NC |
| NEWCONF | P | Geänderte Maschinendaten übernehmen (entspricht "Maschinendatum wirksam setzen") | | + | - | PM-NC |
| NEWMT | F | Neues Multitool anlegen | | + | - | FM-TM |
| NEWT | F | Neues Werkzeug anlegen | | + | - | FM-TM |
| NORM [6] | G | Normaleinstellung im Anfangs-, Endpunkt bei der Werkzeugkorrektur | m | + | | PM-NC |
| NOT | K | Logisches NICHT (Negation) | | + | | PM-NC |
| NPROT | P | Maschinenspezifischer Schutzbereich EIN/AUS | | + | - | PM-NC |
| NPROTDEF | P | Definition eines maschinenspezifischen Schutzbereichs | | + | - | PM-NC |
| NUMBER | F | Eingangsstring in Zahl umwandeln | | + | - | PM-NC |
| OEMIPO1 | G | OEM-Interpolation 1 | m | + | | PM-NC |
| OEMIPO2 | G | OEM-Interpolation 2 | m | + | | PM-NC |
| OF | K | Schlüsselwort in der CASE-Verzweigung | | + | | PM-NC |
| OFFN | A | Aufmaß zur programmierten Kontur | m | + | | PM-NC |
| OMA1 | A | OEM-Adresse 1 | m | + | | PM-NC |
| OMA2 | A | OEM-Adresse 2 | m | + | | PM-NC |
| OMA3 | A | OEM-Adresse 3 | m | + | | PM-NC |
| OMA4 | A | OEM-Adresse 4 | m | + | | PM-NC |
| OMA5 | A | OEM-Adresse 5 | m | + | | PM-NC |
| OR | K | Logischer Operator, ODER-Verknüpfung | | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| ORIAXES | G | Lineare Interpolation der Maschinenachsen oder Orientierungsachsen | m | + | | PM-NC |
| ORIAXPOS | G | Orientierungswinkel über virtuelle Orientierungsachsen mit Rundachspositionen | m | + | | PM-NC |
| ORIC [6] | G | Orientierungsänderungen an Außenecken werden dem einzufügenden Kreissatz überlagert | m | + | | PM-NC |
| ORICONCCW | G | Interpolation auf einer Kreismantelfläche im Gegenuhrzeigersinn | m | + | | PM-NC, FM-TR |
| ORICONCW | G | Interpolation auf einer Kreismantelfläche im Uhrzeigersinn | m | + | | PM-NC, FM-TR |
| ORICONIO | G | Interpolation auf einer Kreismantelfläche mit Angabe einer Zwischenorientierung | m | + | | PM-NC, FM-TR |
| ORICONTO | G | Interpolation auf einer Kreismantelfläche im tangentialen Übergang (Angabe der Endorientierung) | m | + | | PM-NC, FM-TR |
| ORICURVE | G | Interpolation der Orientierung mit Vorgabe der Bewegung zweier Kontaktpunkte des Werkzeugs | m | + | | PM-NC, FM-TR |
| ORID | G | Orientierungsänderungen werden vor dem Kreissatz ausgeführt | m | + | | PM-NC |
| ORIEULER [6] | G | Orientierungswinkel über Euler-Winkel | m | + | | PM-NC |
| ORIMKS | G | Werkzeugorientierung im Maschinen-Koordinatensystem | m | + | | PM-NC |
| ORIPATH | G | Werkzeugorientierung bezogen auf die Bahn | m | + | | PM-NC |
| ORIPATHS | G | Werkzeugorientierung bezogen auf die Bahn, ein Knick im Orientierungsverlauf wird geglättet | m | + | | PM-NC |
| ORIPLANE | G | Interpolation in einer Ebene (entspricht ORIVECT) Großkreisinterpolation | m | + | | PM-NC |
| ORIRESET | P | Grundstellung der Werkzeugorientierung mit bis zu 3 Orientierungsachsen | | + | - | PM-NC |
| ORIROTA [6] | G | Drehwinkel zu einer absolut vorgegebenen Drehrichtung | m | + | | PM-NC |
| ORIROTC | G | Tangentialer Drehvektor zur Bahntangente | m | + | | PM-NC |
| ORIROTR | G | Drehwinkel relativ zur Ebene zwischen Start- und Endorientierung | m | + | | PM-NC |
| ORIROTT | G | Drehwinkel relativ zur Änderung des Orientierungsvektors | m | + | | PM-NC |
| ORIRPY | G | Orientierungswinkel über RPY-Winkel (XYZ) | m | + | | PM-NC |
| ORIRPY2 | G | Orientierungswinkel über RPY-Winkel (ZYX) | m | + | | PM-NC |
| ORIS | A | Orientierungsänderung | m | + | | PM-NC |
| ORISOF [6] | G | Glättung des Orientierungsverlaufs AUS | m | + | | PM-NC |
| ORISOLH | F | Orientierungen berechnen | | + | | PM-NC |
| ORISON | G | Glättung des Orientierungsverlaufs EIN | m | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| ORIVECT [6] | G | Großkreisinterpolation (identisch mit ORIPLANE) | m | + | | PM-NC |
| ORIVIRT1 | G | Orientierungswinkel über virtuelle Orientierungsachsen (Definition 1) | m | + | | PM-NC |
| ORIVIRT2 | G | Orientierungswinkel über virtuelle Orientierungsachsen (Definition 1) | m | + | | PM-NC |
| ORIWKS [6] | G | Werkzeugorientierung im Werkstück-Koordinatensystem | m | + | | PM-NC |
| OS | K | Pendeln ein/aus | | + | | PM-NC |
| OSB | K | Pendeln: Startpunkt | m | + | | PM-NC |
| OSC | G | Konstante Glättung Werkzeugorientierung | m | + | | PM-NC |
| OSCILL | K | Axis: 1 - 3 Zustellachsen | m | + | | PM-NC |
| OSCTRL | K | Optionen pendeln | m | + | | PM-NC |
| OSD | G | Überschleifen de Werkzeugorientierung durch Vorgabe der Überschleiflänge mit SD | m | + | | PM-NC |
| OSE | K | Pendeln Endpunkt | m | + | | PM-NC |
| OSNSC | K | Pendeln: Ausfunkanzahl | m | + | | PM-NC |
| OSOF [6] | G | Glättung der Werkzeugorientierung AUS | m | + | | PM-NC |
| OSP1 | K | Pendeln: linker Umkehrpunkt | m | + | | PM-NC |
| OSP2 | K | Pendeln rechter Umkehrpunkt | m | + | | PM-NC |
| OSS | G | Glättung der Werkzeugorientierung am Satzende | m | + | | PM-NC |
| OSSE | G | Glättung der Werkzeugorientierung am Satzanfang und Satzende | m | + | | PM-NC |
| OST | G | Überschleifen der Werkzeugorientierung durch Vorgabe der Winkeltoleranz in Grad mit dem SD (maximale Abweichung vom programmiert. Orientierungsverlauf) | m | + | | PM-NC |
| OST1 | K | Pendeln: Haltepunkt im linken Umkehrpunkt | m | + | | PM-NC |
| OST2 | K | Pendeln: Haltepunkt im rechten Umkehrpunkt | m | + | | PM-NC |
| OTOL | A | Orientierungstoleranz für Kompressor-Funktionen, Orientierungsglättung und Überschleifarten | m | + | - | PM-NC |
| OTOLG0 | A | Orientierungstoleranz bei Eilgangbewegungen | m | + | - | PM-NC |
| OVR | K | Drehzahlkorrektur | m | + | | PM-NC |
| OVRA | K | Axiale Drehzahlkorrektur | m | + | + | PM-NC |
| OVRRAP | K | Eilgang-Korrektur | m | + | | PM-NC |
| P | A | Anzahl Unterprogrammdurchläufe | | + | | PM-NC |
| PAROT | G | Werkstückkoordinatensystem am Werkstück ausrichten | m | + | | PM-NC |
| PAROTOF [6] | G | Werkstückbezogene Frame-Drehung ausschalten | m | + | | PM-NC |
| PCALL | K | Unterprogramme mit absoluter Pfadangabe und Parameterübergabe aufrufen | | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| PDELAYOF | G | Verzögerung beim Stanzen AUS | m | + | | PM-NC |
| PDELAYON [6] | G | Verzögerung beim Stanzen EIN | m | + | | PM-NC |
| PHI | K | Drehwinkel der Orientierung um die Richtungsachse des Kegels | | + | | PM-NC |
| PHU | K | Physikalische Einheit einer Variablen | | + | | PM-NC |
| PL | A | 1. B-Spline: Knotenabstand<br>2. Polynom-Interpolation: Länge des Parameterintervalls bei Polynom-Interpolation | s | + | | PM-NC |
| PM | K | pro Minute | | + | | PM-NC |
| PO | K | Polynomkoeffizient bei Polynom-Interpolation | s | + | | PM-NC |
| POCKET3 | C (T) | Rechtecktasche fräsen | | + | | PM-NC |
| POCKET4 | C (T) | Kreistasche fräsen | | + | | PM-NC |
| POLF | K | Rückzugsposition LIFTFAST | m | + | | PM-NC |
| POLFA | P | Rückzugsposition von Einzelachsen mit $AA_ESR_TRIGGER starten | m | + | + | PM-NC |
| POLFMASK | P | Achsen für den Rückzug ohne Zusammenhang zwischen den Achsen freigeben | m | + | - | PM-NC |
| POLFMLIN | P | Achsen für den Rückzug mit linearen Zusammenhang zwischen den Achsen freigeben | m | + | - | PM-NC |
| POLY | G | Polynom-Interpolation | m | + | | PM-NC |
| POLYPATH | P | Polynom-Interpolation selektierbar für die Achsgruppen AXIS oder VECT | m | + | - | PM-NC |
| PON | G | Stanzen EIN | m | + | | PM-NC |
| PONS | G | Stanzen EIN im IPO-Takt | m | + | | PM-NC |
| POS | K | Achse positionieren | | + | + | PM-NC |
| POSA | K | Achse positionieren über Satzgrenze | | + | + | PM-NC |
| POSM | P | Magazin positionieren | | + | - | FM-TM |
| POSMT | P | Multitool auf WZ-Halter auf Platznummer positionieren | | + | - | FM-TM |
| POSP | K | Positionieren in Teilstücken (Pendeln) | | + | | PM-NC |
| POSRANGE | F | Ermitteln, ob sich die aktuell interpolierte Sollposition einer Achse in einem Fenster um eine vorgegebene Referenzposition befindet | | + | + | FM-SA |
| POT | F | Quadrat<br>(Arithmetische Funktion) | | + | + | PM-NC |
| PR | K | Pro Umdrehung | | + | | PM-NC |
| PREPRO | PA | Unterprogramme mit Vorbereitung kennzeichnen | | + | | PM-NC |
| PRESETON | P | Istwertsetzen mit Verlust des Referenzierstatus | | + | + | PM-NC |
| PRESETONS | P | Istwertsetzen ohne Verlust des Referenzierstatus | | + | + | PM-NC |
| PRIO | K | Schlüsselwort zum Setzen der Priorität bei der Behandlung von Interrupts | | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| PRLOC | K | Initialisierung der Variablen bei Reset nur nach lokaler Änderung | | + | | PM-NC |
| PROC | K | Erste Anweisung eines Programms | | + | | PM-NC |
| PROTA | P | Neuberechnung des Kollisionsmodells anfordern | | + | | PM-NC |
| PROTD | F | Abstand zweier Schutzbereiche berechnen | | + | | PM-NC |
| PROTS | P | Setzen des Schutzbereichszustandes | | + | | PM-NC |
| PSI | K | Öffnungswinkel des Kegels | | + | | PM-NC |
| PTP | G | Punkt-zu-Punkt-Bewegung (PTP-Fahren) | m | + | | PM-NC |
| PTPG0 | G | Punkt-zu-Punkt-Bewegung nur bei G0, sonst Bahnbewegung CP | m | + | | PM-NC |
| PTPWOC | G | Punkt-zu-Punkt-Bewegung ohne Ausgleichsbewegungen, die durch Orientierungsänderungen verursacht sind | m | + | | PM-NC |
| PUNCHACC | P | Wegabhängige Beschleunigung beim Nibbeln | | + | - | PM-NC |
| PUTFTOC | P | Werkzeugfeinkorrektur für paralleles Abrichten | | + | - | PM-NC |
| PUTFTOCF | P | Werkzeugfeinkorrektur in Abhängigkeit einer mit FCTDEF festgelegten Funktion für paralleles Abrichten | | + | - | PM-NC |
| PW | A | B-Spline, Punkt-Gewicht | s | + | | PM-NC |
| QU | K | Schnelle Zusatz-(Hilfs-)funktionsausgabe | | + | | PM-NC |
| R... | A | Rechenparameter auch als einstellbarer Adressbezeichner und mit numerischer Erweiterung | | + | | PM-NC |
| RAC | K | Absolut satzweise achsspezifische Radiusprogrammierung | s | + | | PM-NC |
| RDISABLE | P | Einlesesperre | | - | + | FM-SA |
| READ | P | Liest in der angegebenen Datei eine oder mehrere Zeilen ein und legt gelesene Informationen im Feld ab | | + | - | PM-NC |
| REAL | K | Datentyp: Gleitpunktvariable mit Vorzeichen (reale Zahlen) | | + | | PM-NC |
| REDEF | K | Redefinition von Systemvariablen, Anwendervariablen und NC-Sprachbefehlen | | + | | PM-NC |
| RELEASE | P | Maschinenachsen zum Achstausch freigeben | | + | + | PM-NC |
| REP | K | Schlüsselwort zur Initialisierung aller Elemente eines Feldes mit demselben Wert | | + | | PM-NC |
| REPEAT | K | Wiederholung einer Programmschleife | | + | | PM-NC |
| REPEATB | K | Wiederholung einer Programmzeile | | + | | PM-NC |
| REPOSA | G | Wiederanfahren an die Kontur linear mit allen Achsen | s | + | | PM-NC |
| REPOSH | G | Wiederanfahren an die Kontur mit Halbkreis | s | + | | PM-NC |
| REPOSHA | G | Wiederanfahren an die Kontur mit allen Achsen; Geometrieachsen im Halbkreis | s | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| REPOSL | G | Wiederanfahren an die Kontur linear | s | + | | PM-NC |
| REPOSQ | G | Wiederanfahren an die Kontur im Viertelkreis | s | + | | PM-NC |
| REPOSQA | G | Wiederanfahren an die Kontur linear mit allen Achsen; Geometrieachsen im Viertelkreis | s | + | | PM-NC |
| RESETMON | P | Sprachbefehl zur Sollwertaktivierung | | + | - | FM-TM |
| RET | P | Unterprogrammende | | + | + | PM-NC |
| RETB | P | Unterprogrammende | | + | + | PM-NC |
| RIC | K | Relativ satzweise achsspezifische Radiusprogrammierung | s | + | | PM-NC |
| RINDEX | F | Index eines Zeichens im Eingangsstring bestimmen | | + | - | PM-NC |
| RMB | G | Wiederanfahren an Satzanfangspunkt | m | + | | PM-NC |
| RMBBL | G | Wiederanfahren an Satzanfangspunkt | s | + | | PM-NC |
| RME | G | Wiederanfahren an Satzendpunkt | m | + | | PM-NC |
| RMEBL | G | Wiederanfahren an Satzendpunkt | s | + | | PM-NC |
| RMI [6] | G | Wiederanfahren an Unterbrechungspunkt | m | + | | PM-NC |
| RMIBL [6] | G | Wiederanfahren an Unterbrechungspunkt | s | + | | PM-NC |
| RMN | G | Wiederanfahren an nächstliegenden Bahnpunkt | m | + | | PM-NC |
| RMNBL | G | Wiederanfahren an nächstliegenden Bahnpunkt | s | + | | PM-NC |
| RND | A | Konturecke verrunden | s | + | | PM-NC |
| RNDM | A | Modales Verrunden | m | + | | PM-NC |
| ROT | G | Programmierbare Drehung | s | + | | PM-NC |
| ROTS | G | Programmierbare Frame-Drehungen mit Raumwinkeln | s | + | | PM-NC |
| ROUND | F | Runden der Nachkommastellen | | + | + | PM-NC |
| ROUNDUP | F | Aufrunden eines Eingabewerts | | + | + | PM-NC |
| RP | A | Polarradius | m/s | + | | PM-NC |
| RPL | A | Drehung in der Ebene | s | + | | PM-NC |
| RT | K | Parameter für Zugriff auf Framedaten: Drehung | | + | | PM-NC |
| RTLIOF | G | G0 ohne Linearinterpolation (Einzelachsinterpolation) | m | + | | PM-NC |
| RTLION [6] | G | G0 mit Linearinterpolation | m | + | | PM-NC |

## Anweisungen S ... Z

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| S | A | Spindeldrehzahl<br>(bei G4, G96/G961 andere Bedeutung) | m/s | + | + | PM-NC |
| SAVE | PA | Attribut zur Rettung von Informationen bei Unterprogrammaufrufen | | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| SBLOF | P | Einzelsatz unterdrücken | | + | - | PM-NC |
| SBLON | P | Einzelsatzunterdrückung aufheben | | + | - | PM-NC |
| SC | K | Parameter für Zugriff auf Framedaten: Skalierung | | + | | PM-NC |
| SCALE | G | Programmierbare Skalierung | s | + | | PM-NC |
| SCC | K | Selektive Zuordnung einer Planachse zu G96/G961/G962. Achsbezeichner können Geo-, Kanal oder Maschinenachse sein. | | + | | PM-NC |
| SCPARA | K | Servo-Parameterersatz programmieren | | + | + | PM-NC |
| SD | A | Spline-Grad | s | + | | PM-NC |
| SET | K | Schlüsselwort zur Initialisierung aller Elemente eines Feldes mit aufgelisteten Werten | | + | | PM-NC |
| SETAL | P | Alarm setzen | | + | + | PM-NC |
| SETDNO | F | D-Nummer der Schneide (CE) eines Werkzeugs (T) zuordnen | | + | - | PM-NC |
| SETINT | K | Festlegung, welche Interruptroutine aktiviert werden soll, wenn ein NC-Eingang ansteht | | + | | PM-NC |
| SETM | P | Setzen von Markern im eigenen Kanal | | + | + | PM-NC |
| SETMS | P | Zurückschalten auf die im Maschinendatum festgelegte Masterspindel | | + | - | PM-NC |
| SETMS(n) | P | Spindel n soll als Masterspindel gelten | | + | | PM-NC |
| SETMTH | P | Masterwerkzeughalternummer setzen | | + | - | FM-TM |
| SETPIECE | P | Stückzahl für alle Werkzeuge berücksichtigen, die der Spindel zugeordnet sind | | + | - | FM-TM |
| SETTA | P | Werkzeug aus Verschleißverbund aktiv setzen | | + | - | FM-TM |
| SETTCOR | F | Veränderung von Werkzeugkomponenten unter Berücksichtigung aller Randbedingungen | | + | - | PM-NC |
| SETTIA | P | Werkzeug aus Verschleißverbund inaktiv setzen | | + | - | FM-TM |
| SF | A | Startpunktversatz für Gewindeschneiden | m | + | | PM-NC |
| SIN | F | Sinus (Trigon. Funktion) | | + | + | PM-NC |
| SIRELAY | F | Die mit SIRELIN, SIRELOUT und SIRELTIME parametrierten Sicherheitfunktionen aktivieren | | - | + | FM-SI |
| SIRELIN | P | Eingangsgrößen des Funktionsbausteins initialisieren | | + | - | FM-SI |
| SIRELOUT | P | Ausgangsgrößen des Funktionsbausteins initialisieren | | + | - | FM-SI |
| SIRELTIME | P | Timer des Funktionsbausteins initialisieren | | + | - | FM-SI |
| SLOT1 | C (T) | Längsnut | | + | | PM-NC |
| SLOT2 | C (T) | Kreisnut | | + | | PM-NC |
| SOFT | G | Ruckbegrenzte Bahnbeschleunigung | m | + | | PM-NC |
| SOFTA | P | Ruckbegrenzte Achsbeschleunigung für die programmierten Achsen einschalten | | + | - | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| SON | G | Nibbeln EIN | m | + | | PM-NC |
| SONS | G | Nibbeln EIN im IPO-Takt | m | + | | PM-NC |
| SPATH [6] | G | Bahnbezug für FGROUP-Achsen ist Bogenlänge | m | + | | PM-NC |
| SPCOF | P | Masterspindel oder Spindel (n) von Lageregelung in Drehzahlregelung umschalten | m | + | - | PM-NC |
| SPCON | P | Masterspindel oder Spindel (n) von Drehzahlregelung in Lageregelung umschalten | m | + | - | PM-NC |
| SPI | F | Konvertiert Spindelnummer in Achsbezeichner | | + | - | PM-NC |
| SPIF1 [6] | G | Schnelle NC-Ein-/Ausgänge für Stanzen/Nibbeln Byte 1 | m | + | | FM-TE |
| SPIF2 | G | Schnelle NC-Ein-/Ausgänge für Stanzen/Nibbeln Byte 2 | m | + | | FM-TE |
| SPLINEPATH | P | Spline-Verband festlegen | | + | - | PM-NC |
| SPN | A | Anzahl der Teilstrecken pro Satz | s | + | | PM-NC |
| SPOF [6] | G | Hub AUS, Stanzen, Nibbeln AUS | m | + | | PM-NC |
| SPOS | K | Spindelposition | m | + | + | PM-NC |
| SPOSA | K | Spindelposition über Satzgrenzen hinweg | m | + | | PM-NC |
| SPP | A | Länge einer Teilstrecke | m | + | | PM-NC |
| SPRINT | F | Liefert einen Eingangsstring formatiert zurück | | + | | PM-NC |
| SQRT | F | Quadratwurzel (arithmetische Funktion) (square root) | | + | + | PM-NC |
| SR | A | Pendelrückzugsweg für Synchronaktion | s | + | | PM-NC |
| SRA | K | Pendelrückzugsweg bei externem Eingang axial für Synchronaktion | m | + | | PM-NC |
| ST | A | Pendelausfeuerzeit für Synchronaktion | s | + | | PM-NC |
| STA | K | Pendelausfeuerzeit axial für Synchronaktion | m | + | | PM-NC |
| START | P | Starten der ausgewählten Programme in mehreren Kanälen gleichzeitig aus dem laufenden Programm | | + | - | PM-NC |
| STARTFIFO [6] | G | Abarbeiten; parallel dazu Auffüllen des Vorlaufpuffers | m | + | | PM-NC |
| STAT | | Stellung der Gelenke | s | + | | PM-NC |
| STOLF | A | G0-Toleranzfaktor | m | + | | PM-NC |
| STOPFIFO | G | Anhalten der Bearbeitung; Auffüllen des Vorlaufpuffers, bis STARTFIFO erkannt wird, Vorlaufpuffer voll oder Programmende | m | + | | PM-NC |
| STOPRE | P | Vorlaufstopp, bis alle vorbereiteten Sätze vom Hauptlauf abgearbeitet sind | | + | - | PM-NC |
| STOPREOF | P | Vorlaufstopp aufheben | | - | + | FM-SA |
| STRING | K | Datentyp: Zeichenkette | | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| STRINGIS | F | Prüft vorhandenen NC-Sprachumfang und speziell für diesen Befehl gehörende NC-Zyklennamen, Anwendervariablen, Makros und Labelnamen, ob diese existieren, gültig, definiert oder aktiv sind. | | + | - | PM-NC |
| STRLEN | F | Länge eines Strings bestimmen | | + | - | PM-NC |
| SUBSTR | F | Index eines Zeichens im Eingangsstring bestimmen | | + | - | PM-NC |
| SUPA | G | Unterdrückung der aktuellen Nullpunktverschiebung, einschließlich programmierter Verschiebungen, Systemframes, Handradverschiebungen (DRF), externer Nullpunktverschiebung und überlagerte Bewegung | s | + | | PM-NC |
| SUPD | G | Unterdrückung der aktiven Werkzeugkorrekturen | s | + | - | PM-NC |
| SVC | K | Werkzeug-Schnittgeschwindigkeit | m | + | | PM-NC |
| SYNFCT | P | Auswertung eines Polynoms abhängig von einer Bedingung in der Bewegungssynchronaktion | | - | + | FM-SA |
| SYNR | K | Lesen der Variable erfolgt synchron, d. h. zum Abarbeitungszeitpunkt | | + | | PM-NC |
| SYNRW | K | Lesen und Schreiben der Variable erfolgt synchron, d. h. zum Abarbeitungszeitpunkt | | + | | PM-NC |
| SYNW | K | Schreiben der Variable erfolgt synchron, d. h. zum Abarbeitungszeitpunkt | | + | | PM-NC |
| T | A | Werkzeug aufrufen<br>(wechseln nur, wenn im Maschinendatum festgelegt; ansonsten M6-Befehl nötig) | | + | | PM-NC |
| TAN | F | Tangens (Trigon. Funktion) | | + | + | PM-NC |
| TANG | P | Tangentialsteuerung: Kopplung definieren | | + | - | PM-NC |
| TANGDEL | P | Tangentialsteuerung: Kopplung löschen | | + | - | PM-NC |
| TANGOF | P | Tangentialsteuerung: Kopplung ausschalten | | + | - | PM-NC |
| TANGON | P | Tangentialsteuerung: Kopplung einschalten | | + | - | PM-NC |
| TCA<br>(828D: _TCA) | P | Werkzeuganwahl / Werkzeugwechsel unabhängig vom Status des Werkzeugs | | + | - | FM-TM |
| TCARR | A | Werkzeugträger (Nummer "m") anfordern | | + | | PM-NC |
| TCI | P | Wechsle Werkzeug aus Zwischenspeicher in das Magazin | | + | - | FM-TM |
| TCOABS [6] | G | Werkzeuglängenkomponenten aus der aktuellen Werkzeugorientierung bestimmen | m | + | | PM-NC |
| TCOFR | G | Werkzeuglängenkomponenten aus der Orientierung des aktiven Frames bestimmen | m | + | | PM-NC |
| TCOFRX | G | Werkzeugorientierung eines aktiven Frames bei der Werkzeugwahl bestimmen, Werkzeug zeigt in X-Richtung | m | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| TCOFRY | G | Werkzeugorientierung eines aktiven Frames bei der Werkzeugwahl bestimmen, Werkzeug zeigt in Y-Richtung | m | + | | PM-NC |
| TCOFRZ | G | Werkzeugorientierung eines aktiven Frames bei der Werkzeugwahl bestimmen, Werkzeug zeigt in Z-Richtung | m | + | | PM-NC |
| THETA | A | Drehwinkel | s | + | | PM-NC |
| TILT | A | Seitwärtswinkel | m | + | | PM-NC |
| TLIFT | P | Tangentialsteuerung: Zwischensatzerzeugung einschalten | | + | - | PM-NC |
| TML | P | Werkzeuganwahl mit Magazin-Platznummer | | + | - | FM-TM |
| TMOF | P | Werkzeugüberwachung abwählen | | + | - | PM-NC |
| TMON | P | Werkzeugüberwachung anwählen | | + | - | PM-NC |
| TO | K | Bezeichnet den Endwert in einer FOR-Zählschleife | | + | | PM-NC |
| TOFF | A | Werkzeuglängen-Offset in Richtung der Werkzeuglängenkomponente, die parallel zu der im Index angegebenen Geometrieachse wirkt. | m | + | | PM-NC |
| TOFFL | A | Werkzeuglängen-Offset in Richtung der Werkzeuglängenkomponente L1, L2 bzw. L3 | m | + | | PM-NC |
| TOFFLR | A | Simultaner Werkzeuglängen- und Werkzeugradius-Offset | m | + | | PM-NC |
| TOFFOF | P | Online-Werkzeuglängenkorrektur rücksetzen | | + | - | PM-NC |
| TOFFON | P | Online-Werkzeuglängenkorrektur aktivieren | | + | - | PM-NC |
| TOFFR | A | Werkzeugradius-Offset | m | + | | PM-NC |
| TOFRAME | G | Z-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten | m | + | | PM-NC |
| TOFRAMEX | G | X-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten | m | + | | PM-NC |
| TOFRAMEY | G | Y-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten | m | + | | PM-NC |
| TOFRAMEZ | G | wie TOFRAME | m | + | | PM-NC |
| TOLOWER | F | Buchstaben eines Strings in Kleinbuchstaben wandeln | | + | - | PM-NC |
| TOOLENV | F | Alle aktuellen Zustände speichern, die für die Bewertung der im Speicher abgelegten Werkzeugdaten von Bedeutung sind | | + | - | PM-NC |
| TOOLGNT | F | Anzahl der Werkzeuge einer Werkzeuggruppe ermitteln | | + | - | FM-TM |
| TOOLGT | F | T-Nummer eines Werkzeugs aus einer Werkzeuggruppe ermitteln | | + | - | FM-TM |
| TOROT | G | Z-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten | m | + | | PM-NC |
| TOROTOF [6] | G | Framedrehungen in Werkzeugrichung AUS | m | + | | PM-NC |
| TOROTX | G | X-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten | m | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| TOROTY | G | Y-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten | m | + | | PM-NC |
| TOROTZ | G | wie TOROT | m | + | | PM-NC |
| TOUPPER | F | Buchstaben eines Strings in Großbuchstaben wandeln | | + | - | PM-NC |
| TOWBCS | G | Verschleißwerte im Basiskoordinatensystem (BKS) | m | + | | PM-NC |
| TOWKCS | G | Verschleißwerte im Koordinatensystem des Werkzeugkopfes bei kinetischer Transformation (unterscheidet sich vom MKS durch Werkzeugdrehung) | m | + | | PM-NC |
| TOWMCS | G | Verschleißwerte im Maschinen-Koordinatensystem (MKS) | m | + | | PM-NC |
| TOWSTD [6] | G | Grundstellungswert für Korrekturen in der Werkzeuglänge | m | + | | PM-NC |
| TOWTCS | G | Verschleißwerte im Werkzeug-Koordinatensystem (Werkzeugträgerbezugspunkt T an der Werkzeughalteraufnahme) | m | + | | PM-NC |
| TOWWCS | G | Verschleißwerte im Werkstück-Koordinatensystem (WKS) | m | + | | PM-NC |
| TR | K | Verschiebungskomponente einer Frame-Variablen | | + | | PM-NC |
| TRAANG | P | Transformation schräge Achse | | + | - | PM-NC |
| TRACON | P | Kaskadierte Transformation | | + | - | PM-NC |
| TRACYL | P | Zylinder: Mantelflächen-Transformation | | + | - | PM-NC |
| TRAFOOF | P | Im Kanal aktive Transformationen ausschalten | | + | - | PM-NC |
| TRAFOON | P | Mit kinematischen Ketten definierte Transformation einschalten | | + | - | PM-NC |
| TRAILOF | P | Achssynchrones Mitschleppen AUS | | + | + | PM-NC |
| TRAILON | P | Achssynchrones Mitschleppen EIN | | + | + | PM-NC |
| TRANS | G | Absolute programmierbare Nullpunktverschiebung | s | + | | PM-NC |
| TRANSMIT | P | Polar-Transformation (Stirnflächenbearbeitung) | | + | - | PM-NC |
| TRAORI | P | 4-, 5-Achstransformation, Generische Transformation | | + | - | PM-NC |
| TRUE | K | Logische Konstante: wahr | | + | | PM-NC |
| TRUNC | F | Abschneiden der Nachkommastellen | | + | + | PM-NC |
| TU | | Achswinkel | s | + | | PM-NC |
| TURN | A | Windungsanzahl für Schraubenlinie | s | + | | PM-NC |
| ULI | K | Oberer Grenzwert von Variablen | | + | | PM-NC |
| UNLOCK | P | Synchronaktion mit ID freigeben (Technologiezyklus fortsetzen) | | - | + | FM-SA |
| UNTIL | K | Bedingung zur Beendigung einer REPEAT-Schleife | | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| UPATH | G | Bahnbezug für FGROUP-Achsen ist Kurvenparameter | m | + | | PM-NC |
| VAR | K | Schlüsselwort: Art der Parameterübergabe | | + | | PM-NC |
| VELOLIM | K | Maximale Achsgeschwindigkeit oder Spindeldrehzahl anpassen | m | + | - | PM-NC |
| VELOLIMA | K | Reduktion oder Überhöhung der maximalen Geschwindigkeit einer Folgeachse | m | + | + | PM-NC |
| WAITC | P | Warten, bis Kopplungssatzwechselkriterium für die Achsen/Spindeln erfüllt ist | | + | - | PM-NC |
| WAITE | P | Warten auf das Programmende in einem anderen Kanal. | | + | - | PM-NC |
| WAITENC | P | Warten auf synchronisierte bzw. restaurierte Achspositionen | | + | - | PM-NC |
| WAITM | P | Warten auf Marker im angegebenen Kanal; vorhergehenden Satz mit Genauhalt beenden. | | + | - | PM-NC |
| WAITMC | P | Warten auf Marker im angegeben. Kanal; Genauhalt nur, wenn die anderen Kanäle den Marker noch nicht erreicht haben. | | + | - | PM-NC |
| WAITP | P | Warten auf Verfahrende der Positionierachse | | + | - | PM-NC |
| WAITS | P | Warten auf Erreichen der Spindelposition | | + | - | PM-NC |
| WALCS0 [6] | G | WKS-Arbeitsfeldbegrenzung abgewählt | m | + | - | PM-NC |
| WALCS1 | G | WKS-Arbeitsfeldbegrenzungsgruppe 1 aktiv | m | + | - | PM-NC |
| WALCS2 | G | WKS-Arbeitsfeldbegrenzungsgruppe 2 aktiv | m | + | - | PM-NC |
| WALCS3 | G | WKS-Arbeitsfeldbegrenzungsgruppe 3 aktiv | m | + | - | PM-NC |
| WALCS4 | G | WKS-Arbeitsfeldbegrenzungsgruppe 4 aktiv | m | + | - | PM-NC |
| WALCS5 | G | WKS-Arbeitsfeldbegrenzungsgruppe 5 aktiv | m | + | - | PM-NC |
| WALCS6 | G | WKS-Arbeitsfeldbegrenzungsgruppe 6 aktiv | m | + | - | PM-NC |
| WALCS7 | G | WKS-Arbeitsfeldbegrenzungsgruppe 7 aktiv | m | + | - | PM-NC |
| WALCS8 | G | WKS-Arbeitsfeldbegrenzungsgruppe 8 aktiv | m | + | - | PM-NC |
| WALCS9 | G | WKS-Arbeitsfeldbegrenzungsgruppe 9 aktiv | m | + | - | PM-NC |
| WALCS10 | G | WKS-Arbeitsfeldbegrenzungsgruppe 10 aktiv | m | + | - | PM-NC |
| WALIMOF | G | BKS-Arbeitsfeldbegrenzung AUS | m | + | - | PM-NC |
| WALIMON [6] | G | BKS-Arbeitsfeldbegrenzung EIN | m | + | - | PM-NC |
| WHEN | K | Die Aktion wird einmal ausgeführt, wenn die Bedingung einmal erfüllt ist. | | - | + | FM-SA |
| WHENEVER | K | Die Aktion wird zyklisch in jedem Interpolatortakt ausgeführt, wenn die Bedingung erfüllt ist. | | - | + | FM-SA |
| WHILE | K | Beginn der WHILE-Programmschleife | | + | | PM-NC |
| WRITE | P | Text ins Dateisystem schreiben.<br>Fügt einen Satz am Ende der angegebenen Datei an. | | + | - | PM-NC |
| WRTPR | P | Zeichenkette in BTSS-Variable schreiben | | + | - | PM-NC |
| X | A | Achsname | m/s | + | + | PM-NC |
| XOR | O | Logisches Exklusiv- ODER | | + | | PM-NC |

| Anweisung | Art [1] | Bedeutung | W [2] | TP [3] | SA [4] | Beschreibung siehe [5] |
|---|---|---|---|---|---|---|
| [1] [2] [3] [4] [5] Erläuterungen siehe Legende (Seite 1215). | | | | | | |
| Y | A | Achsname | m/s | + | + | PM-NC |
| Z | A | Achsname | m/s | + | + | PM-NC |

## Legende

[1] Art der Anweisung:

- A Adresse
  Bezeichner, dem ein Wert zugewiesen wird (z. B. OVR=10). Es gibt auch einige Adressen, die ohne Wertzuweisung eine Funktion ein- oder ausschalten (z. B. CPLON und CPLOF).
- C (A) AST-Zyklus
  Vordefiniertes NC-Programm für das automatische Nachoptimieren mit AST (= automatische Servo Optimierung). Die Anpassung an die konkrete Optimierungssituation erfolgt über Parameter, die dem Zyklus beim Aufruf übergeben werden.
- C (M) Messzyklus
  Vordefiniertes NC-Programm, in dem ein bestimmter Messvorgang, wie z. B. das Bestimmen des Innendurchmessers eines zylindrischen Werkstücks, allgemeingültig programmiert ist. Die Anpassung an die konkrete Messsituation erfolgt über Parameter, die dem Zyklus beim Aufruf übergeben werden.
- C (T) Technologischer Zyklus
  Vordefiniertes NC-Programm, in dem ein bestimmter Bearbeitungsvorgang, wie z. B. das Bohren eines Gewindes oder das Fräsen einer Tasche, allgemeingültig programmiert ist. Die Anpassung an die konkrete Bearbeitungssituation erfolgt über Parameter, die dem Zyklus beim Aufruf übergeben werden.
- F Vordefinierte Funktion (liefert Rückgabewert)
  Der Aufruf der vordefinierten Funktion kann als Operand im Ausdruck stehen.
- G G-Befehl
  G-Befehle sind in Gruppen eingeteilt. Es kann nur ein G-Befehl einer Gruppe in einem Satz geschrieben werden. Ein G-Befehl kann modal wirksam sein (bis auf Widerruf durch einen anderen Befehl derselben Gruppe), oder er ist nur für den Satz wirksam, in dem er steht (satzweise wirksam).
- K Schlüsselwort
  Bezeichner, der die Syntax eines Satzes bestimmt. Einem Schlüsselwort wird kein Wert zugewiesen und mit einem Schlüsselwort kann auch keine NC-Funktion ein-/ausgeschaltet werden.
  Beispiele: Kontrollstrukturen (IF, ELSE, ENDIF, WHEN, ...), Programmablauf (GOTOB, GOTO, RET …)
- O Operator
  Operator für eine mathematische, Vergleichs- oder logische Operation
- P Vordefinierte Prozedur (liefert keinen Rückgabewert)
- PA Programmattribut
  Programmattribute stehen am Ende der Definitionszeile eines Unterprogramms:
  `PROC <Programmname>(...) <Programmattribut>`
  Sie bestimmen das Verhalten beim Ablauf des Unterprogramms.

[2] Wirksamkeit der Anweisung:

- m modal
- s satzweise

[3] Programmierbarkeit im Teileprogramm:

- \+ programmierbar
- \- nicht programmierbar
- M nur vom Maschinenhersteller programmierbar

[4] Programmierbarkeit in Synchronaktionen:

| | |
|---|---|
| + | programmierbar |
| - | nicht programmierbar |
| T | nur in Technologiezyklen programmierbar |

[5] Verweis auf das Dokument, das die ausführliche Beschreibung der Anweisung enthält:

| | |
|---|---|
| FM-A | Funktionshandbuch Achsen und Spindeln |
| FM-B | Funktionshandbuch Basisfunktionen |
| FM-SA | Funktionshandbuch Synchronaktionen |
| FM-SI | Funktionshandbuch Safety Integrated |
| FM-TE | Funktionshandbuch Technologien |
| FM-TM | Funktionshandbuch Werkzeugverwaltung |
| FM-TR | Funktionshandbuch Transformationen |
| PM-MC | Programmierhandbuch Messzyklen |
| PM-NC | Programmierhandbuch NC-Programmierung |

[6] Standardeinstellung bei Programmanfang (im Auslieferungsstand der Steuerung, wenn nichts anderes programmiert ist).

# 4.2 Adressen

## 4.2.1 Adressbuchstaben

| Buchstabe | Bedeutung | numerische Erweiterung |
|---|---|---|
| A | einstellbarer Adressbezeichner | x |
| B | einstellbarer Adressbezeichner | x |
| C | einstellbarer Adressbezeichner | x |
| D | An-/Abwahl der Werkzeuglängenkorrektur, Werkzeugschneide | |
| E | einstellbarer Adressbezeichner | x |
| F | Vorschub<br>Verweilzeit in Sekunden | x |
| G | G-Befehl | |
| H | H-Funktion | x |
| I | einstellbarer Adressbezeichner | x |
| J | einstellbarer Adressbezeichner | x |
| K | einstellbarer Adressbezeichner | x |
| L | Unterprogrammname, -aufruf | |
| M | M-Funktion | x |
| N | Nebensatz-Nummer | |
| O | frei | |
| P | Programmdurchlaufzahl | |
| Q | einstellbarer Adressbezeichner | x |
| R | Variablen-Bezeichner (R-Parameter)<br>einstellbarer Adressbezeichner (ohne numerische Erweiterung) | x |
| S | Spindel-Wert<br>Verweilzeit in Spindelumdrehungen | x<br>x |
| T | Werkzeugnummer | x |
| U | einstellbarer Adressbezeichner | x |
| V | einstellbarer Adressbezeichner | x |
| W | einstellbarer Adressbezeichner | x |
| X | einstellbarer Adressbezeichner | x |
| Y | einstellbarer Adressbezeichner | x |
| Z | einstellbarer Adressbezeichner | x |
| % | Anfangs- und Trennzeichen bei der Übertragung von Dateien | |
| : | Hauptsatznummer | |
| / | Ausblendkennung | |

## 4.2.2 Feste Adressen

### Feste Adressen ohne axiale Erweiterung

| Adressbezeichner | Adresstyp | modal/satzweise | G70/G71 | G700/G710 | G90/G91 | IC | AC | DC, ACN, ACP | CIC, CAC, CDC, CACN, CACP | QU | Datentyp des zugewiesenen Werts |
|---|---|---|---|---|---|---|---|---|---|---|---|
| D | Korrekturnummer | m | | | | | | | | x | vorzeichenlos INT |
| F | Vorschub, Verweilzeit | m, s | x | | | | | | | x | vorzeichenlos REAL |
| G | G-Befehl | siehe Liste der G-Funktionen | | | | | | | | | vorzeichenlos INT |
| H | Hilfsfunktionen | s | | | | | | | | x | M: vorzeichenlos INT<br>H: REAL |
| L | Unterprogrammnummer | s | | | | | | | | | vorzeichenlos INT |
| M | Hilfsfunktionen | s | | | | | | | | x | M: vorzeichenlos INT<br>H: REAL |
| N | Satznummer | s | | | | | | | | | vorzeichenlos INT |
| OVR | Override | m | | | | | | | | | vorzeichenlos REAL |
| OVRRAP | Override für Eilganggeschwindigkeit | m | | | | | | | | | vorzeichenlos REAL |
| P | Unterprogrammdurchlaufzahl | s | | | | | | | | | vorzeichenlos INT |
| S | Spindel, Verweilzeit | m, s | | | | | | | | x | vorzeichenlos REAL |
| SCC | Zuordnung einer Planachse zu G96 /G961/G962 | m | | | | | | | | | REAL |

| Adressbezeichner | Adresstyp | modal/satzweise | G70/G71 | G700/G710 | G90/G91 | IC | AC | DC, ACN, ACP | CIC, CAC, CDC, CACN, CACP | QU | Datentyp des zugewiesenen Werts |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SPOS | Spindelposition | m | | | | x | x | x | | | REAL |
| SPOSA | Spindelposition über Satzgrenzen hinweg | m | | | | x | x | x | | | REAL |
| T | Werkzeugnummer | m | | | | | | | | x | vorzeichenlos INT |

## Feste Adressen mit axialer Erweiterung

| Adressbezeichner | Adresstyp | modal/satzweise | G70/G71 | G700/G710 | G90/G91 | IC | AC | DC, ACN, ACP | CIC, CAC, CDC, CACN, CACP | QU | Datentyp des zugewiesenen Werts |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ACC | Axiale Beschleunigung | m | | | | | | | | | vorzeichenlos REAL |
| ACCLIMA | Axiale Beschleunigungsbegrenzung Folgeachse | m | | | | | | | | | vorzeichenlos REAL |
| AX | Variabler Achsbezeichner | [1] | x | x | x | x | x | x | | | REAL |
| FA | Axialer Vorschub | m | x | | | | | | | x | vorzeichenlos REAL |
| FDA | Axialer Vorschub für Handradüberlagerung | s | x | | | | | | | | vorzeichenlos REAL |
| FGREF | Bezugsradius | m | x | x | | | | | | | vorzeichenlos REAL |
| FL | Axialer Grenz-Vorschub | m | x | | | | | | | | vorzeichenlos REAL |
| FMA | Axialer Synchron-Vorschub | m | | | | | | | | | vorzeichenlos REAL |
| FOC | Satzweises Fahren mit begrenztem Moment | s | | | | | | | | | REAL |

| Adressbezeichner | Adresstyp | modal/satzweise | G70/G71 | G700/G710 | G90/G91 | IC | AC | DC, ACN, ACP | CIC, CAC, CDC, CACN, CACP | QU | Datentyp des zugewiesenen Werts |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FOCOF | Modales Fahren mit begrenztem Moment AUS | m | | | | | | | | | REAL |
| FOCON | Modales Fahren mit begrenztem Moment EIN | m | | | | | | | | | REAL |
| FXS | Fahren auf Festanschlag EIN | m | | | | | | | | | vorzeichenlos INT |
| FXST | Momentgrenze für Fahren auf Festanschlag | m | | | | | | | | | REAL |
| FXSW | Überwachungsfenster für Fahren auf Festanschlag | m | | | | | | | | | REAL |
| IP | Variabler Interpolationsparameter | s | x | x | x | x | x | | | | REAL |
| JERKLIM | axiale Ruckbegrenzung | m | | | | | | | | | vorzeichenlos REAL |
| JERKLIMA | axiale Ruckbegrenzung Folgeachse | m | | | | | | | | | vorzeichenlos REAL |
| MEAC | Zyklisches Messen | s | | | | | | | | | INT<br>Modus und<br>1 - 4 Triggerereignisse |
| MEASA | Axiales Messen mit Restweglöschen | s | | | | | | | | | INT<br>Modus und<br>1 - 4 Triggerereignisse |
| MEAWA | Axiales Messen ohne Restweglöschen | s | | | | | | | | | INT<br>Modus und<br>1 - 4 Triggerereignisse |
| MOV | Positionierachse starten | m | x | x | x | x | x | x | x | | REAL |
| OS | Pendeln EIN/AUS | m | | | | | | | | | vorzeichenlos INT |

| Adressbezeichner | Adresstyp | modal/satzweise | G70/G71 | G700/G710 | G90/G91 | IC | AC | DC, ACN, ACP | CIC, CAC, CDC, CACN, CACP | QU | Datentyp des zugewiesenen Werts |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSB | Pendeln Startpunkt | m | x | x | x | x | x | x | | | REAL |
| OSCILL | Achszuordnung für Pendeln, Pendeln einschalten | m | | | | | | | | | Axis:<br>1 - 3 Zustellachsen |
| OSCTRL | Optionen Pendeln | m | | | | | | | | | vorzeichenlos INT: Setzoptionen, vorzeichenlos INT: Rücksetzoptionen |
| OSE | Pendeln Endpunkt | m | x | x | x | x | x | x | | | REAL |
| OSNSC | Ausfunkanzahl Pendeln | m | | | | | | | | | vorzeichenlos INT |
| OSP1 | Linker Umkehrpunkt (Pendeln) | m | x | x | x | x | x | x | | | REAL |
| OSP2 | Rechter Umkehrpunkt (Pendeln) | m | x | x | x | x | x | x | | | REAL |
| OST1 | Haltezeit im linken Umkehrpunkt (Pendeln) | m | | | | | | | | | REAL |
| OST2 | Haltezeit im rechten Umkehrpunkt (Pendeln) | m | | | | | | | | | REAL |
| OVRA | Axialer Override | m | x | | | | | | | | vorzeichenlos REAL |
| PO | Polynomkoeffizient | s | x | x | | x | x | x | | | vorzeichenlos REAL |
| POLF | Position LIFTFAST | m | x | x | | | | | | | vorzeichenlos REAL |
| POS | Positionierachse | m | x | x | x | x | x | x | x | | REAL |
| POSA | Positionierachse über Satzgrenzen | m | x | x | x | x | x | x | x | | REAL |

| Adressbezeichner | Adresstyp | modal/satzweise | G70/G71 | G700/G710 | G90/G91 | IC | AC | DC, ACN, ACP | CIC, CAC, CDC, CACN, CACP | QU | Datentyp des zugewiesenen Werts |
|---|---|---|---|---|---|---|---|---|---|---|---|
| POSP | Positionieren in Teilstücken (Pendeln) | m | x | x | x | x | x | x | | | REAL: Endposition<br>Real: Teillänge<br>INT: Option |
| STA | Ausfeuerzeit (axial) | m | | | | | | | | | vorzeichenlos REAL |
| SRA | Rückzugsweg bei externem Eingang (axial) | m | | | | | | | | | vorzeichenlos REAL |
| VELOLIM | Axiale Geschwindigkeitsbegrenzung | m | | | | | | | | | vorzeichenlos REAL |
| VELOLIMA | Axiale Geschwindigkeitsbegrenzung Folgeachse | m | | | | | | | | | vorzeichenlos REAL |

[1] absolute Endpunkte: modal, inkrementelle Endpunkte: satzweise, ansonsten modal/satzweise in Abhängigkeit vom Syntax-bestimmenden G-Befehl

## 4.2.3 Einstellbare Adressen

| Adressbezeichner (Standardeinstellung) | Adresstyp | modal/satzweise | G90/G91 | IC | AC | DC, ACN, ACP | CIC, CAC, CDC, CACN, CACP | PR, PM | QU | Max. Anzahl | Datentyp des zugewiesenen Werts |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Achswerte und Endpunkte** | | | | | | | | | | | |
| X, Y, Z, A, B, C | Achse | [1] | x | x | x | x | | | | 8 | REAL |
| AP | Polarwinkel | m/s [1] | x | x | x | | | | | 1 | REAL |
| RP | Polarradius | m/s [1] | x | x | x | | | | | 1 | vorzeichenlos REAL |
| **Werkzeugorientierung** | | | | | | | | | | | |
| A2, B2, C2 | Eulerwinkel oder RPY-Winkel | s | | | | | | | | 3 | REAL |

| Adressbezeichner (Standardeinstellung) | Adresstyp | modal/satzweise | G90/G91 | IC | AC | DC, ACN, ACP | CIC, CAC, CDC, CACN, CACP | PR, PM | QU | Max. Anzahl | Datentyp des zugewiesenen Werts |
|---|---|---|---|---|---|---|---|---|---|---|---|
| A3, B3, C3 | Komponenten des Richtungsvektors | s | | | | | | | | 3 | REAL |
| A4, B4, C4 | Komponenten des Flächennormalenvektors am Satzanfang | s | | | | | | | | 3 | REAL |
| A5, B5, C5 | Komponenten des Flächennormalenvektors am Satzende | s | | | | | | | | 3 | REAL |
| A6, B6, C6 | Komponenten des Richtungsvektors für die Drehachse des Kegels | s | | | | | | | | 3 | REAL |
| A7, B7, C7 | Vektorkomponenten für die Zwischenorientierung auf der Kegelmantelfläche | s | | | | | | | | 3 | REAL |
| LEAD | Voreilwinkel | m | | | | | | | | 1 | REAL |
| THETA | Drehwinkel Drehung um die WZ-Richtung | m | | x | x | | | | | 1 | REAL |
| TILT | Seitwärtswinkel | m | | | | | | | | 1 | REAL |
| ORIS | Orientierungsänderung (bezogen auf die Bahn) | m | | | | | | | | 1 | REAL |
| **Interpolationsparameter** | | | | | | | | | | | |
| I, J, K | Interpolationsparameter Zwischenpunktkoordinate | s | | x [2)] | x [2)] | | | | | 3 | REAL |
| I1, J1, K1 | | s | x | x | x | | | | | 3 | REAL |
| RPL | Drehung in der Ebene | s | | | | | | | | 1 | REAL |
| CR | Kreisradius | s | | | | | | | | 1 | vorzeichenlos REAL |
| AR | Öffnungswinkel | s | | | | | | | | 1 | vorzeichenlos REAL |
| TURN | Windungsanzahl für Schraubenlinie | s | | | | | | | | 1 | vorzeichenlos INT |

| Adressbezeichner (Standardeinstellung) | Adresstyp | modal/satzweise | G90/G91 | IC | AC | DC, ACN, ACP | CIC, CAC, CDC, CACN, CACP | PR, PM | QU | Max. Anzahl | Datentyp des zugewiesenen Werts |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PL | Parameter Intervall-Länge | s | | | | | | | | 1 | vorzeichenlos REAL |
| PW | Punkt-Gewicht | s | | | | | | | | 1 | vorzeichenlos REAL |
| SD | Spline-Grad | m | | | | | | | | 1 | vorzeichenlos INT |
| TU | Achswinkel | s | | | | | | | | 1 | vorzeichenlos INT |
| STAT | Stellung der Gelenke | m | | | | | | | | 1 | vorzeichenlos INT |
| SF | Startpunktversatz für Gewindeschneiden | m | | | | | | | | 1 | REAL |
| DISCL | Sicherheitsabstand WAB | s | | | | | | | | 1 | vorzeichenlos REAL |
| DISR | Repos-Abstand / WAB-Abstand | s | | | | | | | | 1 | vorzeichenlos REAL |
| DISPR | Repos-Bahndifferenz | s | | | | | | | | 1 | vorzeichenlos REAL |
| ALF | Schnellabhebe-Winkel | m | | | | | | | | 1 | vorzeichenlos INT |
| DILF | Schnellabhebe-Länge | m | | | | | | | | 1 | REAL |
| FP | Festpunkt: Nummer des anzufahrenden Festpunkts | s | | | | | | | | 1 | vorzeichenlos INT |
| RNDM | Rundung modal | m | | | | | | | | 1 | vorzeichenlos REAL |
| RND | Rundung satzweise | s | | | | | | | | 1 | vorzeichenlos REAL |
| CHF | Fase satzweise | s | | | | | | | | 1 | vorzeichenlos REAL |
| CHR | Fase in ursprünglicher Bewegungsrichtung | s | | | | | | | | 1 | vorzeichenlos REAL |
| ANG | Konturzug-Winkel | s | | | | | | | | 1 | REAL |
| ISD | Eintauchtiefe | m | | | | | | | | 1 | REAL |
| DISC | Überhöhung Übergangskreis Werkzeugradiuskorrektur | m | | | | | | | | 1 | vorzeichenlos REAL |

| Adressbezeichner (Standardeinstellung) | Adresstyp | modal/satzweise | G90/G91 | IC | AC | DC, ACN, ACP | CIC, CAC, CDC, CACN, CACP | PR, PM | QU | Max. Anzahl | Datentyp des zugewiesenen Werts |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OFFN | Offset Kontur - normal | m | | | | | | | | 1 | REAL |
| DITS | Gewindeeinlaufweg | m | | | | | | | | 1 | REAL |
| DITE | Gewindeauslaufweg | m | | | | | | | | 1 | REAL |
| **Überschleifkriterien** | | | | | | | | | | | |
| ADIS | Überschleifabstand | m | | | | | | | | 1 | vorzeichenlos REAL |
| ADISPOS | Überschleifabstand für Eilgang | m | | | | | | | | 1 | vorzeichenlos REAL |
| **Messen** | | | | | | | | | | | |
| MEAS | Messen mit schaltendem Taster | s | | | | | | | | 1 | vorzeichenlos INT |
| MEAW | Messen mit schaltendem Taster ohne Restweglöschen | s | | | | | | | | 1 | vorzeichenlos INT |
| **Achs-, Spindelverhalten** | | | | | | | | | | | |
| LIMS | Spindeldrehzahlbegrenzung | m | | | | | | | | 1 | vorzeichenlos REAL |
| COARSEA | Satzwechselverhalten: Genauhalt grob axial | m | | | | | | | | | |
| FINEA | Satzwechselverhalten: Genauhalt fein axial | m | | | | | | | | | |
| IPOENDA | Satzwechselverhalten: Interpolator-Stopp axial | m | | | | | | | | | |
| DIACYCOFA | Planachse: axiale Durchmesserprogrammierung AUS in Zyklen | m | | | | | | | | | |
| DIAM90A | Planachse: axiale Durchmesserprogrammierung bei G90 | m | | | | | | | | | |

| Adressbezeichner (Standardeinstellung) | Adresstyp | modal/satzweise | G90/G91 | IC | AC | DC, ACN, ACP | CIC, CAC, CDC, CACN, CACP | PR, PM | QU | Max. Anzahl | Datentyp des zugewiesenen Werts |
|---|---|---|---|---|---|---|---|---|---|---|---|
| DIAMCHAN | Planachse: Übernahme aller Planachsen in den Kanalzustand der Durchmesserprogrammierung | m | | | | | | | | | |
| DIAMCHANA | Planachse: Übernahme Kanalzustand der Durchmesserprogrammierung | m | | | | | | | | | |
| DIAMOFA | Planachse: axiale Durchmesserprogrammierung AUS | m | | | | | | | | | |
| DIAMONA | Planachse: axiale Durchmesserprogrammierung EIN | m | | | | | | | | | |
| GP | Position: Indirekte Programmierung von Positionsattributen | m | | | | | | | | | |
| **Vorschübe** | | | | | | | | | | | |
| FAD | Geschwindigkeit der langsamen Zustellbewegung | s | | | | | | x | | 1 | vorzeichenlos REAL |
| FD | Bahnvorschub für Handradüberlagerung | s | | | | | | | | 1 | vorzeichenlos REAL |
| FRC | Vorschub für Radius und Fase | s | | | | | | | | 1 | vorzeichenlos REAL |
| FRCM | Vorschub für Radius und Fase modal | m | | | | | | | | 1 | vorzeichenlos REAL |
| FB | Satzweiser Vorschub | s | | | | | | | | 1 | vorzeichenlos REAL |
| **Nibbeln / Stanzen** | | | | | | | | | | | |
| SPN | Anzahl der Teilstrecken pro Satz | s | | | | | | | | 1 | INT |
| SPP | Länge einer Teilstrecke | m | | | | | | | | 1 | REAL |

| Adressbezeichner (Standardeinstellung) | Adresstyp | modal/satzweise | G90/G91 | IC | AC | DC, ACN, ACP | CIC, CAC, CDC, CACN, CACP | PR, PM | QU | Max. Anzahl | Datentyp des zugewiesenen Werts |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Schleifen** | | | | | | | | | | | |
| ST | Ausfeuerzeit | s | | | | | | | | 1 | vorzeichenlos REAL |
| SR | Rückzugsweg | s | | | | | | | | 1 | vorzeichenlos REAL |
| **Werkzeuganwahl** | | | | | | | | | | | |
| TCARR | Werkzeugträger | m | | | | | | | | 1 | INT |
| **Werkzeugverwaltung** | | | | | | | | | | | |
| DL | Werkzeugsummenkorrektur | m | | | | | | | | 1 | INT |
| **OEM Adressen** | | | | | | | | | | | |
| OMA1 | OEM-Adresse 1 | m | | x | x | x | | | | 1 | REAL |
| OMA2 | OEM-Adresse 2 | m | | x | x | x | | | | 1 | REAL |
| OMA3 | OEM-Adresse 3 | m | | x | x | x | | | | 1 | REAL |
| OMA4 | OEM-Adresse 4 | m | | x | x | x | | | | 1 | REAL |
| OMA5 | OEM-Adresse 5 | m | | x | x | x | | | | 1 | REAL |
| **Sonstiges** | | | | | | | | | | | |
| CUTMOD | Modifikation der Korrekturdaten bei drehbaren Werkzeugen (in Verbindung mit orientierbaren Werkzeugträgern) | m | | | | | | | | | INT |
| CUTMODK | Modifikation der Korrekturdaten bei drehbaren Werkzeugen (in Verbindung mit Orientierungstransformatione, die mittels kinematischer Ketten definiert wurden) | m | | | | | | | | | STRING |
| TOFF | Werkzeuglängen-Offset parallel zur angegebenen Geometrieachse | m | | | | | | | | | REAL |

| Adressbezeichner (Standardeinstellung) | Adresstyp | modal/satzweise | G90/G91 | IC | AC | DC, ACN, ACP | CIC, CAC, CDC, CACN, CACP | PR, PM | QU | Max. Anzahl | Datentyp des zugewiesenen Werts |
|---|---|---|---|---|---|---|---|---|---|---|---|
| TOFFL | Werkzeuglängen-Offset in Richtung der WZ-Längenkomponente L1, L2 bzw. L3 | m | | | | | | | | | REAL |
| TOFFR | Werkzeugradius-Offset | m | | | | | | | | | REAL |
| TOFFLR | Simultaner Werkzeuglängen- und Werkzeugradius-Offset | m | | | | | | | | | REAL |

1) absolute Endpunkte: modal, inkrementelle Endpunkte: satzweise, ansonsten modal/satzweise in Abhängigkeit vom syntaxbestimmenden G-Befehl

2) Als Kreismittelpunkte wirken IPO-Parameter inkrementell. Mit AC können sie absolut programmiert werden. Bei anderen Bedeutungen (z. B. Gewindesteigung) wird die Adressmodifikation ignoriert.

# 4.3 G-Befehle

## 4.3.1 G-Befehle

G-Befehle sind in G-Gruppen eingeteilt. In Teileprogrammen oder Synchronaktionen kann in einem Satz nur ein G-Befehl einer G-Gruppe geschrieben werden. Eine G-Befehl kann modal wirksam oder satzweise wirksam sein.

Modal: bis zur Programmierung eines anderen G-Befehles derselben G-Gruppe.

## 4.3.2 G-Gruppe 1: Modal wirksame Bewegungsbefehle

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G0 | 1 | Eilgangsbewegung | + | m | | |
| G1 | 2 | Linearinterpolation (Geradeninterpolation) | + | m | x | |
| G2 | 3 | Kreisinterpolation im Uhrzeigersinn | + | m | | |
| G3 | 4 | Kreisinterpolation gegen Uhrzeigersinn | + | m | | |
| CIP | 5 | Kreisinterpolation über Zwischenpunkt | + | m | | |
| ASPLINE | 6 | Akima-Spline | + | m | | |
| BSPLINE | 7 | B-Spline | + | m | | |
| CSPLINE | 8 | Kubischer Spline | + | m | | |
| POLY | 9 | Polynom-Interpolation | + | m | | |
| G33 | 10 | Gewindeschneiden mit konstanter Steigung | + | m | | |
| G331 | 11 | Gewindebohren | + | m | | |
| G332 | 12 | Rückzug (Gewindebohren) | + | m | | |
| OEMIPO1 | 13 | reserviert | + | m | | |
| OEMIPO2 | 14 | reserviert | + | m | | |
| CT | 15 | Kreis mit tangentialem Übergang | + | m | | |
| G34 | 16 | Gewindeschneiden mit linear zunehmender Steigung | + | m | | |
| G35 | 17 | Gewindeschneiden mit linear abnehmender Steigung | + | m | | |
| INVCW | 18 | Evolventen-Interpolation im Uhrzeigersinn | + | m | | |
| INVCCW | 19 | Evolventen-Interpolation gegen den Uhrzeigersinn | + | m | | |
| G335 | 20 | Drehen eines balligen Gewindes im Uhrzeigersinn | + | m | | |
| G336 | 21 | Drehen eines balligen Gewindes gegen den Uhrzeigersinn | + | m | | |

### 4.3.3 G-Gruppe 2: Satzweise wirksame Bewegungen, Verweilzeit

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G4 | 1 | Verweilzeit, zeitlich vorbestimmt | - | s | | |
| G63 | 2 | Gewindebohren ohne Synchronisation | - | s | | |
| G74 | 3 | Referenzpunktfahren mit Synchronisation | - | s | | |
| G75 | 4 | Festpunkt anfahren | - | s | | |
| REPOSL | 5 | Wiederanfahren an die Kontur linear | - | s | | |
| REPOSQ | 6 | Wiederanfahren an die Kontur im Viertelkreis | - | s | | |
| REPOSH | 7 | Wiederanfahren an die Kontur im Halbkreis | - | s | | |
| REPOSA | 8 | Wiederanfahren an die Kontur linear mit allen Achsen | - | s | | |
| REPOSQA | 9 | Wiederanfahren an die Kontur mit allen Achsen, Geometrieachsen im Viertelkreis | - | s | | |
| REPOSHA | 10 | Wiederanfahren an die Kontur mit allen Achsen, Geometrieachsen im Halbkreis | - | s | | |
| G147 | 11 | Anfahren der Kontur mit Gerade | - | s | | |
| G247 | 12 | Anfahren der Kontur mit Viertelkreis | - | s | | |
| G347 | 13 | Anfahren der Kontur mit Halbkreis | - | s | | |
| G148 | 14 | Verlassen der Kontur mit Gerade | - | s | | |
| G248 | 15 | Verlassen der Kontur mit Viertelkreis | - | s | | |
| G348 | 16 | Verlassen der Kontur mit Halbkreis | - | s | | |
| G5 | 17 | Schrägeinstechschleifen | - | s | | |
| G7 | 18 | Ausgleichsbewegung beim Schrägeinstechschleifen | - | s | | |

### 4.3.4 G-Gruppe 3: Programmierbarer Frame, Arbeitsfeldbegrenzung und Polprogrammierung

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| TRANS | 1 | TRANSLATION: programmierbare Verschiebung | - | s | | |
| ROT | 2 | ROTATION: programmierbare Drehung | - | s | | |
| SCALE | 3 | SCALE: programmierbare Skalierung | - | s | | |
| MIRROR | 4 | MIRROR: programmierbare Spiegelung | - | s | | |
| ATRANS | 5 | Additive TRANSLATION: additive programmierbare Verschiebung | - | s | | |
| AROT | 6 | Additive ROTATION: programmierbare Drehung | - | s | | |
| ASCALE | 7 | Additive SCALE: programmierbare Skalierung | - | s | | |
| AMIRROR | 8 | Additive MIRROR: programmierbare Spiegelung | - | s | | |
| - | 9 | frei | - | - | | |
| G25 | 10 | Minimale Arbeitsfeldbegrenzung/Spindeldrehzahlbegrenzung | - | s | | |

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G26 | 11 | Maximale Arbeitsfeldbegrenzung/Spindeldrehzahlbegrenzung | - | s | | |
| G110 | 12 | Polprogrammierung relativ zur letzten programmierten Sollposition | - | s | | |
| G111 | 13 | Polprogrammierung relativ zum Nullpunkt des aktuellen Werkstückkoordinatensystems | - | s | | |
| G112 | 14 | Polprogrammierung relativ zum letzten gültigen Pol | - | s | | |
| G58 | 15 | Absolute programmierbare Nullpunktverschiebung | - | s | | |
| G59 | 16 | Additive programmierbare Nullpunktverschiebung | - | s | | |
| ROTS | 17 | Rotation mit Raumwinkel | - | s | | |
| AROTS | 18 | Additive Rotation mit Raumwinkel | - | s | | |

## 4.3.5 G-Gruppe 4: FIFO

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| STARTFIFO | 1 | Start FIFO<br>Abarbeiten und parallel dazu Auffüllen des Vorlaufpuffers | + | m | x | |
| STOPFIFO | 2 | Stopp FIFO,<br>Anhalten der Bearbeitung; Füllen des Vorlaufspeichers bis STARTFIFO erkannt wird, Vorlaufspeicher voll oder Programmende | + | m | | |
| FIFOCTRL | 3 | Einschalten der automatischen Vorlaufpuffersteuerung | + | m | | |

## 4.3.6 G-Gruppe 6: Ebenenanwahl

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G17 | 1 | Ebenenanwahl 1. - 2. Geometrieachse | + | m | x | |
| G18 | 2 | Ebenenanwahl 3. - 1. Geometrieachse | + | m | | |
| G19 | 3 | Ebenenanwahl 2. - 3. Geometrieachse | + | m | | |

## 4.3.7 G-Gruppe 7: Werkzeugradiuskorrektur

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G40 | 1 | Keine Werkzeugradiuskorrektur | + | m | x | |
| G41 | 2 | Werkzeugradiuskorrektur links von der Kontur | - | m | | |
| G42 | 3 | Werkzeugradiuskorrektur rechts von der Kontur | - | m | | |

## 4.3.8 G-Gruppe 8: Einstellbare Nullpunktverschiebung

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G500 | 1 | Ausschalten der einstellbaren Nullpunktverschiebung (G54 ... G57, G505 ... G599) | + | m | x | |
| G54 | 2 | 1. einstellbare Nullpunktverschiebung | + | m | | |
| G55 | 3 | 2. einstellbare Nullpunktverschiebung | + | m | | |
| G56 | 4 | 3. einstellbare Nullpunktverschiebung | + | m | | |
| G57 | 5 | 4. einstellbare Nullpunktverschiebung | + | m | | |
| G505 | 6 | 5. einstellbare Nullpunktverschiebung | + | m | | |
| ... | ... | ... | + | m | | |
| G599 | 100 | 99. einstellbare Nullpunktverschiebung | + | m | | |
| Mit den G-Befehlen dieser G-Gruppe wird jeweils ein einstellbarer Anwender-Frame $P_UIFR[ ] aktiviert.<br>G54 entspricht Frame $P_UIFR[1], G505 entspricht Frame $P_UIFR[5].<br>Die Anzahl der einstellbaren Anwender-Frames und damit die Anzahl von G-Befehle dieser G-Gruppe, ist einstellbar über MD28080 $MC_MM_NUM_USER_FRAMES. | | | | | | |

## 4.3.9 G-Gruppe 9: Frame- und Werkzeugkorrektur-Unterdrückung

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G53 | 1 | Unterdrückung der aktuellen Frames: programmierbares Frame einschließlich Systemframe für TOROT und TOFRAME und aktives einstellbares Frame (G54 ... G57, G505 ... G599) | - | s | | |
| SUPA | 2 | Wie G153 inklusive der Unterdrückung der Systemframes für Istwersetzen, Ankratzen, externe Nullpunktverschiebung, PAROT einschließlich Handradverschiebungen (DRF), [externe Nullpunktverschiebung], überlagerte Bewegung | - | s | | |
| G153 | 3 | Wie G53 inklusive der Unterdrückung aller kanalspezifischen und/oder NCU-globalen Basisframes | - | s | | |
| SUPD | 4 | Unterdrückung der aktiven Werkzeugkorrekturen | - | s | | |

## 4.3.10 G-Gruppe 10: Genauhalt - Bahnsteuerbetrieb

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G60 | 1 | Genauhalt | + | m | x | |
| G64 | 2 | Bahnsteuerbetrieb | + | m | | |
| G641 | 3 | Bahnsteuerbetrieb mit Überschleifen nach Wegkriterium (= programmierbarer Überschleifabstand) | + | m | | |
| G642 | 4 | Bahnsteuerbetrieb mit Überschleifen unter Einhaltung definierter Toleranzen | + | m | | |
| G643 | 5 | Bahnsteuerbetrieb mit Überschleifen unter Einhaltung definierter Toleranzen (satzintern) | + | m | | |
| G644 | 6 | Bahnsteuerbetrieb mit Überschleifen mit maximal möglicher Dynamik | + | m | | |
| G645 | 7 | Bahnsteuerbetrieb mit Überschleifen von Ecken und tangentialer Satzübergänge unter Einhaltung definierter Toleranzen | + | m | | |

## 4.3.11 G-Gruppe 11: Genauhalt satzweise

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G9 | 1 | Genauhalt | - | s | | |

## 4.3.12 G-Gruppe 12: Satzwechselkriterien bei Genauhalt (G60/G9)

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G601 | 1 | Satzwechsel bei Genauhalt fein | + | m | x | |
| G602 | 2 | Satzwechsel bei Genauhalt grob | + | m | | |
| G603 | 3 | Satzwechsel bei IPO-Satzende | + | m | | |

## 4.3.13 G-Gruppe 13: Werkstückvermaßung Inch/metrisch

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G70 | 1 | Eingabesystem inch (Längen) | + | m | | |
| G71 | 2 | Eingabesystem metrisch mm (Längen) | + | m | x | |

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G700 | 3 | Eingabesystem inch, inch/min<br>(Längen + Geschwindigkeit + Systemvariable) | + | m | | |
| G710 | 4 | Eingabesystem metrisch mm, mm/min<br>(Längen + Geschwindigkeit + Systemvariable) | + | m | | |

## 4.3.14 G-Gruppe 14: Werkstückvermaßung absolut/inkremental

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G90 | 1 | Absolutmaßangabe | + | m | x | |
| G91 | 2 | Kettenmaßangabe | + | m | | |

## 4.3.15 G-Gruppe 15: Vorschubtyp

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G93 | 1 | Zeitreziproker Vorschub 1/min | + | m | | |
| G94 | 2 | Linearvorschub in mm/min, inch/min | + | m | x | |
| G95 | 3 | Umdrehungsvorschub in mm/U, inch/U | + | m | | |
| G96 | 4 | Umdrehungsvorschub (wie bei G95) und konstante Schnittgeschwindigkeit | + | m | | |
| G97 | 5 | Umdrehungsvorschub und konstante Spindeldrehzahl (konstante Schnittgeschwindigkeit AUS) | + | m | | |
| G931 | 6 | Vorschubvorgabe durch Verfahrzeit, konstante Bahngeschwindigkeit ausschalten | + | m | | |
| G961 | 7 | Linearvorschub (wie bei G94) und konstante Schnittgeschwindigkeit | + | m | | |
| G971 | 8 | Linearvorschub und konstante Spindeldrehzahl (konstante Schnittgeschwindigkeit AUS) | + | m | | |
| G942 | 9 | Linearvorschub und konstante Schnittgeschwindigkeit oder konstante Spindeldrehzahl | + | m | | |
| G952 | 10 | Umdrehungsvorschub und konstante Schnittgeschwindigkeit oder konstante Spindeldrehzahl | + | m | | |
| G962 | 11 | Linearvorschub oder Umdrehungsvorschub und konstante Schnittgeschwindigkeit | + | m | | |
| G972 | 12 | Linearvorschub oder Umdrehungsvorschub und konstante Spindeldrehzahl (konstante Schnittgeschwindigkeit AUS) | + | m | | |
| G973 | 13 | Umdrehungsvorschub ohne Spindeldrehzahlbegrenzung und konstante Spindeldrehzahl (G97 ohne LIMS für ISO-Modus) | + | m | | |

## 4.3.16 G-Gruppe 16: Vorschub-Override an Innen- und Außenkrümmung

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| CFC | 1 | Konstanter Vorschub an der Kontur wirksam bei Innen- und Außenkrümmung | + | m | x | |
| CFTCP | 2 | Konstanter Vorschub in Werkzeugschneidenbezugspunkt (Mittelpunktsbahn) | + | m | | |
| CFIN | 3 | Konstanter Vorschub bei Innenkrümmung, Beschleunigung bei Außenkrümmung | + | m | | |

## 4.3.17 G-Gruppe 17: An-/Abfahrverhalten Werkzeugkorrektur

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| NORM | 1 | Normalenstellung im Anfangs-/Endpunkt | + | m | x | |
| KONT | 2 | Kontur umfahren im Anfangs-/Endpunkt | + | m | | |
| KONTT | 3 | Tangentenstetiges An-/Abfahren | + | m | | |
| KONTC | 4 | Krümmungsstetiges An-/Abfahren | + | m | | |

## 4.3.18 G-Gruppe 18: Eckenverhalten Werkzeugkorrektur

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G450 | 1 | Übergangskreis<br>(Werkzeug umfährt Werkstückecken auf einer Kreisbahn) | + | m | x | |
| G451 | 2 | Schnittpunkt der Äquidistanten<br>(Werkzeug schneidet in der Werkstückecke frei) | + | m | | |

## 4.3.19 G-Gruppe 19: Kurvenübergang bei Spline-Beginn

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| BNAT | 1 | Natürlicher Kurvenübergang zum ersten Spline-Satz | + | m | x | |
| BTAN | 2 | Tangentialer Kurvenübergang zum ersten Spline-Satz | + | m | | |
| BAUTO | 3 | Festlegung des ersten Spline-Abschnitts durch die folgenden 3 Punkte | + | m | | |

## 4.3.20 G-Gruppe 20: Kurvenübergang bei Spline-Ende

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| ENAT | 1 | Natürlicher Kurvenübergang zum nächsten Verfahrsatz | + | m | x | |
| ETAN | 2 | Tangentialer Kurvenübergang zum nächsten Verfahrsatz | + | m | | |
| EAUTO | 3 | Festlegung des letzten Spline-Abschnitts durch die letzten 3 Punkte | + | m | | |

## 4.3.21 G-Gruppe 21: Beschleunigungsprofil

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| BRISK | 1 | Sprungförmige Bahnbeschleunigung | + | m | x | |
| SOFT | 2 | Rückbegrenzte Bahnbeschleunigung | + | m | | |
| DRIVE | 3 | Geschwindigkeitsabhängige Bahnbeschleunigung | + | m | | |

## 4.3.22 G-Gruppe 22: Werkzeugkorrekturtyp

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| CUT2D | 1 | 2½-D-WRK | + | m | x | |
| CUT2DF | 2 | 2½-D-WRK, relativ zum aktuellen Frame (schräge Ebene) | + | m | | |
| CUT3DC | 3 | 3D-WRK für das Umfangsfräsen | + | m | | |
| CUT3DF | 4 | 3D-WRK für das Stirnfräsen mit Orientierungsänderung | + | m | | |
| CUT3DFS | 5 | 3D-WRK für das Stirnfräsen mit konstanter Orientierung. Die Werkzeugorientierung ist durch G17 - G19 festgelegt und wird durch Frames nicht beeinflusst. | + | m | | |
| CUT3DFF | 6 | 3D-WRK für das Stirnfräsen mit konstanter Orientierung. Die Werkzeugorientierung ist die durch G17 - G19 festgelegte und gegebenenfalls durch einen Frame gedrehte Richtung. | + | m | | |
| CUT3DCC | 7 | 3D-WRK für das Umfangsfräsen unter Berücksichtigung einer Begrenzungsfläche mit 3D-Radiuskorrektur: Kontur an der Bearbeitungsfläche | + | m | | |
| CUT3DCCD | 8 | 3D-WRK für das Umfangsfräsen unter Berücksichtigung einer Begrenzungsfläche mit Differenzwerkzeug auf der Werkzeugmittelpunktbahn: Zustellung zur Begrenzungsfläche | + | m | | |

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| CUT2DD | 9 | Auf ein Differenzwerkzeug bezogene 2½-D-WRK | + | m | | |
| CUT2DFD | 10 | Auf ein Differenzwerkzeug bezogene 2½-D-WRK, relativ zum aktuellen Frame (schräge Ebene) | + | m | | |
| CUT3DCD | 11 | Auf ein Differenzwerkzeug bezogene 3D-WRK für das Umfangsfräsen | + | m | | |
| CUT3DFD | 12 | Auf ein Differenzwerkzeug bezogene 3D-WRK für das Stirnfräsen mit Orientierungsänderung | + | m | | |

## 4.3.23 G-Gruppe 23: Kollisionsüberwachung an Innenkonturen

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| CDOF | 1 | Kollisionsüberwachung AUS | + | m | x | |
| CDON | 2 | Kollisionsüberwachung EIN | + | m | | |
| CDOF2 | 3 | Kollisionsüberwachung AUS beim 3D-Umfangsfräsen | + | m | | |

## 4.3.24 G-Gruppe 24: Vorsteuerung

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| FFWOF | 1 | Vorsteuerung AUS | + | m | x | |
| FFWON | 2 | Vorsteuerung EIN | + | m | | |

## 4.3.25 G-Gruppe 25: Bezug Werkzeugorientierung

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| ORIWKS | 1 | Werkzeugorientierung im Werkstück-Koordinatensystem (WKS) | + | m | x | |
| ORIMKS | 2 | Werkzeugorientierung im Maschinen-Koordinatensystem (MKS) | + | m | | |

## 4.3.26 G-Gruppe 26: Wiederanfahrmodus für REPOS (modal wirksam)

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| RMB | 1 | Wiederanfahren an Satzanfangspunkt | - | m | | |
| RMI | 2 | Wiederanfahren an Unterbrechungspunkt | - | m | x | |
| RME | 3 | Wiederanfahren an Satzendpunkt | - | m | | |
| RMN | 4 | Wiederanfahren an nächstliegenden Bahnpunkt | - | m | | |

## 4.3.27 G-Gruppe 27: Werkzeugkorrektur bei Orientierungsänderung an Außenecken

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| ORIC | 1 | Orientierungsänderungen an Außenecken werden dem einzufügenden Kreissatz überlagert | + | m | x | |
| ORID | 2. | Orientierungsänderungen werden vor dem Kreissatz ausgeführt | + | m | | |

## 4.3.28 G-Gruppe 28: Arbeitsfeldbegrenzung

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| WALIMON | 1 | Arbeitsfeldbegrenzung EIN | + | m | x | |
| WALIMOF | 2 | Arbeitsfeldbegrenzung AUS | + | m | | |

## 4.3.29 G-Gruppe 29: Radius-/Durchmesserprogrammierung

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| DIAMOF | 1 | Modal wirksame kanalspezifische Durchmesserprogrammierung AUS<br>Mit dem Ausschalten wird die kanalspezifische Radiusprogrammierung wirksam. | + | m | x | |
| DIAMON | 2 | Modal wirksame unabhängige kanalspezifische Durchmesserprogrammierung EIN<br>Die Wirkung ist unabhängig vom programmierten Maßangabe-Modus (G90/G91). | + | m | | |

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| DIAM90 | 3 | Modal wirksame abhängige kanalspezifische Durchmesserprogrammierung EIN<br>Die Wirkung ist abhängig vom programmierten Maßangabe-Modus (G90/G91). | + | m | | |
| DIAMCYCOF | 4 | Modal wirksame kanalspezifische Durchmesserprogrammierung während der Zyklusbearbeitung AUS | + | m | | |

## 4.3.30 G-Gruppe 30: NC-Satz-Kompression

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| COMPOF | 1 | NC-Satz-Kompression AUS | + | m | x | |
| COMPON | 2 | Kompressor-Funktion COMPON EIN | + | m | | |
| COMPCURV | 3 | Kompressor-Funktion COMPCURV EIN | + | m | | |
| COMPCAD | 4 | Kompressor-Funktion COMPCAD EIN | + | m | | |
| COMPSURF | 5 | Kompressor-Funktion COMPSURF EIN | + | m | | |

## 4.3.31 G-Gruppe 31: OEM-G-Befehle

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G810 | 1 | OEM-G-Befehl | - | m | | |
| G811 | 2 | OEM-G-Befehl | - | m | | |
| G812 | 3 | OEM-G-Befehl | - | m | | |
| G813 | 4 | OEM-G-Befehl | - | m | | |
| G814 | 5 | OEM-G-Befehl | - | m | | |
| G815 | 6 | OEM-G-Befehl | - | m | | |
| G816 | 7 | OEM-G-Befehl | - | m | | |
| G817 | 8 | OEM-G-Befehl | - | m | | |
| G818 | 9 | OEM-G-Befehl | - | m | | |
| G819 | 10 | OEM-G-Befehl | - | m | | |
| Zwei G-Gruppen sind für den OEM-Anwender reserviert. Damit gibt er die Programmierung der von ihm eingebrachten Funktionen zur Programmierung nach außen. | | | | | | |

## 4.3.32 G-Gruppe 32: OEM-G-Befehle

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G820 | 1 | OEM - G-Befehl | - | m | | |
| G821 | 2 | OEM - G-Befehl | - | m | | |
| G822 | 3 | OEM - G-Befehl | - | m | | |
| G823 | 4 | OEM - G-Befehl | - | m | | |
| G824 | 5 | OEM - G-Befehl | - | m | | |
| G825 | 6 | OEM - G-Befehl | - | m | | |
| G826 | 7 | OEM - G-Befehl | - | m | | |
| G827 | 8 | OEM - G-Befehl | - | m | | |
| G828 | 9 | OEM - G-Befehl | - | m | | |
| G829 | 10 | OEM - G-Befehl | - | m | | |
| Zwei G-Gruppen sind für den OEM-Anwender reserviert. Damit gibt er die Programmierung der von ihm eingebrachten Funktionen zur Programmierung nach außen. | | | | | | |

## 4.3.33 G-Gruppe 33: Einstellbare Werkzeugfeinkorrektur

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| FTOCOF | 1 | Online wirksame Werkzeugfeinkorrektur AUS | + | m | x | |
| FTOCON | 2 | Online wirksame Werkzeugfeinkorrektur EIN | - | m | | |

## 4.3.34 G-Gruppe 34: Glättung Werkzeugorientierung

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| OSOF | 1 | Glättung Werkzeugorientierung AUS | + | m | x | |
| OSC | 2 | Konstante Glättung Werkzeugorientierung | + | m | | |
| OSS | 3 | Glättung Werkzeugorientierung am Satzende | + | m | | |
| OSSE | 4 | Glättung Werkzeugorientierung am Satzanfang und -ende | + | m | | |
| OSD | 5 | Satzinternes Überschleifen mit Vorgabe der Weglänge | + | m | | |
| OST | 6 | Satzinternes Überschleifen mit Vorgabe der Winkeltoleranz | + | m | | |

## 4.3.35 G-Gruppe 35: Stanzen und Nibbeln

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| SPOF | 1 | Hub US, Stanzen und Nibbeln AUS | + | m | x | |
| SON | 2 | Nibbeln EIN | + | m | | |
| PON | 3 | Stanzen EIN | + | m | | |
| SONS | 4 | Nibbeln EIN im IPO-Takt | - | m | | |
| PONS | 5 | Stanzen EIN im IPO-Takt | - | m | | |

## 4.3.36 G-Gruppe 36: Stanzen mit Verzögerung

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| PDELAYON | 1 | Verzögerung beim Stanzen EIN | + | m | x | |
| PDELAYOF | 2 | Verzögerung beim Stanzen AUS | + | m | | |

## 4.3.37 G-Gruppe 37: Vorschubprofil

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| FNORM | 1 | Vorschub normal nach DIN66025 | + | m | x | |
| FLIN | 2 | Vorschub linear veränderlich | + | m | | |
| FCUB | 3 | Vorschub nach kubischem Spline veränderlich | + | m | | |

## 4.3.38 G-Gruppe 38: Zuordnung schnelle Ein-/Ausgänge für Stanzen/Nibbeln

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| SPIF1 | 1 | Schnelle NC-Ein-/Ausgänge für Stanzen/Nibbeln Byte 1 | + | m | x | |
| SPIF2 | 2 | Schnelle NC-Ein-/Ausgänge für Stanzen/Nibbeln Byte 2 | + | m | | |

## 4.3.39 G-Gruppe 39: Programmierbare Konturgenauigkeit

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| CPRECOF | 1 | Programmierbare Konturgenauigkeit AUS | + | m | x | |
| CPRECON | 2 | Programmierbare Konturgenauigkeit EIN | + | m | | |

## 4.3.40 G-Gruppe 40: Werkzeugradiuskorrektur konstant

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| CUTCONOF | 1 | Konstante Werkzeugradiuskorrektur AUS | + | m | x | |
| CUTCONON | 2 | Konstante Werkzeugradiuskorrektur EIN | + | m | | |

## 4.3.41 G-Gruppe 41: Gewindeschneiden unterbrechbar

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| LFOF | 1 | Gewindeschneiden unterbrechbar AUS | + | m | x | |
| LFON | 2 | Gewindeschneiden unterbrechbar EIN | + | m | | |

## 4.3.42 G-Gruppe 42: Werkzeugträger

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| TCOABS | 1 | Werkzeuglängenkomponenten aus der aktuellen Werkzeugorientierung bestimmen | + | m | x | |
| TCOFR | 2 | Werkzeuglängenkomponenten aus der Orientierung des aktiven Frames bestimmen | + | m | | |
| TCOFRZ | 3 | Werkzeugorientierung eines aktiven Frames bei der Werkzeugwahl bestimmen, Werkzeug zeigt in Z-Richtung | + | m | | |
| TCOFRY | 4 | Werkzeugorientierung eines aktiven Frames bei der Werkzeugwahl bestimmen, Werkzeug zeigt in Y-Richtung | + | m | | |
| TCOFRX | 5 | Werkzeugorientierung eines aktiven Frames bei der Werkzeugwahl bestimmen, Werkzeug zeigt in X-Richtung | | m | | |

## 4.3.43 G-Gruppe 43: Anfahrrichtung WAB

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G140 | 1 | Anfahrrichtung WAB festgelegt durch G41/G42 | + | m | x | |
| G141 | 2 | Anfahrrichtung WAB links von der Kontur | + | m | | |
| G142 | 3 | Anfahrrichtung WAB rechts von der Kontur | + | m | | |
| G143 | 4 | Anfahrrichtung WAB tangentenabhängig | + | m | | |

## 4.3.44 G-Gruppe 44: Wegaufteilung WAB

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G340 | 1 | Anfahrsatz räumlich, d. h. Tiefenzustellung und Anfahren in der Ebene in einem Satz | + | m | x | |
| G341 | 2 | Zuerst Zustellen in der senkrechten Achse (Z), dann Anfahren in der Ebene | + | m | | |

## 4.3.45 G-Gruppe 45: Bahnbezug der FGROUP-Achsen

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| SPATH | 1 | Bahnbezug für FGROUP-Achsen ist die Bogenlänge | + | m | x | |
| UPATH | 2 | Bahnbezug für FGROUP-Achsen ist der Kurvenparameter | + | m | | |

## 4.3.46 G-Gruppe 46: Ebenenanwahl für Schnellabheben

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| LFTXT | 1 | Ebene wird bestimmt aus der Bahntangente und der aktuellen Werkzeugorientierung | + | m | x | |
| LFWP | 2 | Ebene wird bestimmt durch die aktuelle Arbeitsebene (G17/G18/G19) | + | m | | |
| LFPOS | 3 | Axiales Abheben auf eine Position | + | m | | |

## 4.3.47 G-Gruppe 47: Modus-Umschaltung für externen NC-Code

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G290 | 1 | SINUMERIK-Sprachmodus aktivieren | + | m | x | |
| G291 | 2 | ISO-Sprachmodus aktivieren | + | m | | |

## 4.3.48 G-Gruppe 48: An- / Abfahrverhalten bei Werkzeugradiuskorrektur

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| G460 | 1 | Kollisionsüberwachung für An- und Abfahrsatz EIN | + | m | x | |
| G461 | 2 | Randsatz mit Kreisbogen verlängern, wenn kein Schnittpunkt im WRK-Satz | + | m | | |
| G462 | 3 | Randsatz mit Gerade verlängern, wenn kein Schnittpunkt im WRK-Satz | + | m | | |

## 4.3.49 G-Gruppe 49: Punkt-zu-Punkt-Bewegung

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| CP | 1 | Bahnbewegung | + | m | x | |
| PTP | 2 | Punkt-zu-Punkt-Bewegung (Synchronachsbewegung) | + | m | | |
| PTPG0 | 3 | Punkt-zu-Punkt-Bewegung nur bei G0, sonst Bahnbewegung CP | + | m | | |
| PTPWOC | 4 | Punkt-zu-Punkt-Bewegung ohne Ausgleichsbewegungen, die durch Orientierungsänderungen verursacht sind | + | m | | |

## 4.3.50 G-Gruppe 50: Orientierungsprogrammierung

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| ORIEULER | 1 | Orientierungswinkel über Euler-Winkel | + | m | x | |
| ORIRPY | 2 | Orientierungswinkel über RPY-Winkel (Drehreihenfolge XYZ) | + | m | | |
| ORIVIRT1 | 3 | Orientierungswinkel über virtuelle Orientierungsachsen (Definition 1) | + | m | | |
| ORIVIRT2 | 4 | Orientierungswinkel über virtuelle Orientierungsachsen (Definition 2) | + | m | | |

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| ORIAXPOS | 5 | Orientierungswinkel über virtuelle Orientierungsachsen mit Rundachspositionen | + | m | | |
| ORIRPY2 | 6 | Orientierungswinkel über RPY-Winkel (Drehreihenfolge ZYX) | + | m | | |

## 4.3.51 G-Gruppe 51: Interpolationsart Orientierungsprogrammierung

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| ORIVECT | 1 | Großkreisinterpolation (identisch mit ORIPLANE) | + | m | x | |
| ORIAXES | 2 | Lineare Interpolation der Maschinenachsen oder Orientierungsachsen | + | m | | |
| ORIPATH | 3 | Werkzeugorientierungspfad bezogen auf die Bahn | + | m | | |
| ORIPLANE | 4 | Interpolation in der Ebene (identisch mit ORIVECT) | + | m | | |
| ORICONCW | 5 | Interpolation auf einer Kegelmantelfläche im Uhrzeigersinn | + | m | | |
| ORI-CONCCW | 6 | Interpolation auf einer Kegelmantelfläche gegen Uhrzeigersinn | + | m | | |
| ORICONIO | 7 | Interpolation auf einer Kegelmantelfläche mit Angabe einer Zwischenorientierung | + | m | | |
| ORICONTO | 8 | Interpolation auf einer Kegelmantelfläche mit tangentialem Übergang | + | m | | |
| ORICURVE | 9 | Interpolation mit zusätzlicher Raumkurve für die Orientierung | + | m | | |
| ORIPATHS | 10 | Werkzeugorientierung bezogen auf die Bahn, Knick im Orientierungsverlauf wird geglättet | + | m | | |

## 4.3.52 G-Gruppe 52: Werkstückbezogene Frame-Drehung

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| PAROTOF | 1 | Werkstückbezogene Frame-Drehung AUS | + | m | x | |
| PAROT | 2 | Werkstückbezogene Frame-Drehung EIN<br>Das Werkstückkoordinatensystem wird am Werkstück ausgerichtet. | + | m | | |

## 4.3.53 G-Gruppe 53: Werkzeugbezogene Frame-Drehung

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| TOROTOF | 1 | Werkzeugbezogene Frame-Drehung AUS | + | m | x | |
| TOROT | 2 | Z-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten | + | m | | |
| TOROTZ | 3 | wie TOROT | + | m | | |
| TOROTY | 4 | Y-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten | + | m | | |
| TOROTX | 5 | X-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten | + | m | | |
| TOFRAME | 6 | Z-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten | + | m | | |
| TOFRAMEZ | 7 | wie TOFRAME | + | m | | |
| TOFRAMEY | 8 | Y-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten | + | m | | |
| TOFRAMEX | 9 | X-Achse des WKS durch Frame-Drehung parallel zur Werkzeugorientierung ausrichten | + | m | | |

## 4.3.54 G-Gruppe 54: Vektordrehung bei Polynomprogrammierung

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| ORIROTA | 1 | Vektordrehung absolut | + | m | x | |
| ORIROTR | 2 | Vektordrehung relativ | + | m | | |
| ORIROTT | 3 | Vektordrehung tangential | + | m | | |
| ORIROTC | 4 | Tangentialer Drehvektor zur Bahntangente | + | m | | |

## 4.3.55 G-Gruppe 55: Eilgangbewegung mit/ohne Linear-Interpolation

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| RTLION | 1 | Eilgangbewegung mit Linear-Interpolation EIN | + | m | x | |
| RTLIOF | 2 | Eilgangbewegung mit Linear-Interpolation AUS<br>Die Eilgangbewegung wird mit Einzelachsinterpolation durchgeführt. | + | m | | |

## 4.3.56 G-Gruppe 56: Einrechnung des Werkzeugverschleißes

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| TOWSTD | 1 | Grundstellungswert für Korrekturen in der Werkzeuglänge | + | m | x | |
| TOWMCS | 2 | Verschleißwerte im Maschinenkoordinatensystem (MKS) | + | m | | |
| TOWWCS | 3 | Verschleißwerte im Werkstückkoordinatensystem (WKS) | + | m | | |
| TOWBCS | 4 | Verschleißwerte im Basiskoordinatensystem (BKS) | + | m | | |
| TOWTCS | 5 | Verschleißwerte im Werkzeugkoordinatensystem (Werkzeugträgerbezugspunkt T an der Werkzeughalteraufnahme) | + | m | | |
| TOWKCS | 6 | Verschleißwerte im Koordinatensystem des Werkzeugkopfes bei kinetischer Transformation (unterscheidet sich vom MKS durch Werzeugdrehung) | + | m | | |

## 4.3.57 G-Gruppe 57: Eckenverzögerung

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| FENDNORM | 1 | Eckenverzögerung AUS | + | m | x | |
| G62 | 2 | Eckenverzögerung an Innenecken bei aktiver Werkzeugradiuskorrektur (G41/G42) | + | m | | |
| G621 | 3 | Eckenverzögerung an allen Ecken | + | m | | |

## 4.3.58 G-Gruppe 59: Dynamikmodus für Bahninterpolation

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| DYNNORM | 1 | Normale Dynamik | + | m | x | |
| DYNPOS | 2 | Positionierbetrieb, Gewindebohren | + | m | | |
| DYNROUGH | 3 | Schruppen | + | m | | |
| DYNSEMIFIN | 4 | Vorschlichten | + | m | | |
| DYNFINISH | 5 | Schlichten | + | m | | |
| DYNPREC | 6 | Feinschlichten | + | m | | |

## 4.3.59 G-Gruppe 60: Arbeitsfeldbegrenzung

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| WALCS0 | 1 | WKS-Arbeitsfeldbegrenzung AUS | + | m | x | |
| WALCS1 | 2 | WKS-Arbeitsfeldbegrenzungsgruppe 1 aktiv | + | m | | |
| WALCS2 | 3 | WKS-Arbeitsfeldbegrenzungsgruppe 2 aktiv | + | m | | |
| WALCS3 | 4 | WKS-Arbeitsfeldbegrenzungsgruppe 3 aktiv | + | m | | |
| WALCS4 | 5 | WKS-Arbeitsfeldbegrenzungsgruppe 4 aktiv | + | m | | |
| WALCS5 | 6 | WKS-Arbeitsfeldbegrenzungsgruppe 5 aktiv | + | m | | |
| WALCS6 | 7 | WKS-Arbeitsfeldbegrenzungsgruppe 6 aktiv | + | m | | |
| WALCS7 | 8 | WKS-Arbeitsfeldbegrenzungsgruppe 7 aktiv | + | m | | |
| WALCS8 | 9 | WKS-Arbeitsfeldbegrenzungsgruppe 8 aktiv | + | m | | |
| WALCS9 | 10 | WKS-Arbeitsfeldbegrenzungsgruppe 9 aktiv | + | m | | |
| WALCS10 | 11 | WKS-Arbeitsfeldbegrenzungsgruppe 10 aktiv | + | m | | |

## 4.3.60 G-Gruppe 61: Glättung Werkzeugorientierung

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| ORISOF | 1 | Glättung Werkzeugorientierung AUS | + | m | x | |
| ORISON | 2 | Glättung Werkzeugorientierung EIN | + | m | | |

## 4.3.61 G-Gruppe 62: Wiederanfahrmodus für REPOS (satzweise wirksam)

| G-Befehl | Nr. [1] | Bedeutung | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| RMBBL | 1 | Wiederanfahren an Satzanfangspunkt | - | s | | |
| RMIBL | 2 | Wiederanfahren an Unterbrechungspunkt | - | s | x | |
| RMEBL | 3 | Wiederanfahren an Satzendpunkt | - | s | | |
| RMNBL | 4 | Wiederanfahren an nächstliegenden Bahnpunkt | - | s | | |

## 4.3.62 G-Gruppe 64: Schleifframes

| G-Befehl | Nr. [1] | Bedeutung<br>Aktiver Schleifframe im Kanal $P_GFRAME = | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| GFRAME[ 0 ] | 1 | Schleifframe der Datenhaltung $P_GFR[ 0 ] (Nullframe) | + | m | x | |
| GFRAME[ 1 ] | 2 | Schleifframe der Datenhaltung $P_GFR[ 1 ] | + | m | | |

| G-Befehl | Nr. [1] | Bedeutung<br>Aktiver Schleifframe im Kanal $P_GFRAME = | MD20150 [2] | W [3] | STD [4] | |
|---|---|---|---|---|---|---|
| | | | | | SAG | MH |
| GFRAME[ 2 ] | 3 | Schleifframe der Datenhaltung $P_GFR[ 2 ] | + | m | | |
| ... | ... | | + | m | | |
| GFRA-<br>ME[ 100 ] | 101 | Schleifframe der Datenhaltung $P_GFR[ 100 ] | + | m | | |

**Legende**

1) Interne Nummer (z. B. für PLC-Schnittstelle)

2) Projektierbarkeit des G-Befehls als Löschstellung der G-Gruppe bei Hochlauf, Reset bzw. Teileprogrammende (mit MD20150 $MC_GCODE_RESET_VALUES):

   \+ projektierbar

   \- nicht projektierbar

3) Wirksamkeit des G-Befehls:

   m modal (satzübergreifend)

   s satzweise

4) Löschstellung, siehe folgende Maschinendaten:

- MD20149 $MC_GCODE_RESET_S_VALUES (Löschstellung der G-Gruppen (fix) )
- MD20150 $MC_GCODE_RESET_VALUES (Löschstellung der G-Gruppen)
- MD20151 $MC_GCODE_RESET_S_MODE (Resetverhalten der G-Gruppen (fix) )
- MD20152 $MC_GCODE_RESET_MODE (Resetverhalten der G-Gruppen)
- MD20154 $MC_EXTERN_GCODE_RESET_VALUES (Löschstellung der G-Gruppen im ISO-Mode)
- MD20156 $MC_EXTERN_GCODE_RESET_MODE (Resetverhalten der externen G-Gruppen)

SAG Standardeinstellung **S**iemens **AG**

MH Standardeinstellung **M**aschinen**h**ersteller (siehe Angaben des Maschinen-Herstellers)

# 4.4 Vordefinierte Prozeduren

Durch den Aufruf einer vordefinierten Prozedur wird die Ausführung einer vordefinierten NC-Funktion angestoßen. Eine vordefinierte Prozedur liefert im Unterschied zur vordefinierten Funktion **keinen** Rückgabewert zurück.

| **Koordinatensystem** | | | | | |
|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | **Erläuterung** |
| | | | | | |
| | **1.** | **2.** | **3. - 15.** | **4. - 16.** | |
| PRESETON | AXIS *): Achsbezeichner Maschinenachse | REAL: Preset-Verschiebung G700/G710 Kontext | wie 1 ... | wie 2 ... | Istwertsetzen für die programmierten Achsen mit Verlust des Referenzierstatus |
| PRESETONS | AXIS *): Achsbezeichner Maschinenachse | REAL: Preset-Verschiebung G700/G710 Kontext | wie 1 ... | wie 2 ... | Istwertsetzen für die programmierten Achsen ohne Verlust des Referenzierstatus |
| | | | | | |
| DRFOF | | | | | DRF-Verschiebung löschen für alle dem Kanal zugeordneten Achsen |

*) Anstelle der Maschinenachsbezeichner können generell auch die Geometrie- oder Zusatzachsbezeichner stehen, sofern eine eindeutige Abbildung möglich ist.

| **Achsverbände** | | | | | |
|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | **Erläuterung** |
| | | | | | |
| GEOAX | **1.** | **2.** | **3. / 5.** | **4. / 6.** | Anwahl eines parallelen Koordinatensystems |
| | INT: Geometrieachsnummer 1 - 3 | AXIS: Kanalachsbezeichner | wie 1 | wie 2 | |
| | | | | | |
| FGROUP | **1. – 8.** | | | | Variabler F-Wert-Bezug: Festlegung der Achsen, auf die sich der Bahnvorschub bezieht<br>Maximale Achsanzahl: 8<br>Mit FGROUP ( ) ohne Angabe von Parametern wird die Standardeinstellung für den F-Wert-Bezug aktiviert. |
| | AXIS: Kanalachsbezeichner | | | | |
| | | | | | |
| SPLINEPATH | **1.** | **2. - 9.** | | | Festlegung des Spline-Verbandes<br>Maximale Achsanzahl: 8 |
| | INT: Spline-Verband (muss 1 sein) | AXIS: Geometrie- oder Zusatzbezeichner | | | |

| Achsverbände | | | | |
|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | **Erläuterung** |
| | | | | |
| POLYPATH | **1.** | **2.** | | Einschalten der Polynominterpolation für selektive Achsgruppen |
| | STRING | STRING | | |

| Mitschleppen | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | | | **Erläuterung** |
| | **1.** | **2.** | **3.** | **4.** | **5.** | **6.** | |
| TANG | AXIS: Achsname Folgeachse | AXIS : Leitachse 1 | AXIS: Leitachse 2 | REAL: Koppelfaktor | CHAR: Option: "B": Nachführung im BKS "W": Nachführung im WKS | CHAR Optimierung: "S": Standard "P": autom. mit Überschleifweg, Winkeltoleranz | Tangentialsteuerung: Kopplung definieren<br>Aus den beiden angegebenen Leitachsen wird die Tangente für die Nachführung bestimmt. Der Koppelfaktor gibt den Zusammenhang zwischen einer Änderung des Winkels der Tangente und der nachgeführten Achse an. Er ist in der Regel 1. |
| TANGON | AXIS: Achsname Folgeachse | REAL: Offset-Winkel | REAL: Überschleifweg | REAL: Winkeltoleranz | | | Tangentialsteuerung: Kopplung einschalten |
| TANGOF | AXIS: Achsname Folgeachse | | | | | | Tangentialsteuerung: Kopplung ausschalten |
| TLIFT | AXIS: Nachgeführte Achse | | | | | | Tangentialsteuerung: Zwischensatzerzeugung einschalten |
| TRAILON | AXIS: Folgeachse | AXIS: Leitachse | REAL: Koppelfaktor | | | | Achssynchrones Mitschleppen einschalten |
| TRAILOF | AXIS: Folgeachse | AXIS: Leitachse | | | | | Achssynchrones Mitschleppen ausschalten |
| TANGDEL | AXIS: Folgeachse | | | | | | Tangentialsteuerung: Kopplung löschen |

| **Kurventabellen** | | | | | | |
|---|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | | **Erläuterung** |
| | **1.** | **2.** | **3.** | **4.** | **5.** | |
| CTABDEF | AXIS: Folgeachse | AXIS: Leitachse | INT: Tabellen-nummer | INT: Verhalten an Rändern des Def.-Bereichs | STRING: Angabe des Speicherorts | Tabellendefinition EIN<br>Die nachfolgenden Bewegungssätze bestimmen die Kurventabelle. |
| CTABEND | AXIS: Folgeachse | AXIS: Leitachse | INT: Tabellen-nummer | INT: Verhalten an Rändern des Def.-Bereichs | | Tabellendefinition AUS |
| CTABDEL | INT: Tabellen-nummer n | INT: Tabellen-nummer m | STRING: Angabe des Speicherorts | | | Kurventabelle löschen |
| CTABLOCK | INT: Tabellen-nummer n | | | | | Sperrt die Kurventabelle mit der Nummer n, d. h. diese Tabelle kann nicht gelöscht/überschrieben werden. |
| CTABUNLOCK | INT: Tabellen-nummer n | | | | | Gibt die mit CTABLOCK geschützte Tabelle mit der Nummer n wieder frei |
| LEADON | AXIS: Folgeachse | AXIS: Leitachse | INT: Tabellen-nummer | | | Leitwertkopplung EIN |
| LEADOF | AXIS: Folgeachse | AXIS: Leitachse | | | | Leitwertkopplung AUS |

| **Beschleunigungsprofil axial** | | |
|---|---|---|
| **Bezeichner** | **Parameter** | **Erläuterung** |
| | **1. – 8.** | |
| BRISKA | AXIS | Sprungförmige Achsbeschleunigung für die programmierten Achsen einschalten |
| SOFTA | AXIS | Ruckbegrenzte Achsbeschleunigung für die programmierten Achsen einschalten |
| DRIVEA | AXIS | Geknickte Beschleunigungskennlinie für die programmierten Achsen einschalten |
| JERKA | AXIS | Das über Maschinendatum $MA_AX_JERK_ENABLE eingestellte Beschleunigungsverhalten wirkt für die programmierten Achsen. |

| **Umdrehungsvorschub** | | | |
|---|---|---|---|
| **Bezeichner** | **Parameter** | | **Erläuterung** |
| | | | |
| FPRAON | **1.** | **2.** | Umdrehungsvorschub axial EIN |
| | AXIS:<br>Achse, für die Umdrehungsvorschub eingeschaltet wird | AXIS:<br>Achse/Spindel, von der Umdrehungsvorschub abgeleitet wird.<br>Ist keine Achse programmiert, so wird der Umdrehungsvorschub von der Masterspindel abgeleitet. | |
| | | | |
| FPRAOF | **1. - n.** | | Umdrehungsvorschub axial AUS<br>Der Umdrehungsvorschub kann für mehrere Achsen gleichzeitig ausgeschaltet werden. Es können so viele Achsen programmiert werden, wie maximal pro Satz zulässig sind. |
| | AXIS:<br>Achsen, für die Umdrehungsvorschub ausgeschaltet wird | | |
| | | | |
| FPR | **1.** | | Auswahl einer Rundachse/Spindel, von der der Umdrehungsvorschub der Bahn bei G95 abgeleitet wird.<br>Die mit FPR getroffene Einstellung gilt modal. |
| | AXIS:<br>Achse/Spindel, von der Umdrehungsvorschub abgeleitet wird.<br>Ist keine Achse programmiert, so wird der Umdrehungsvorschub von der Masterspindel abgeleitet. | | |

| **Transformationen** | | | | |
|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | **Erläuterung** |
| | **1.** | **2.** | **3.** | |
| TRACYL | REAL:<br>Arbeitsdurchmesser | INT:<br>Nummer der Transformation | | Zylinder: Mantelflächen-Transformation<br>Pro Kanal können mehrere Transformationen eingestellt werden. Die Transformationsnummer gibt an, welche Transformation aktiviert werden soll. Entfällt der 2. Parameter, so wird der über MD eingestellte Transformationsverband aktiviert. |
| TRANSMIT | INT:<br>Nummer der Transformation | | | Transmit: Polar-Transformation<br>Pro Kanal können mehrere Transformationen eingestellt werden. Die Transformationsnummer gibt an, welche Transformation aktiviert werden soll. Entfällt der Parameter, so wird der über MD eingestellte Transformationsverband aktiviert. |

| **Transformationen** | | | | |
|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | **Erläuterung** |
| | **1.** | **2.** | **3.** | |
| TRAANG | REAL:<br>Winkel | INT:<br>Nummer der Transformation | | Transformation schräge Achse<br>Pro Kanal können mehrere Transformationen eingestellt werden. Die Transformationsnummer gibt an, welche Transformation aktiviert werden soll. Entfällt der 2. Parameter, so wird der über MD eingestellte Transformationsverband aktiviert.<br>Wird der Winkel nicht programmiert<br>(TRAANG ( ,2) oder TRAANG)<br>so ist der letzte Winkel modal wirksam. |
| TRAORI | INT:<br>Nummer der Transformation | | | 4-, 5-Achs-Transformation<br>Pro Kanal können mehrere Transformationen eingestellt werden. Die Transformationsnummer gibt an, welche Transformation aktiviert werden soll. |
| TRACON | INT:<br>Nummer der Transformation | REAL: weitere Parameter MD-abh. | | Kaskadierte Transformation<br>Die Bedeutung der Parameter hängt von der Art der Kaskadierung ab. |
| TRAFOOF | | | | Transformation ausschalten |
| TRAFOON | STRING:<br>Name des Transformationsdatensatzes | REAL:<br>Bezugs- bzw. Arbeitsdurchmesser (nur TRACYL) | BOOL:<br>mit/ohne Nutwandkorrektur (nur TRACYL) | Mit kinematischen Ketten definierte Transformation einschalten |

| **Spindel** | | | |
|---|---|---|---|
| **Bezeichner** | **Parameter** | | **Erläuterung** |
| | **1** | **2. - n.** | |
| SPCON | INT:<br>Spindelnummer | INT:<br>Spindelnummer | Umschalten in den lagegeregelten Spindelbetrieb |
| SPCOF | INT:<br>Spindelnummer | INT:<br>Spindelnummer | Umschalten in den drehzahlgeregelten Spindelbetrieb |
| SETMS | INT:<br>Spindelnummer | | Deklaration der Spindel als Master-Spindel für den aktuellen Kanal<br>Mit SETMS( ) ohne Angabe von Parametern wird die über Maschinendaten getroffene Voreinstellung wirksam. |

| **Schleifen** | | |
|---|---|---|
| **Bezeichner** | **Parameter** | **Erläuterung** |
| | **1.** | |
| GWPSON | INT:<br>Spindelnummer | Konstante Scheibenumfangsgeschwindigkeit EIN<br>Wird die Spindelnummer nicht programmiert, so wird für die Spindel des aktiven Werkzeugs Scheibenumfangsgeschwindigkeit angewählt. |
| GWPSOF | INT:<br>Spindelnummer | Konstante Scheibenumfangsgeschwindigkeit AUS<br>Wird die Spindelnummer nicht programmiert, so wird für die Spindel des aktiven Werkzeugs Scheibenumfangsgeschwindigkeit abgewählt. |
| TMON | INT:<br>T-Nummer | Schleifspezifische Werkzeugüberwachung EIN<br>Wird keine T-Nummer programmiert, so wird die Überwachung für das aktive Werkzeug eingeschaltet. |
| TMOF | INT:<br>T-Nummer | Werkzeugüberwachung AUS<br>Wird keine T-Nummer programmiert, so wird die Überwachung für das aktive Werkzeug ausgeschaltet. |

| **Abspanen** | | | | | |
|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | **Erläuterung** |
| | **1.** | **2.** | **3.** | **4.** | |
| CONTPRON | REAL [ ,11]:<br>Konturtabelle | CHAR: Bearbeitungsart | INT:<br>Anzahl der Hinterschnitte | INT:<br>Status der Berechnung | Referenzaufbereitung einschalten<br>Die im Folgenden aufgerufenen Konturprogramme bzw. NC-Sätze werden in einzelne Bewegungen aufgeteilt und in der Konturtabelle abgelegt.<br>Die Anzahl der Hinterschnitte wird zurückgeliefert. |
| CONTDCON | REAL [ , 6]:<br>Konturtabelle | INT:<br>Bearbeitungsrichtung | | | Konturdecodierung<br>Die Sätze einer Kontur werden mit einer Tabellenzeile je Satz speichergünstig codiert in einer benannten Tabelle gespeichert. |
| EXECUTE | INT: Fehlerstatus | | | | Programmausführung einschalten<br>Damit wird aus dem Referenzaufbereitungsmodus oder nach Aufbau eines Schutzbereichs zur normalen Programmbearbeitung zurückgeschaltet. |

| **Tabelle abarbeiten** | | |
|---|---|---|
| **Bezeichner** | **Parameter** | **Erläuterung** |
| | **1.** | |
| EXECTAB | REAL [ 11]:<br>Element aus Bewegungstabelle | Ein Element aus einer Bewegungstabelle abarbeiten |

| **Schutzbereiche** | | | | | | |
|---|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | | **Erläuterung** |
| | **1.** | **2.** | **3.** | **4.** | **5.** | |
| CPROTDEF | INT:<br>Nummer des Schutzbereichs | BOOL:<br>TRUE:<br>Werkzeugorientierter Schutzbereich | INT:<br>0:<br>4. und 5. Parameter werden nicht ausgewertet<br>1:<br>4. Parameter wird ausgewertet<br>2:<br>5. Parameter wird ausgewertet<br>3:<br>4. und 5. Parameter werden ausgewertet | REAL: Begrenzung in Plus-Richtung | REAL: Begrenzung in Minus-Richtung | Definition eines Kanalspezifischen Schutzbereichs |
| NPROTDEF | INT:<br>Nummer des Schutzbereichs | BOOL:<br>TRUE:<br>Werkzeugorientierter Schutzbereich | INT:<br>0:<br>4. und 5. Parameter werden nicht ausgewertet<br>1:<br>4. Parameter wird ausgewertet<br>2:<br>5. Parameter wird ausgewertet<br>3:<br>4. und 5. Parameter werden ausgewertet | REAL: Begrenzung in Plus-Richtung | REAL: Begrenzung in Minus-Richtung | Definition eines Maschinen-spezifischen Schutzbereichs |

| Schutzbereiche | | | | | | |
|---|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | | **Erläuterung** |
| | **1.** | **2.** | **3.** | **4.** | **5.** | |
| CPROT | INT: Nummer des Schutzbereichs | INT: Option<br>0: Schutzbereich aus<br>1: Schutzbereich voraktivieren<br>2: Schutzbereich ein<br>3: Schutzbereich voraktivieren mit bedingtem Stopp, nur bei aktiven Schutz-bereichen | REAL: Verschiebung des Schutzbereichs in der ersten Geometrieachse | REAL: Verschiebung des Schutzbereichs in der zweiten Geometrieachse | REAL: Verschiebung des Schutzbereichs in der dritten Geometrieachse | Kanal-spezifischen Schutzbereich EIN/AUS |
| NPROT | INT: Nummer des Schutzbereichs | INT: Option<br>0: Schutzbereich aus<br>1: Schutzbereich voraktivieren<br>2: Schutzbereich ein<br>3: Schutzbereich voraktivieren mit bedingtem Stopp, nur bei aktiven Schutz-bereichen | REAL: Verschiebung des Schutzbereichs in der ersten Geometrieachse | REAL: Verschiebung des Schutzbereichs in der zweiten Geometrieachse | REAL: Verschiebung des Schutzbereichs in der dritten Geometrieachse | Maschinen-spezifischen Schutzbereich EIN/AUS |

| Vorlauf / Einzelsatz | | |
|---|---|---|
| **Bezeichner** | **Parameter** | **Erläuterung** |
| STOPRE | | Vorlaufstopp, bis alle vorbereiteten Sätze vom Hauptlauf abgearbeitet sind |
| SBLOF | | Einzelsatzbearbeitung unterdrücken |
| SBLON | | Unterdrückung der Einzelsatzbearbeitung aufheben |

| **Interrupts** | | |
|---|---|---|
| **Bezeichner** | **Parameter** | **Erläuterung** |
| | **1.** | |
| DISABLE | INT:<br>Nummer des Interrupt-Eingangs | Die Interruptroutine, die dem angegebenen Hardware-Eingang zugeordnet ist, wird inaktiv geschaltet. Auch Schnellabheben wird nicht ausgeführt. Die mit SETINT getroffene Zuordnung zwischen Hardware-Eingang und Interruptroutine bleibt erhalten und kann mit ENABLE wieder aktiviert werden. |
| ENABLE | INT:<br>Nummer des Interrupt-Eingangs | Reaktivierung der mit DISABLE inaktiv geschalteten Interruptroutinen-Zuordnung. |
| CLRINT | INT:<br>Nummer des Interrupt-Eingangs | Zuordnung von Interruptroutinen und Attributen zu einem Interrupt-Eingang löschen. Die Interruptroutine ist somit abgewählt. Beim Eintreffen des Interrupts erfolgt keine Reaktion. |

| **Synchronaktionen** | | |
|---|---|---|
| **Bezeichner** | **Parameter** | **Erläuterung** |
| | **1. – n.** | |
| CANCEL | INT:<br>Nummer der Synchronaktion | Abbrechen der modalen Synchronaktion mit der angegebenen ID. Es können auch mehrere IDs – durch Kommata getrennt -- angegeben werden. |
| | | |
| CANCELSUB | | Aktuelle Unterprogrammebene abbrechen |

| **Funktionsdefinition** | | | | | |
|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | **Erläuterung** |
| | **1.** | **2.** | **3.** | **4.-7.** | |
| FCTDEF | INT:<br>Funktionsnummer | REAL:<br>Unterer Grenzwert | REAL:<br>Oberer Grenzwert | REAL:<br>Koeffizienten a0-a3 | Polynomfunktion definieren<br>Diese wird in SYNFCT oder PUTFTOCF ausgewertet. |

| **Kommunikation** | | | |
|---|---|---|---|
| **Bezeichner** | **Parameter** | | **Erläuterung** |
| | **1.** | **2.** | |
| MMC | STRING:<br>Kommando | CHAR:<br>Quittungsmodus*)<br>"N": ohne Quittung<br>"S": synchrone Quittung<br>"A": asynchrone Quittung | Kommando an HMI-Kommando-Interpreter für die Projektierung von Fenstern über NC-Programm |

*) Kommandos werden auf Anforderung von der ausführenden Komponente (Kanal, NC …) quittiert.

| **Programmkoordinierung** | | | | | |
|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | **Erläuterung** |
| | | | | | |
| INIT | **1.** | **2.** | **3.** | | Anwahl eines NC-Programms zur Abarbeitung in einem Kanal |
| | INT: Kanalnummer<br>**oder**<br>Kanalname aus MD20000*) | STRING: Pfadangabe | CHAR: Quittungsmodus**) | | |
| | | | | | |
| | **1. - n.** | | | | |
| START | INT:<br>Kanalnummer<br>**oder**<br>Kanalname aus MD20000 *) | | | | Starten der angewählten Programme in mehreren Kanälen gleichzeitig aus dem laufenden Programm<br>Dieses Kommando hat für den eigenen Kanal keine Wirkung. |
| WAITE | INT:<br>Kanalnummer<br>**oder**<br>Kanalname aus MD20000 *) | | | | Warten auf das Programmende in einem oder mehreren anderen Kanälen |
| | | | | | |
| | **1.** | **2. - n.** | | | |
| WAITM | INT:<br>Markennummer | INT:<br>Kanalnummer<br>**oder**<br>Kanalname aus MD20000 *) | | | Warten auf das Erreichen einer Marke in den angegebenen Kanälen<br>Der vorhergehende Satz wird mit Genauhalt beendet. |
| WAITMC | INT:<br>Markennummer | INT:<br>Kanalnummer<br>**oder**<br>Kanalname aus MD20000 *) | | | Warten auf das Erreichen einer Marke in den angegebenen Kanälen<br>Genauhalt wird nur eingeleitet, wenn die anderen Kanäle die Marke noch nicht erreicht haben. |
| | | | | | |
| | **1. - n.** | | | | |
| SETM | INT:<br>Markennummer | | | | Setzen einer oder mehrerer Marken für die Kanalkoordinierung<br>Die Bearbeitung im eigenen Kanal wird davon nicht beeinflusst. |
| CLEARM | INT:<br>Markennummer | | | | Löschen einer oder mehrerer Marken für die Kanalkoordinierung<br>Die Bearbeitung im eigenen Kanal wird davon nicht beeinflusst. |
| | | | | | |
| | **1. - n.** | | | | |

| **Programmkoordinierung** | | | | | |
|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | **Erläuterung** |
| WAITP | AXIS:<br>Achsbezeichner | | | | Warten, bis die angegebenen Positionierachsen, die zuvor mit POSA programmiert wurden, ihren programmierten Endpunkt erreichen |
| WAITS | INT:<br>Spindelnummer | | | | Warten, bis die angegebenen Spindeln, die zuvor mit SPOSA programmiert wurden, ihren programmierten Endpunkt erreichen |
| | | | | | |
| RET | **1.** | **2.** | **3.** | **4.** | Unterprogrammende ohne Funktions-Ausgabe an die PLC<br>Bei Angabe des 1. Parameters (Sprungziel) erfolgt der Rücksprung zuerst auf den Satz nach dem Aufrufsatz. Anschließend wird das Ziel abhängig von der Programmierung (RET oder RETB) nach folgender Strategie gesucht:<br>• RET:<br>Suche in Richtung Programmende. Ist die Suche nicht erfolgreich, wird als nächstes in Richtung Programmanfang gesucht.<br>• RETB::<br>Suche in Richtung Programmanfang. Ist die Suche nicht erfolgreich, wird als nächstes in Richtung Programmende gesucht. |
| | INT (oder STRING):<br>Sprungziel (Satz-Nr. / Marke) für Rücksprung | INT:<br>0:<br>Rücksprung auf Sprungziel aus 1. Par.<br>> 0:<br>Rücksprung auf den Folgesatz | INT:<br>Anzahl der zu überspringenden Unterprogrammebenen | BOOL:<br>Rücksprung auf ersten Satz im Hauptprogramm | |
| RETB | INT (oder STRING):<br>Sprungziel (Satz-Nr. / Marke) für Rücksprung | INT:<br>0:<br>Rücksprung auf Sprungziel aus 1. Par.<br>> 0:<br>Rücksprung auf den Folgesatz | INT:<br>Anzahl der zu überspringenden Unterprogrammebenen | BOOL:<br>Rücksprung auf ersten Satz im Hauptprogramm | |
| | | | | | |
| | **1. - n.** | | | | |
| GET | AXIS:<br>Achsbezeichner ***) | | | | Maschinenachse(n) belegen<br>Die angegebenen Achsen müssen mit RELEASE im anderen Kanal freigegeben werden. |
| GETD | AXIS:<br>Achsbezeichner ***) | | | | Maschinenachse(n) direkt belegen<br>Die angegebenen Achsen müssen **nicht** mit RELEASE freigegeben sein. |
| RELEASE | AXIS:<br>Achsbezeichner ***) | | | | Maschinenachse(n) freigeben |
| | | | | | |
| | **1.** | **2.** | **3.** | **4.** | |

| **Programmkoordinierung** | | | | | |
|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | **Erläuterung** |
| PUTFTOC | REAL: Korrekturwert | INT: Parameternummer | INT: Kanalnummer<br>**oder**<br>Kanalname aus MD20000*) | INT: Spindelnummer | Ändern der Werkzeugfeinkorrektur |
| PUTFTOCF | INT: Nr. der Funktion | VAR REAL: Bezugswert | INT: Parameter-nummer | INT: Kanalnummer<br>**oder**<br>Kanalname aus MD20000*) | Ändern der Werkzeugfeinkorrektur in Abhängigkeit einer mit FCTDEF festgelegten Funktion<br>(Polynom max. 3. Grades)<br>Bei FCTDEF ist die hier verwendete Nr. anzugeben. |
| | | | | | |
| AXTOCHAN | **1.** | **2.** | **3. - n.** | **4. - m.** | Achsen an andere Kanäle übergeben |
| | AXIS: Achsbezeichner | INT: Kanalnummer<br>**oder**<br>Kanalname aus MD20000*) | wie 1 ... | wie 2 ... | |

*) Anstelle von Kanalnummern können auch die über MD20000 $MC_CHAN_NAME definierten Kanalnamen programmiert werden.

**) Kommandos werden auf Anforderung von der ausführenden Komponente (Kanal, NC …) quittiert.

***) Anstelle der Achse kann mittels der Funktion SPI jeweils auch eine Spindel programmiert werden: z. B. GET(SPI(1))

| **Datenzugriffe** | | |
|---|---|---|
| **Bezeichner** | **Parameter** | **Erläuterung** |
| | | |
| CHANDATA | **1.**<br>INT: Kanal-nummer | Kanalnummer für Kanaldatenzugriffe einstellen (nur im Initialisierungsbaustein zulässig). Die folgenden Zugriffe beziehen sich auf den mit CHANDATA eingestellten Kanal. |
| | | |
| NEWCONF | | Geänderte Maschinendaten übernehmen |

| **Meldungen** | | | |
|---|---|---|---|
| **Bezeichner** | **Parameter** | | **Erläuterung** |
| | **1.** | **2.** | |
| MSG | STRING: Meldung | INT: Ausführung | Beliebige Zeichenkette als Meldung an die Bedienoberfläche ausgeben |
| WRTPR | STRING: Zeichenkette | INT: Ausführung | Zeichenkette in BTSS-Variable schreiben |

| **Dateizugriffe** | | | | | | |
|---|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | | **Erläuterung** |
| | | | | | | |
| READ | **1.** | **2.** | **3.** | **4.** | **5.** | Sätze aus Filesystem lesen |
| | VAR INT: Fehler | CHAR[160]: Dateiname | INT: Anfangszeile des zu lesenden Dateibereichs | INT: Anzahl der zu lesenden Zeilen | VAR CHAR[255]: Variablenfeld, in dem die gelesene Information abgelegt wird | |
| | | | | | | |
| WRITE | **1.** | **2.** | **3.** | **4.** | | Satz ins Filesystem (oder auf ein externes Gerät/Datei) schreiben |
| | VAR INT: Fehler | CHAR[160]: Dateiname | STRING: Gerät/Datei für externe Ausgabe | CHAR[200]: Satz | | |
| | | | | | | |
| DELETE | **1.** | **2.** | | | | Datei löschen |
| | VAR INT: Fehler | CHAR[160]: Dateiname | | | | |

| **Alarme** | | | |
|---|---|---|---|
| **Bezeichner** | **Parameter** | | **Erläuterung** |
| | **1.** | **2.** | |
| SETAL | INT: Alarmnummer (Zyklenalarme) | STRING: Zeichenkette | Alarm setzen<br>Zur Alarmnummer kann zusätzlich eine Zeichenkette mit bis zu 4 Parametern angegeben werden.<br>Es stehen folgende vordefinierte Parameter zur Verfügung:<br>%1 = Kanalnummer<br>%2 = Satznummer, Label<br>%3 = Textindex für Zyklenalarme<br>%4 = zusätzlichen Alarmparameter |

| **Werkzeugverwaltung** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | | | **Erläuterung** |
| | | | | | | | |
| | **1.** | **2.** | | | | | |
| DELDL | INT:<br>T-Nr. | INT:<br>D-Nr. | | | | | Alle Summenkorrekturen einer Schneide löschen (oder eines Werkzeugs, wenn D nicht angegeben wird) |
| DELT | STRING[32]:<br>Werkzeugbezeichner | INT:<br>Duplo-Nr. | | | | | Werkzeug löschen<br>Duplonummer kann entfallen. |
| DELTC | INT:<br>Datensatz-Nr. n | INT:<br>Datensatz-Nr. m | | | | | Werkzeugträger-Datensatznummer n bis m löschen |
| | | | | | | | |
| DZERO | | | | | | | D-Nummern aller Werkzeuge der dem Kanal zugeordneten TO-Einheit ungültig setzen |
| | | | | | | | |
| | **1.** | **2.** | **3.** | **4.** | **5.** | **6.** | |
| GETFREELOC | VAR INT:<br>Magazin-Nr. (Rückgabewert) | VAR INT:<br>Platz-Nr. (Rückgabewert) | INT:<br>T-Nr. | INT:<br>Referenz-Magazin-Nr. | CHAR:<br>Angabe abh. von 4. Parameter | INT:<br>Reservierungsmodus | Leerplatz für ein Werkzeug suchen |
| | | | | | | | |
| | **1.** | **2.** | | | | | |
| GETSELT | VAR INT:<br>T-Nr. (Rückgabewert) | INT:<br>Spindel-Nr. | | | | | Liefert die T-Nummer des für die Spindel vorgewählten Werkzeugs |
| GETEXET | VAR INT:<br>T-Nr. (Rückgabewert) | INT:<br>Spindel-Nr. | | | | | Liefert die T-Nummer des aus Sicht des NC-Programms aktiven Werkzeugs |
| GETTENV | STRING:<br>Name der Werkzeugumgebung | INT ARRAY[3]:<br>Rückgabewerte | | | | | Liest die in einer Werkzeugumgebung abgelegten T-, D- und DL-Nummern |
| | | | | | | | |
| | **1.** | **2.** | **3.** | **4.** | | | |
| POSM | INT:<br>Nr. des Platzes, auf den positioniert werden soll | INT:<br>Nr. des Magazins, das bewegt werden soll | INT:<br>Platz-Nr. des internen Magazins | INT: Magazin-Nr. des internen Magazins | | | Magazin positionieren |

| **Werkzeugverwaltung** | | | | | | |
|---|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | | **Erläuterung** |
| RESETMON | VAR INT: Status = Ergebnis der Operation (Rückgabewert) | INT: interne T-Nr. | INT: D-Nr. des Werkzeugs | INT: Optionaler bitcodierter Parameter | | Istwert des Werkzeugs auf den Sollwert setzen |
| | | | | | | |
| SETDNO | **1.** | **2.** | **3.** | | | Korrekturnummer (D) der Schneide des Werkzeugs (T) setzen |
| | INT: T-Nr. | INT: Schneiden-Nr. | INT: D-Nr. | | | |
| | | | | | | |
| SETMTH | **1.** | | | | | Werkzeughalter-Nr. setzen |
| | INT: Werkzeughalter-Nr. | | | | | |
| | | | | | | |
| SETPIECE | **1.** | **2.** | | | | Werkstückzähler der Spindel dekrementieren<br>Damit kann der Anwender die Stückzahl-Überwachungsdaten der an dem Bearbeitungsprozess beteiligten Werkzeuge aktualisieren. |
| | INT: Wert, um den dekrementiert wird | INT: Spindel-Nr. | | | | |
| | | | | | | |
| | **1.** | **2.** | **3.** | **4.** | | |
| SETTA | VAR INT: Status = Ergebnis der Operation (Rückgabewert) | INT: Magazin-Nr. | INT: Verschleißverbund-Nr. | INT: Werkzeuguntergruppe | | Werkzeug aus Verschleißverbund aktiv setzen |
| SETTIA | VAR INT: Status = Ergebnis der Operation (Rückgabewert) | INT: Magazin-Nr. | INT: Verschleißverbund-Nr. | INT: Werkzeuguntergruppe | | Werkzeug aus Verschleißverbund inaktiv setzen |
| | | | | | | |
| TCA | **1.** | **2.** | **3.** | | | WZ-Anwahl/-Wechsel unabhängig vom Status des Werkzeugs |
| | STRING[32]: Werkzeugbezeichner | INT: Duplo-Nr. | INT: Werkzeughalter-Nr. | | | |
| | | | | | | |

| **Werkzeugverwaltung** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | | | **Erläuterung** |
| TCI | **1.** | **2.** | | | | | Werkzeug aus Zwischenspeicher in das Magazin wechseln |
| | INT: Nr. des Zwischenspeichers | INT: Werkzeughalter-Nr. | | | | | |
| | | | | | | | |
| MVTOOL | **1.** | **2.** | **3.** | **4.** | **5.** | | Sprachbefehl zum Bewegen eines Werkzeugs |
| | INT: Status | INT: Magazin-Nr. | INT: Platz-Nr. | INT: Magazin-Nr. nach der Bewegung | INT: Zielplatz-Nr. nach der Bewegung | | |

| **Werkzeugorientierung** | | | | |
|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | **Erläuterung** |
| | **1.** | **2.** | **3.** | |
| ORIRESET | REAL: Grundstellung 1. Geoachse | REAL: Grundstellung 2. Geoachse | REAL: Grundstellung 3. Geoachse | Grundstellung der Werkzeugorientierung |

| **Synchronspindel** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | | | **Erläuterung** |
| | **1.** | **2.** | **3.** | **4.** | **5.** | **6.** | |
| COUPDEF | AXIS: Folgespindel | AXIS: Leitspindel | REAL: Zähler Übersetzungs-verhältnis | REAL: Nenner Übersetzungsverhältnis | STRING[8]: Satzwechselverhalten | STRING[2]: Koppelart | Synchronspindel-Verband definieren |
| COUPDEL | AXIS: Folgespindel | AXIS: Leitspindel | | | | | Synchronspindel-Verband löschen |
| COUPRES | AXIS: Folgespindel | AXIS: Leitspindel | | | | | Kopplungsparameter zurücksetzen auf projektierte MD- und SD-Werte |

| **Synchronspindel** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | | | **Erläuterung** |
| | **1.** | **2.** | **3.** | **4.** | **5.** | **6.** | |
| COUPON | AXIS: Folgespindel | AXIS: Leitspindel | REAL: Einschaltposition der Folgespindel | | | | Synchronspindel-Kopplung einschalten<br>Wird für die Folgespindel eine Einschaltposition (Winkelversatz zwischen FS und LS, der sich -- absolut oder inkrementell -- auf die Null-Grad-Position der LS in positiver Drehrichtung bezieht) angegeben, dann wird die Kopplung erst eingeschaltet, wenn die angegebene Position überfahren wurde. |
| COUPONC | AXIS: Folgespindel | AXIS: Leitspindel | | | | | Synchronspindel-Kopplung einschalten<br>Mit COUPONC wird beim Einschalten der Kopplung die aktuell wirksame Drehzahl der Folgespindel ( M3/M4 S...) übernommen. |
| COUPOF | AXIS: Folgespindel | AXIS: Leitspindel | REAL: Ausschaltposition der Folgespindel (absolut) | REAL: Ausschaltposition der Leitspindel (absolut) | | | Synchronspindel-Kopplung ausschalten<br>Werden Positionen angegeben, dann wird die Kopplung erst aufgelöst, wenn alle angegebenen Positionen überfahren wurden.<br>Die Folgespindel dreht mit der letzten Drehzahl vor Ausschalten der Kopplung weiter. |

| **Synchronspindel** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | | | **Erläuterung** |
| | **1.** | **2.** | **3.** | **4.** | **5.** | **6.** | |
| COUPOFS | AXIS: Folgespindel | AXIS: Leitspindel | REAL: Ausschaltposition der Folgespindel (absolut) | | | | Synchronspindel-Kopplung ausschalten mit Stopp der Folgespindel<br>Wird eine Position angegeben, dann wird die Kopplung erst aufgelöst, wenn die angegebene Position überfahren wurde |
| WAITC | AXIS: Folgespindel | STRING [8]: Satzwechselverhalten | AXIS: Folgespindel | STRING[8]: Satzwechselverhalten | | | Warten, bis das Kopplungssatzwechselkriterium für die Spindeln (max. 2) erfüllt ist<br>Wird das Satzwechselverhalten nicht angegeben, gilt das bei der Definition mit COUPDEF angegebene Satzwechselverhalten. |

| **Elektronisches Getriebe** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | | | **Erläuterung** |
| EGDEL | **1.** | | | | | | Kopplungsdefinition für die Folgeachse löschen |
| | AXIS: Folgeachse | | | | | | |
| EGDEF | **1.** | **2. / 4. / 6. / 8. / 10.** | **3. / 5. / 7. / 9. / 11.** | | | | Definition eines elektronischen Getriebes |
| | AXIS: Folgeachse | AXIS: Leitachse | INT: Kopplungsart | | | | |
| EGON | **1.** | **2.** | **3. / 6. / 9. / 12. / 15.** | **4. / 7. / 10. / 13. / 16.** | **5. / 8. / 11. / 14. / 17.** | | Elektronisches Getriebe EIN ohne Synchronisation |
| | AXIS: Folgeachse | STRING: Satzwechselverhalten | AXIS: Leitachse | REAL: Zähler des Koppelfaktors | REAL: Nenner des Koppelfaktors | | |

| **Elektronisches Getriebe** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | | | | | **Erläuterung** |
| EGONSYN | **1.** | **2.** | **3.** | **4. / 8. / 12. / 16. / 20.** | **5. / 9. / 13. / 17. / 21.** | **6. / 10. / 14. / 18. / 22.** | **7. / 11. / 15. / 19. / 23.** | | Elektronisches Getriebe EIN mit Synchronisation |
| | AXIS: Folgeachse | STRING: Satzwechselverhalten | REAL: Synchronposition der Folgeachse | AXIS: Leitachse | REAL: Synchronposition der Leitachse | REAL: Zähler des Koppelfaktors | REAL: Nenner des Koppelfaktors | | |
| | | | | | | | | | |
| EGONSYNE | **1.** | **2.** | **3.** | **4.** | **5. / 9. / 13. / 17. / 21.** | **6. / 10. / 14. / 18. / 22.** | **7. / 11. / 15. / 19. / 23.** | **8. / 12. / 16. / 20. / 24.** | Elektronisches Getriebe EIN mit Synchronisation und Vorgabe des Anfahrmodus |
| | AXIS: Folgeachse | STRING: Satzwechselverhalten | REAL: Synchronposition der Folgeachse | STRING: Anfahrmodus | AXIS: Leitachse | REAL: Synchronposition der Leitachse | REAL: Zähler des Koppelfaktors | REAL: Nenner des Koppelfaktors | |
| | | | | | | | | | |
| EGOFS | **1.** | **2. - n.** | | | | | | | Elektronisches Getriebe selektiv ausschalten |
| | AXIS: Folgeachse | AXIS: Leitachse | | | | | | | |
| | | | | | | | | | |
| EGOFC | **1.** | | | | | | | | Elektronisches Getriebe ausschalten (Variante nur für Spindeln) |
| | AXIS: Folgespindel | | | | | | | | |

| **Nibbeln** | | | | | |
|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | **Erläuterung** |
| | **1.** | **2.** | **3.** | **4.** | |
| PUNCHAAC | REAL: kleinster Lochabstand | REAL: Anfangsbeschleunigung | REAL: größter Lochabstand | REAL: Endbeschleunigung | Wegabhängige Beschleunigung aktivieren |

| Auskunftsfunktionen im passiven Filesystem | | | | |
|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | **Erläuterung** |
| | **1.** | **2.** | **3.** | |
| FILEDATE | VAR INT: Fehlermeldung | CHAR[160]: Dateiname | VAR CHAR[8]: Datum im Format "dd.mm.yy" | Liefert das Datum des letzten Schreibzugriffs auf eine Datei |
| FILETIME | VAR INT: Fehlermeldung | CHAR[160]: Dateiname | VAR CHAR[8]: Uhrzeit im Format "hh.mm.ss" | Liefert die Uhrzeit des letzten Schreibzugriffs auf eine Datei |
| FILESIZE | VAR INT: Fehlermeldung | CHAR[160]: Dateiname | VAR INT: Dateigröße | Liefert die aktuelle Größe einer Datei |
| FILESTAT | VAR INT: Fehlermeldung | CHAR[160]: Dateiname | VAR CHAR[5]: Datum im Format "rwxsd" | Liefert den Status einer Datei bezüglich folgender Rechte:<br>• Lesen (r: read)<br>• Schreiben (w: write)<br>• Ausführen (x: execute)<br>• Anzeigen (s: show)<br>• Löschen (d: delete) |
| FILEINFO | VAR INT: Fehlermeldung | CHAR[160]: Dateiname | VAR CHAR[32]: Datum im Format "rwxsd nnnnnnnn dd.mm.yy hh:mm:ss" | Liefert für eine Datei die Summe der Informationen, die über FILEDATE, FILETIME, FILESIZE und FILESTAT auslesbar sind |

| Achscontainer | | |
|---|---|---|
| **Bezeichner** | **Parameter** | **Erläuterung** |
| | **1. - n.** | |
| AXCTSWE | AXIS: Achscontainer | Achscontainer drehen |
| AXCTSWED | AXIS: Achscontainer | Achscontainer drehen (Befehlsvariante für die Inbetriebnahme!) |
| AXCTSWEC: | AXIS: Achscontainer | Freigabe zur Achscontainer-Drehung zurücknehmen |

| **Master/Slave-Kopplung** | | |
|---|---|---|
| **Bezeichner** | **Parameter** | **Erläuterung** |
| | **1. - n.** | |
| MASLON | AXIS:<br>Achsbezeichner | Master/Slave-Kopplung einschalten |
| MASLOF | AXIS:<br>Achsbezeichner | Master/Slave-Kopplung trennen |
| MASLOFS | AXIS:<br>Achsbezeichner | Master/Slave-Kopplung trennen und Slave-Spindeln automatisch abbremsen |
| MASLDEF | AXIS:<br>Achsbezeichner | Master/Slave-Kopplung definieren<br>Die letzte Achse ist die Master-Achse. |
| MASLDEL | AXIS:<br>Achsbezeichner | Master/Slave-Kopplung trennen und Definition des Verbandes löschen |

| **Online-Werkzeuglängenkorrektur** | | | |
|---|---|---|---|
| **Bezeichner** | **Parameter** | | **Erläuterung** |
| | **1.** | **2.** | |
| TOFFON | AXIS:<br>Korrekturrichtung | REAL:<br>Offset-Wert in Korrekturrichtung | Online- Werkzeuglängenkorrektur in der angegebenen Korrekturrichtung aktivieren |
| TOFFOF | AXIS:<br>Korrekturrichtung | | Online-Werkzeuglängenkorrektur in der angegebenen Korrekturrichtung zurücksetzen |

| **SERUPRO** | | |
|---|---|---|
| **Bezeichner** | **Parameter** | **Erläuterung** |
| IPTRLOCK | | Beginn des suchunfähigen Programmabschnitts |
| IPTRUNLOCK | | Ende des suchunfähigen Programmabschnitts |

| **Rückzug** | | |
|---|---|---|
| **Bezeichner** | **Parameter** | **Erläuterung** |
| | | |
| | **1. - n.** | |
| POLFMASK | AXIS:<br>Geometrie- bzw. Maschinenachsname | Achsen für den Schnellrückzug freigeben (ohne Zusammenhang zwischen den Achsen) |
| POLFMLIN | AXIS:<br>Geometrie- bzw. Maschinenachsname | Achsen für den linearen Schnellrückzug freigeben |
| | | |

| **Rückzug** | | | | |
|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | **Erläuterung** |
| POLFA | **1.** | **2.** | **3.** | Rückzugsposition für Einzelachsen |
| | AXIS:<br>Kanalachsbezeich-ner | INT:<br>Typ | REAL:<br>Wert | |

| **Kollisionsvermeidung** | | | |
|---|---|---|---|
| **Bezeichner** | **Parameter** | | **Erläuterung** |
| | | | |
| | **1.** | | |
| PROTA | STRING:<br>"R" | | Neuberechnung des Kollisionsmodells anfordern |
| | | | |
| PROTS | **1.** | **2. - n.** | Schutzbereichszustand setzen |
| | CHAR:<br>Status | STRING:<br>Schutzbe-reichsname | |

| **Intelligente Lastanpassung** | | | | | |
|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | **Erläuterung** |
| | | | | | |
| CADAPTON | **1.** | **2.** | **3.** | **4.** | Lastanpassung einschalten |
| | INT:<br>Status | AXIS:<br>Maschinen-achsname | INT:<br>Eingangsgrö-ße | REAL:<br>Eingangswert (optional) | |
| | | | | | |
| CADAPTOF | **1.** | **2.** | **3.** | | Lastanpassung ausschalten |
| | INT:<br>Status | AXIS:<br>Maschinen-achsname | INT:<br>Eingangsgrö-ße | | |

| **Ruckanpassung** | | |
|---|---|---|
| **Bezeichner** | **Parameter** | **Erläuterung** |
| CALCFIR | | FIR-Ruckfilter an Dynamikmodus anpassen<br>Eingesetzt wird CALCFIR im Herstellerzyklus CUST_832. |

# 4.5 Vordefinierte Prozeduren in Synchronaktionen

Die folgenden vordefinierten Prozeduren stehen ausschließlich in Synchronaktionen zur Verfügung.

| Synchronprozeduren | | |
|---|---|---|
| **Bezeichner** | **Parameter** | **Erläuterung** |
| | | |
| STOPREOF | | Vorlaufstopp aufheben<br>Eine Synchronaktion mit einem STOPREOF-Befehl bewirkt einen Vorlaufstopp nach dem nächsten Ausgabesatz (= Satz an den Hauptlauf). Der Vorlaufstopp wird aufgehoben mit dem Ende des Ausgabesatzes oder wenn die STOPREOF-Bedingung erfüllt ist. Sämtliche Synchronaktionsanweisungen mit STOPREOF-Befehl gelten dann als bearbeitet. |
| RDISABLE | | Einlesesperre |
| | | |
| DELDTG | **1.**<br>AXIS:<br>Achse für axiales Restweglöschen (optional). Entfällt die Achse, wird Restweglöschen für Bahnweg angestoßen. | Restweglöschen<br>Eine Synchronaktion mit einem DELDTG-Befehl bewirkt einen Vorlaufstopp nach dem nächsten Ausgabesatz (= Satz an den Hauptlauf). Der Vorlaufstopp wird aufgehoben mit dem Ende des Ausgabesatzes oder wenn die erste DELDTG-Bedingung erfüllt ist. In $AA_DELT[<Achse>] findet man die axiale Entfernung zum Zielpunkt bei axialem Restweglöschen, in $AC_DELT den Bahnrestweg. |

| Programmkoordinierung Technologiezyklen | | |
|---|---|---|
| **Bezeichner** | **Parameter** | **Erläuterung** |
| | | |
| | **1.** | |
| LOCK | INT:<br>ID der Synchronaktion, die gesperrt werden soll | Synchronaktion mit ID sperren bzw. Technologiezyklus stoppen<br>Es können eine oder mehrere IDs programmiert werden. |
| UNLOCK | INT:<br>ID der Synchronaktion, die freigegeben werden soll | Synchronaktion mit ID freigeben bzw. Technologiezyklus fortsetzen<br>Es können eine oder mehrere IDs programmiert werden. |
| | | |
| ICYCON | | Jeden Satz eines Technologiezyklus nach ICYCON in einem separaten IPO-Takt abarbeiten |
| ICYCOF | | Alle Sätze eines Technologiezyklus nach ICYCOF in einem IPO-Takt abarbeiten |

| **Polynomfunktionen** | | | | | | |
|---|---|---|---|---|---|---|
| **Bezeichner** | **Parameter** | | | | | **Erläuterung** |
| | | | | | | |
| SYNFCT | **1.** | **2.** | **3.** | | | Wenn in der Bewegungssynchronaktion die Bedingung erfüllt ist, wird das durch den ersten Ausdruck bestimmte Polynom an der Eingangsvariablen ausgewertet. Der Wert wird dann nach unten und nach oben begrenzt und der Ergebnisvariablen zugewiesen. |
| | INT: Nummer der Polynomfunktion, die mit FCTDEF definiert wurde | VAR REAL: Ergebnisvariable *) | VAR REAL: Eingangsvariable **) | | | |
| | | | | | | |
| FTOC | **1.** | **2.** | **3.** | **4.** | **5.** | Ändern der Werkzeugfeinkorrektur in Abhängigkeit einer mit FCTDEF festgelegten Funktion (Polynom max. 3. Grades).<br>Bei FCTDEF muss die hier verwendete Nummer angegeben werden. |
| | INT: Nummer der Polynomfunktion, die mit FCTDEF definiert wurde | VAR REAL: Eingangsvariable **) | INT: Länge 1, 2, 3 | INT: Kanalnummer | INT: Spindelnummer | |

*) Als Ergebnisvariable sind nur spezielle Systemvariablen zulässig (siehe Funktionshandbuch Synchronaktionen).

**) Als Eingangsvariable sind nur spezielle Systemvariablen zulässig (siehe Funktionshandbuch Synchronaktionen).

# 4.6 Vordefinierte Funktionen

Durch den Aufruf einer vordefinierten Funktion wird die Ausführung einer vordefinierten NC-Funktion angestoßen, die im Unterschied zur vordefinierten Prozedur einen Rückgabewert liefert. Der Aufruf der vordefinierten Funktion kann als Operand im Ausdruck stehen.

| **Koordinatensystem** | | | | | | |
|---|---|---|---|---|---|---|
| **Bezeichner** | **Rückgabewert** | **Parameter** | | | | **Erläuterung** |
| | | | | | | |
| | | **1.** | **2.** | **3. - 15.** | **4. - 16.** | |
| CTRANS | FRAME | AXIS: Achsbezeichner | REAL: Verschiebung | AXIS: Achsbezeichner | REAL: Verschiebung | Translation: Nullpunktverschiebung GROB für mehrere Achsen |
| CFINE | FRAME | AXIS: Achsbezeichner | REAL: Verschiebung | AXIS: Achsbezeichner | REAL: Verschiebung | Translation: Nullpunktverschiebung FINE für mehrere Achsen |
| CSCALE | FRAME | AXIS: Achsbezeichner | REAL: Maßstabsfaktor | AXIS: Achsbezeichner | REAL: Maßstabsfaktor | Scale: Maßstabsfaktor für mehrere Achsen |
| | | | | | | |
| | | **1.** | **2.** | **3. und 5.** | **4. und 6.** | |
| CROT | FRAME | AXIS: Achsbezeichner | REAL: Drehung | AXIS: Achsbezeichner | REAL: Drehung | Rotation: Drehung des aktuellen Koordinatensystems<br>Maximale Parameteranzahl: 6<br>(je ein Achsbezeichner und Wert pro Geometrieachse). |
| CROTS | FRAME | AXIS: Achsbezeichner | REAL: Drehung mit Raumwinkel | AXIS: Achsbezeichner | REAL: Drehung mit Raumwinkel | Rotation: Drehung des aktuellen Koordinatensystems mit Raumwinkel<br>Maximale Parameteranzahl: 6<br>(je ein Achsbezeichner und Wert pro Geometrieachse). |
| | | | | | | |
| CMIRROR | | **1.** | **2. - 8.** | | | Mirror: Spiegeln an einer Koordinatenachse |
| | FRAME | AXIS | AXIS | | | |
| | | | | | | |

| **Koordinatensystem** | | | | | | |
|---|---|---|---|---|---|---|
| **Bezeichner** | **Rückgabewert** | **Parameter** | | | | **Erläuterung** |
| | | **1.** | **2.** | | | |
| CRPL | FRAME | INT: Drehachse | REAL: Drehwinkel | | | Frame-Drehung in einer beliebigen Ebene |
| ADDFRAME | INT:<br>0: OK<br>1: Zielangabe (String) ist falsch<br>2: Ziel-Frame ist nicht projektiert<br>3: Drehung im Frame ist nicht erlaubt | FRAME: additives gemessenes oder berechnetes Frame | STRING: spezifiziert Zielframe | | | Berechnet den Ziel-Frame, der durch den String spezifiziert ist<br>Der Ziel-Frame wird so berechnet, dass sich das neue Gesamt-Frame als Verkettung des alten Gesamt-Frames mit dem übergebenen Frame ergibt. |
| | | | | | | |
| INVFRAME | FRAME | **1.**<br>FRAME | | | | Berechnet aus einem Frame den inversen Frame<br>Die Frame-Verkettung eines Frames mit seinem inversen Frame ergibt immer einen Null-Frame |
| | | | | | | |
| MEAFRAME | FRAME | **1.**<br>REAL[3,3]: Koordinaten der gemessenen Raumpunkte | **2.**<br>REAL[3,3]: Koordinaten der Sollpunkte | **3.**<br>VAR REAL: Variable, mit der Informationen zur Qualität der FRAME-Berechnung zurückgegeben werden | | Frame-Berechnung aus 3 Messpunkten im Raum |

| **Geometrie-Funktionen** | | | | | |
|---|---|---|---|---|---|
| **Bezeichner** | **Rückgabewert** | **Parameter** | | | **Erläuterung** |
| | | **1.** | **2.** | **3.** | |
| CALCDAT | BOOL:<br>Fehlerstatus | VAR REAL [n, 2]:<br>Tabelle (Abszisse, Ordinate) der Punkte 1 bis n | INT:<br>Anzahl Punkte | VAR REAL [3]:<br>Ergebnis: Abszisse, Ordinate und Radius des errechneten Kreismittelpunkts | Berechnet die Mittelpunktskoordinaten und den Radius des Kreises aus 3 oder 4 Punkten.<br>Die Punkte müssen verschieden sein. |
| INTERSEC | BOOL:<br>Fehlerstatus | VAR REAL [11]:<br>Erstes Konturelement | VAR REAL [11]:<br>Zweites Konturelement | VAR REAL [2]:<br>Ergebnisvektor zur Schnittpunktkoordinate: Abszisse und Ordinate | Berechnet die Schnittpunktkoordinaten zwischen zwei Konturelementen.<br>Der Fehlerstatus gibt an, ob ein Schnittpunkt gefunden wurde. |

| **Kurventabellen-Funktionen** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Bezeichner** | **Rückgabewert** | **Parameter** | | | | | | **Erläuterung** |
| | | **1.** | **2.** | **3.** | **4.** | **5.** | **6.** | |
| CTAB | REAL:<br>Folgeachsposition | REAL:<br>Leitachsposition | INT:<br>Tabellennummer | VAR REAL[ ]:<br>Ergebnis Steigung | AXIS:<br>Folgeachse für Skalierung | AXIS:<br>Leitachse für Skalierung | | Ermittelt die Folgeachsposition zur angegebenen Leitachsposition aus Kurventabelle.<br>Sind Parameter 4/5 nicht programmiert, wird mit Standard-Skalierung gerechnet. |
| CTABINV | REAL:<br>Leitachsposition | REAL:<br>Folgeachsposition | REAL:<br>Leitposition | INT:<br>Tabellennummer | VAR REAL[ ]:<br>Ergebnis Steigung | AXIS:<br>Folgeachse für Skalierung | AXIS:<br>Leitachse für Skalierung | Ermittelt die Leitachsposition zur angegebenen Folgeachsposition aus Kurventabelle.<br>Sind Parameter 5/6 nicht programmiert, wird mit Standard-Skalierung gerechnet. |
| CTABID | INT:<br>Kurventabellennummer | INT:<br>Eintragsnummer im Speicher | STRING:<br>Speicherort:<br>"SRAM",<br>"DRAM" | | | | | Ermittelt die Kurventabellennummer, die unter der angegebenen Nummer im Speicher eingetragen ist. |

| **Kurventabellen-Funktionen** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Bezeichner** | **Rückgabewert** | **Parameter** | | | | | | **Erläuterung** |
| | | **1.** | **2.** | **3.** | **4.** | **5.** | **6.** | |
| CTABISLOCK | INT: Sperrzustand | INT: Tabellennummer | | | | | | Ermittelt die Sperrzustand der Kurventabelle:<br>> 0: Tabelle ist gesperrt<br>1: CTABLOCK<br>2: aktive Kopplung<br>3: CTABLOCK und aktive Kopplung<br>0: Tabelle ist nicht gelockt<br>-1: Tabelle existiert nicht |
| CTABEXISTS | INT: Existenz | INT: Tabellennummer | | | | | | Ermittelt die Existenz der Kurventabelle im statischen oder dynamischen NC-Speicher:<br>0: FALSE<br>1: TRUE |
| CTABMEMTYP | INT: Speicherort | INT: Tabellennummer | | | | | | Ermittelt die Speicherort der Kurventabelle:<br>1: DRAM<br>0: SRAM<br>-1: Tabelle existiert nicht |
| CTABPERIOD | INT: Periodizität | INT: Tabellennummer | | | | | | Ermittelt die Periodizität der Kurventabelle:<br>0: nicht periodisch<br>1: periodisch in Leitachse<br>2: periodisch in Leit- und Folgeachse<br>-1: Tabelle existiert nicht |
| CTABNO | INT: Anzahl Kurventabellen | | | | | | | Ermittelt die Anzahl der definierten Kurventabellen (im statischen und dynamischen NC-Speicher) |
| CTABNOMEM | INT: Anzahl Kurventabellen | STRING: Speicherort: "SRAM", "DRAM" | | | | | | Ermittelt die Anzahl der definierten Kurventabellen im angegebenen Speicher |

| **Kurventabellen-Funktionen** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Bezeichner** | **Rückgabewert** | **Parameter** | | | | | | **Erläuterung** |
| | | **1.** | **2.** | **3.** | **4.** | **5.** | **6.** | |
| CTABFNO | INT: Anzahl der Tabellen | STRING: Speicherort: "SRAM", "DRAM" | | | | | | Ermittelt die Anzahl der noch möglichen Kurventabellen im angegebenen Speicher |
| CTABSEG | INT: Anzahl der Kurvensegmente | STRING: Speicherort: "SRAM", "DRAM" | STRING: Segmentart: "L": Linear "P": Polynom | | | | | Ermittelt die Anzahl der verwendeten Kurvensegmente der angegebenen Segmentart im angegebenen Speicher<br>>=0: Anzahl<br>-1: Speichertyp ungültig<br>Wird Parameter 2 nicht programmiert, wird die Summe aus Linear- und Polynom-Segmenten ausgegeben. |
| CTABFSEG | INT: Anzahl der Kurvensegmente | STRING: Speicherort: "SRAM", "DRAM" | STRING: Segmentart: "L": Linear "P": Polynom | | | | | Ermittelt die Anzahl der noch möglichen Kurvensegmente der angegebenen Segmentart im angegebenen Speicher<br>>=0: Anzahl<br>-1: Speichertyp ungültig |
| CTABSEGID | INT: Anzahl der Kurvensegmente | INT: Tabellennummer | STRING: Segmentart: "L": Linear "P": Polynom | | | | | Ermittelt die Anzahl der Kurvensegmente der angegebenen Segmentart, die von der Kurventabelle verwendet werden<br>>=0: Anzahl<br>-1: Tabelle existiert nicht |
| CTABMSEG | INT: Anzahl der Kurvensegmente | STRING: Speicherort: "SRAM", "DRAM" | STRING: Segmentart: "L": Linear "P": Polynom | | | | | Ermittelt die Anzahl der maximal möglichen Kurvensegmente der angegebenen Segmentart im angegebenen Speicher<br>>=0: Anzahl<br>-1: Tabelle existiert nicht |

| Kurventabellen-Funktionen | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Bezeichner** | **Rückgabewert** | **Parameter** | | | | | | **Erläuterung** |
| | | **1.** | **2.** | **3.** | **4.** | **5.** | **6.** | |
| CTABPOL | INT: Anzahl der Kurvenpolynome | STRING: Speicherort: "SRAM", "DRAM" | | | | | | Ermittelt die Anzahl der verwendeten Kurvenpolynome im angegebenen Speicher<br>>=0: Anzahl<br>-1: Tabelle existiert nicht |
| CTABPOLID | INT: Anzahl der Kurvenpolynome | INT: Tabellennummer | | | | | | Ermittelt die Anzahl der Kurvenpolynome, die von der Kurventabelle verwendet werden<br>>=0: Anzahl<br>-1: Tabelle existiert nicht |
| CTABFPOL | INT: Anzahl der Kurvenpolynome | STRING: Speicherort: "SRAM", "DRAM | | | | | | Ermittelt die Anzahl der maximal möglichen Kurvenpolynome im angegebenen Speicher:<br>>=0: Anzahl<br>-1: Tabelle existiert nicht |
| CTABMPOL | INT: Anzahl der Kurvenpolynome | STRING: Speicherort: "SRAM", "DRAM | | | | | | Ermittelt die Anzahl der maximal möglichen Kurvenpolynome im angegebenen Speicher:<br>>=0: Anzahl<br>-1: Tabelle existiert nicht |
| CTABSSV | REAL: Folgeachsposition | REAL: Leitachsposition | INT: Tabellennummer | VAR REAL[ ]: Ergebnis Steigung | AXIS: Folgeachse für Skalierung | AXIS: Leitachse für Skalierung | | Ermittelt die Folgeachsposition am Anfang des zum angegebenen Leitachswert gehörenden Kurvensegments. |
| CTABSEV | REAL: Folgeachsposition | REAL: Leitachsposition | INT: Tabellennummer | VAR REAL[ ]: Ergebnis Steigung | AXIS: Folgeachse für Skalierung | AXIS: Leitachse für Skalierung | | Ermittelt die Folgeachsposition am Ende des zum angegebenen Leitachswert gehörenden Kurvensegments. |
| CTABTSV | REAL: Folgeachsposition | INT: Tabellennummer | VAR REAL[ ]: Ergebnis Steigung Start der Tabelle | AXIS: Folgeachse | | | | Ermittelt die Folgeachsposition am Anfang der Kurventabelle. |

| **Kurventabellen-Funktionen** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Bezeichner** | **Rückgabewert** | **Parameter** | | | | | | **Erläuterung** |
| | | **1.** | **2.** | **3.** | **4.** | **5.** | **6.** | |
| CTABTEV | REAL: Folgeachsposition | INT: Tabellennummer | VAR REAL[ ]: Ergebnis Steigung Ende der Tabelle | AXIS: Folgeachse | | | | Ermittelt die Folgeachsposition am Ende der Kurventabelle. |
| CTABTSP | REAL: Leitachsposition | INT: Tabellennummer | VAR REAL[ ]: Ergebnis Steigung Start der Tabelle | AXIS: Leitachse | | | | Ermittelt die Leitachsposition am Anfang der Kurventabelle. |
| CTABTEP | REAL: Leitachsposition | INT: Tabellennummer | VAR REAL[ ]: Ergebnis Steigung Ende der Tabelle | AXIS: Leitachse | | | | Ermittelt die Leitachsposition am Ende der Kurventabelle. |
| CTABTMIN | REAL: Minimalwert | INT: Tabellennummer | REAL: Leitwertintervall untere Grenze | REAL: Leitwertintervall obere Grenze | AXIS: Folgeachse | AXIS: Leitachse | | Ermittelt den Minimalwert der Folgeachse im gesamten Definitionsbereich der Kurventabelle oder in einem definierten Intervall. |
| CTABTMAX | REAL: Maximalwert | INT: Tabellennummer | REAL: Leitwertintervall untere Grenze | REAL: Leitwertintervall obere Grenze | AXIS: Folgeachse | AXIS: Leitachse | | Ermittelt den Maximalwert der Folgeachse im gesamten Definitionsbereich der Kurventabelle oder in einem definierten Intervall. |
| **Hinweis:**<br>Die Kurventabellen-Funktionen können auch in Synchronaktionen programmiert werden. | | | | | | | | |

| **Achsfunktionen** | | | | | | |
|---|---|---|---|---|---|---|
| **Bezeichner** | **Rückgabewert** | **Parameter** | | | | |
| | | **1.** | **2.** | **3.** | **4.** | **Erläuterung** |
| AXNAME | AXIS: Achsbezeichner | STRING [ ]: Eingangsstring | | | | Konvertiert Eingangsstring in Achsbezeichner |
| AXSTRING | STRING[ ]: Achsname | AXIS: Achsbezeichner | | | | Konvertiert Achsbezeichner in String |

| **Achsfunktionen** | | | | | | |
|---|---|---|---|---|---|---|
| **Bezeichner** | **Rückgabewert** | **Parameter** | | | | |
| | | **1.** | **2.** | **3.** | **4.** | **Erläuterung** |
| ISAXIS | BOOL:<br>Achse vorhanden (TRUE) oder nicht (FALSE) | INT:<br>Nummer der Geometrieachse<br>(1 bis 3) | | | | Prüft, ob die als Parameter angegebene Geometrieachse 1 bis 3 entsprechend Maschinendatum MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB vorhanden ist |
| SPI | AXIS:<br>Achsbezeichner | INT:<br>Spindelnummer | | | | Konvertiert Spindelnummer in Achsbezeichner |
| AXTOSPI | INT:<br>Spindelnummer | AXIS:<br>Achsbezeichner | | | | Konvertiert Achsbezeichner in Spindelnummer |
| MODAXVAL | REAL:<br>Modulo-Wert | AXIS:<br>Achsbezeichner | REAL:<br>Achsposition | | | Berechnet aus der eingegebenen Achsposition den Modulo-Rest<br>Ist die angegebene Achse keine Modulo-Achse, wird die Achsposition unverändert zurückgegeben. |
| POSRANGE | BOOL:<br>Sollposition innerhalb des Positionsfensters (TRUE) oder nicht (FALSE) | AXIS:<br>Achsbezeichner | REAL:<br>Referenzposition im Koordinatensystem | REAL:<br>Positionsfensterbreite | INT:<br>Koordinatensystem | Ermittelt, ob sich die Sollposition einer Achse in einem Fenster um eine vorgegebene Referenzposition befindet |

| **Werkzeugverwaltung** | | | | | |
|---|---|---|---|---|---|
| **Bezeichner** | **Rückgabewert** | **Parameter** | | | **Erläuterung** |
| | | **1.** | **2.** | **3.** | |
| CHKDM | INT:<br>Status: Ergebnis der Prüfung | INT:<br>Magazinnummer | INT:<br>D-Nummer | | Prüft die Eindeutigkeit der D-Nummer innerhalb eines Magazins |
| CHKDNO | INT:<br>Status: Ergebnis der Prüfung | INT:<br>T-Nummer des 1. Werkzeugs | INT:<br>T-Nummer des 2. Werkzeugs | INT:<br>D-Nummer | Prüft die Eindeutigkeit der D-Nummer |
| GETACTT | INT:<br>Status | INT:<br>T-Nummer | STRING[32]:<br>Werkzeugname | | Ermittelt das aktive Werkzeug aus einer Gruppe von gleichnamigen Werkzeugen |
| GETACTTD | INT:<br>Status: Ergebnis der Prüfung | VAR INT:<br>gefundene T-Nummer (Rückgabewert) | INT:<br>D-Nummer | | Ermittelt zur absoluten D-Nummer die zugehörige T-Nummer |

| **Werkzeugverwaltung** | | | | | |
|---|---|---|---|---|---|
| **Bezeichner** | **Rückgabewert** | **Parameter** | | | **Erläuterung** |
| | | **1.** | **2.** | **3.** | |
| GETDNO | INT:<br>D-Nummer | INT:<br>T-Nummer | INT:<br>Schneidennummer | | Ermittelt die D-Nummer der Schneide des Werkzeugs T |
| GETT | INT:<br>T-Nummer | STRING[32]:<br>Werkzeugname | INT:<br>Duplo-Nummer | | Ermittelt die T-Nummer zum Werkzeugnamen |
| NEWT | INT:<br>T-Nummer | STRING[32]:<br>Werkzeugname | INT:<br>Duplo-Nummer | | Legt neues Werkzeug an (Werkzeugdaten bereitstellen)<br>Die Duplo-Nummer kann entfallen. |
| TOOLENV | INT:<br>Status | STRING:<br>Name | | | Speichert die Werkzeugumgebung mit angegebenem Namen im statischen NC-Speicher |
| DELTOOLENV | INT:<br>Status | STRING:<br>Name | | | Löscht die Werkzeugumgebung mit angegebenem Namen im statischen NC-Speicher<br>Löscht alle Werkzeugumgebungen, wenn kein Name angegeben ist. |
| GETTENV | INT:<br>Status | STRING:<br>Name | VAR INT:<br>T-Nummer [0]<br>D-Nummer [1]<br>DL-Nummer [2] | | Ermittelt T-Nummer D-Nummer und DL-Nummer aus einer Werkzeugumgebung mit angegebenem Namen |

| **Arithmetik** | | | | | |
|---|---|---|---|---|---|
| **Bezeichner** | **Rückgabewert** | **Parameter** | | | **Erläuterung** |
| | | **1.** | **2.** | **3.** | |
| SIN | REAL | REAL | | | Sinus |
| ASIN | REAL | REAL | | | Arcus-Sinus |
| COS | REAL | REAL | | | Cosinus |
| ACOS | REAL | REAL | | | Arcus-Cosinus |
| TAN | REAL | REAL | | | Tangens |
| ATAN2 | REAL | REAL | REAL | | Arcus-Tangens 2 |
| SQRT | REAL | REAL | | | Quadratwurzel |
| POT | REAL | REAL | | | Quadrat |
| TRUNC | REAL | REAL | | | Ganzzahliger Anteil |
| ROUND | REAL | REAL | | | Abrunden |
| ROUNDUP | REAL | REAL | | | Aufrunden |
| ABS | REAL | REAL | | | Absolutwert |
| LN | REAL | REAL | | | Natürlicher Logarithmus |
| EXP | REAL | REAL | | | Exponentialfunktion $e^x$ |
| MINVAL | REAL | REAL | REAL | | Ermittelt den kleineren Wert der beiden Parameter |

| **Arithmetik** | | | | | |
|---|---|---|---|---|---|
| **Bezeichner** | **Rückgabewert** | **Parameter** | | | **Erläuterung** |
| | | **1.** | **2.** | **3.** | |
| MAXVAL | REAL | REAL | REAL | | Ermittelt den größeren Wert der beiden Parameter |
| BOUND | REAL: Prüfstatus | REAL: Untergrenze | REAL: Obergrenze | REAL: Vergleichswert | Ermittelt, ob der Vergleichswert innerhalb der Grenzen liegt. |
| **Hinweis:**<br>Die Arithmetik Funktionen können auch in Synchronaktionen programmiert werden. Die Berechnung bzw. Auswertung dieser Arithmetik-Funktionen erfolgt dann im Hauptlauf. Für Berechnungen und als Zwischenspeicher kann auch der Synchronaktions-Parameter $AC_PARAM[<n>] genutzt werden. | | | | | |

| **String-Funktionen** | | | | | |
|---|---|---|---|---|---|
| **Bezeichner** | **Rückgabewert** | **Parameter** | | | **Erläuterung** |
| | | **1.** | **2.** | **3.** | |
| ISNUMBER | BOOL | STRING: Eingangsstring | | | Prüft, ob der Eingangsstring in eine Zahl gewandelt werden kann |
| NUMBER | REAL | STRING: Eingangsstring | | | Wandelt den Eingangsstring in eine Zahl |
| TOUPPER | STRING | STRING: Eingangsstring | | | Wandelt den Eingangsstring in Großbuchstaben |
| TOLOWER | STRING | STRING: Eingangsstring | | | Wandelt den Eingangsstring in Kleinbuchstaben |
| STRLEN | INT | STRING: Eingangsstring | | | Ermittelt die Länge des Eingangsstrings bis Stringende (/0) |
| INDEX | INT | STRING: Eingangsstring | CHAR: Suchzeichen | | Ermittelt die Stelle des Zeichens im Eingangsstring von links nach rechts.<br>Das 1. Zeichen des Strings von links hat den Index 0. |
| RINDEX | INT | STRING: Eingangsstring | CHAR: Suchzeichen | | Ermittelt die Stelle des Zeichens im Eingangsstring von rechts nach links.<br>Das 1. Zeichen des Strings von rechts hat den Index 0. |
| MINDEX | INT | STRING: Eingangsstring | STRING: Suchzeichen | | Ermittelt die Stelle eines der im 2. Parameter angegebenen Zeichen im Eingangsstring von links nach rechts.<br>Das 1. Zeichen des Eingangsstrings von links hat den Index 0. |
| SUBSTR | STRING | STRING: Eingangsstring | INT | INT | Ermittelt den durch Beginn (2. Parameter) und Anzahl von Zeichen (3. Parameter) beschriebenen Teilstring des Eingangsstrings. |
| SPRINT | STRING | STRING: Eingangsstring | | | Ermittelt den formatierten Eingangsstring |

| **Funktionen für Messzyklen** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Bezeichner** | **Rückgabewert** | **Parameter** | | | | | | **Erläuterung** |
| | | **1.** | **2.** | **3.** | **4.** | **5.** | **6.** | |
| CALCPOSI | INT: Status | REAL[3]: Ausgangsposition im WKS | REAL[3]: inkrementelle Wegvorgabe bezogen auf Ausgangsposition | REAL[5]: Mindestabstände zu den Überwachungsgrenzen | REAL[3]: Rückgabefeld für den mögl. inkr. Weg | BOOL: Umrechnung des Maßsystems ja/ nein | INT: Art der Grenzüberwachung | Prüft, ob ausgehend von einem gegebenen Startpunkt die Geometrieachsen einen vorgegebenen Weg verfahren können, ohne die Achsgrenzen zu verletzen<br>Für den Fall, das der vorgegebene Weg ohne Verletzungen nicht gefahren werden kann, wird der maximal zulässige Wert zurückgegeben. |
| GETTCOR | INT: Status | REAL [11]: | STRING: WZ-Längenkomponente : Koordinatensystem | STRING: Name der WZ-Umgebung | INT: interne T-Nr. des Werkzeugs | INT: Schneidennummer (D-Nr.) des Werkzeugs | INT: Nummer der ortsabhängigen Korrektur (DL-Nr. des WZs)) | Ermittelt die Werkzeuglängen und Werkzeuglängenkomponenten aus WZ-Umgebung bzw. aktueller Umgebung |
| LENTOAX | INT: Status | INT[3]: Achszuordnung der Geoachsen | REAL[3]: Matrix zur Abbildung der WZ-Längen im Koordinatensystem | STRING: Koordinatensystem für die Zuordnung | | | | Ermittelt Informationen über die Zuordnung der WZ-Längen L1, L2, L3 des aktiven WZs zu Abszisse, Ordinate, Applikate<br>Die Zuordnung zu den Geometrieachsen wird durch Frames und die aktive Ebene (G17 -G19) beeinflusst. |

| SETTCOR | INT: Status | 1. | 2. | 3. | 4. | 5. | 6. | 7. | 8. | 9. | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | REAL [3]: Korrekturvektor im Raum | STR.: Komponentenbezeichner | INT: zu korr. Komponente(n) 0 - 11 | INT: Art der Schreiboperation 0 - 3 | INT: Index der Geoachse | STRING: Name der WZ-Umgebung | INT: int. T-Nr. des WZs | INT: D-Nr. des WZs | INT: DL-Nr. des WZs | Verändert Werkzeugkomponenten unter Berücksichtigung aller Randbedingungen, die in die Bewertung der einzelnen Komponenten eingehen |

| **Sonstige Funktionen** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Bezeichner** | **Rückgabewert** | **Parameter** | | | | | | **Erläuterung** |
| | | **1.** | **2.** | **3.** | **4.** | **5.** | **6.** | |
| STRINGIS | INT: Info über den String | STRING: Name des zu prüfenden Elements | | | | | | Prüft, ob der angegebene String als Element der NC-Programmiersprache im aktuellen Sprachumfang zur Verfügung steht |
| ISVAR | BOOL: Variable bekannt ja/nein | STRING: Name der Variablen | | | | | | Prüft, ob der Übergabeparameter eine in der NC bekannte Variable enthält (Maschinendatum, Settingdatum, Systemvariable, allgemeine Variablen wie GUD's) |
| GETVARTYP | INT: Datentyp | STRING: Name der Variablen | | | | | | Ermittelt den Datentyp einer System- /Anwendervariablen |
| GETVARPHU | INT: Zahlenwert der physikalischen Einheit | STRING: Name der Variablen | | | | | | Ermittelt die physikalische Einheit einer System- /Anwendervariablen |
| GETVARAP | INT: Schutzstufe für den Zugriff | STRING: Name der Variablen | STRING: Art des Zugriffs | | | | | Ermittelt das Zugriffsrecht auf eine System- / Anwendervariable |

| Sonstige Funktionen | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Bezeichner** | **Rückgabewert** | **Parameter** | | | | | | **Erläuterung** |
| | | **1.** | **2.** | **3.** | **4.** | **5.** | **6.** | |
| GETVARLIM | INT: Status | STRING: Name der Variablen | CHAR: Gibt an, welcher Grenzwert ausgelesen werden soll | VAR REAL: Rückgabe des Grenzwerts | | | | Ermittelt den unteren/ oberen Grenzwert einer System- /Anwendervariablen |
| GETVARDFT | INT: Status | STRING: Name der Variablen | VAR REAL/ STRING/ FRAME: Rückgabe des Standardwerts | INT: Index auf erste Dimension (optional) | INT: Index auf zweite Dimension (optional) | INT: Index auf dritte Dimension (optional) | | Ermittelt den Standardwert einer System- / Anwendervariablen |
| COLLPAIR | INT: Prüfergebnis | STRING: Name des 1. Schutzbereichs | STRING: Name des 2. Schutzbereichs | BOOL: Alarmunterdrückung(optional) | | | | Prüft auf Zugehörigkeit zu einem Kollisionspaar |
| PROTD | REAL: Abstand der beiden Schutzbereiche | STRING: Name des 1. Schutzbereichs | STRING: Name des 2. Schutzbereichs | VAR REAL: Rückgabewert: 3-dimensionaler Abstandsvektor | BOOL: Maßsystem für Abstand und Abstandsvektor (optional) | | | Ermittelt den Abstand der beiden angegebenen Schutzbereiche |
| DELOBJ | INT: Fehlernummer | STRING: Typ der zu löschenden Komponente | INT: Start-Index der zu löschenden Komponenten (optional) | INT: Ende-Index der zu löschenden Komponenten (optional) | BOOL: Alarmunterdrückung (optional) | | | Löscht Elementen von kinematischen Ketten, Schutzbereichen, Schutzbereichselementen, Kollisionspaaren und Transformationsdaten |
| NAMETOINT | INT: Systemvariablenindex | STRING: Name des Systemvariablenfeldes | STRING: Zeichenkette/ Name | BOOL: Alarmunterdrückung (optional) | | | | Ermittelt anhand der Zeichenkette den zugehörigen Systemvariablenindex |

| Sonstige Funktionen | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Bezeichner** | **Rückgabewert** | **Parameter** | | | | | | **Erläuterung** |
| | | **1.** | **2.** | **3.** | **4.** | **5.** | **6.** | |
| ORISOLH | INT:<br>Fehlernummer | INT:<br>Steuert das Verhalten der Funktion | REAL:<br>Erster Winkel | REAL:<br>Zweiter Winkel | | | | Unterstützt den Anwender dabei, die Rundachspositionen einer Maschine so einzustellen, dass ein Drehwerkzeug in eine definierte, kinematikunabhängige Lage relativ zum Werkstück gebracht wird.<br>Voraussetzung:<br>Es ist eine 6-Achstransformation aktiv ist, die mit kinematischen Ketten parametriert ist. |
| CORRTRAFO | INT:<br>Fehlernummer | REAL:<br>Korrekturvektor | INT:<br>Zu modifizierendes Element | INT:<br>Korrekturmodus | BOOL:<br>Alarmunterdrückung (optional) | | | Offsetvektoren oder Richtungsvektoren der Orientierungsachsen im kinematischen Modell der Maschine modifizieren. |
| CORRTC | INT:<br>Fehlernummer | REAL:<br>Korrekturvektor | INT:<br>Zu modifizierendes Element | INT:<br>Korrekturmodus | BOOL:<br>Alarmunterdrückung (optional) | | | Offsetvektoren oder Richtungsvektoren orientierbarer Werkzeugträger nach Maschinenvermessung modifizieren. |

## 4.7 Aktuelle Sprache im HMI

Die folgende Tabelle enthält alle auf der Bedienoberfläche verfügbaren Sprachen.

Die aktuell eingestellte Sprache ist im Teileprogramm und in Synchronaktionen über folgende Systemvariable abfragbar:

$AN_LANGUAGE_ON_HMI = <Wert>

| <Wert> | Sprache | Sprachkürzel |
|---|---|---|
| 1 | Deutsch (Deutschland) | DEU |
| 2 | Französisch | FRA |
| 3 | Englisch (Vereinigtes Königreich) | ENG |
| 4 | Spanisch | ESP |
| 6 | Italienisch | ITA |
| 7 | Niederländisch | NLD |
| 8 | Chinesisch (vereinfacht) | CHS |
| 9 | Schwedisch | SVE |
| 18 | Ungarisch | HUN |
| 19 | Finnisch | FIN |
| 28 | Tschechisch | CSY |
| 50 | Portugiesisch (Brasilien) | PTB |
| 53 | Polnisch | PLK |
| 55 | Dänisch | DAN |
| 57 | Russisch | RUS |
| 68 | Slowakisch | SKY |
| 72 | Rumänisch | ROM |
| 80 | Chinesisch (traditionell) | CHT |
| 85 | Koreanisch | KOR |
| 87 | Japanisch | JPN |
| 89 | Türkisch | TRK |

**Hinweis**

Eine Aktualisierung von $AN_LANGUAGE_ON_HMI erfolgt:

- nach Systemhochlauf.
- nach NC- und/oder PLC-Reset.
- nach dem Umschalten auf einen anderen NC im Rahmen von M2N.
- nach Sprachumschaltung auf HMI.

# Anhang

# A

## A.1 Liste der Abkürzungen

| A | |
|---|---|
| A | Ausgang |
| ADI4 | Analog Drive Interface for 4 Axes |
| AC | Adaptive Control |
| ALM | Active Line Module |
| ARM | Asynchroner rotatorischer Motor |
| AS | Automatisierungssystem |
| ASCII | American Standard Code for Information Interchange: Amerikanische Code-Norm für den Informationsaustausch |
| ASIC | Application Specific Integrated Circuit: Anwender-Schaltkreis |
| ASUP | Asynchrones Unterprogramm |
| AUXFU | Auxiliary Function: Hilfsfunktion |
| AWL | Anweisungsliste |
| AWP | Anwenderprogramm |

| B | |
|---|---|
| BA | Betriebsart |
| BAG | Betriebsartengruppe |
| BCD | Binary Coded Decimals: Im Binärcode verschlüsselte Dezimalzahlen |
| BERO | Berührungsloser Näherungsschalter |
| BI | Binector Input |
| BICO | Binector Connector |
| BIN | Binary Files: Binärdateien |
| BIOS | Basic Input Output System |
| BKS | Basiskoordinatensystem |
| BO | Binector Output |
| BTSS | Bedientafelschnittstelle |

| C | |
|---|---|
| CAD | Computer-Aided Design |
| CAM | Computer-Aided Manufacturing |
| CC | Compile Cycle: Compile-Zyklen |
| CEC | Cross Error Compensation |
| CI | Connector Input |
| CNC | Computerized Numerical Control: Computerunterstützte numerische Steuerung |

| C | |
|---|---|
| CO | Connector Output |
| CoL | Certificate of License |
| COM | Communication |
| CPA | Compiler Projecting Data: Projektierdaten des Compilers |
| CRT | Cathode Ray Tube: Bildröhre |
| CSB | Central Service Board: PLC-Baugruppe |
| CU | Control Unit |
| CP | Communication Processor |
| CPU | Central Processing Unit: Zentrale Rechnereinheit |
| CR | Carriage Return |
| CST | Configured Stop: Konfigurierter Halt |
| CTS | Clear To Send: Meldung der Sendebereitschaft bei seriellen Daten-Schnittstellen |
| CUTCOM | Cutter Radius Compensation: Werkzeugradiuskorrektur |

| D | |
|---|---|
| DAU | Digital-Analog-Umwandler |
| DB | Datenbaustein (PLC) |
| DBB | Datenbaustein-Byte (PLC) |
| DBD | Datenbaustein-Doppelwort (PLC) |
| DBW | Datenbaustein-Wort (PLC) |
| DBX | Datenbaustein-Bit (PLC) |
| DDE | Dynamic Data Exchange |
| DDS | Drive Data Set: Antriebsdatensatz |
| DIN | Deutsche Industrie Norm |
| DIO | Data Input/Output: Datenübertragungs-Anzeige |
| DIR | Directory: Verzeichnis |
| DLL | Dynamic Link Library |
| DO | Drive Object |
| DPM | Dual Port Memory |
| DPR | Dual Port RAM |
| DRAM | Dynamischer Speicher (ungepuffert) |
| DRF | Differential Resolver Function: Differential-Drehmelder-Funktion (Handrad) |
| DRIVE-CLiQ | Drive Component Link with IQ |
| DRY | Dry Run: Probelaufvorschub |
| DSB | Decoding Single Block: Dekodierungseinzelsatz |
| DSC | Dynamic Servo Control / Dynamic Stiffness Control |
| DW | Datenwort |
| DWORD | Doppelwort (aktuell 32 Bit) |

| **E** | |
|---|---|
| E | Eingang |
| EES | Execution from External Storage |
| E/A | Ein-/Ausgabe |
| ENC | Encoder: Istwertgeber |
| EFP | Einfach Peripheriemodul (PLC–E/A–Baugruppe) |
| EGB | Elektronisch gefährdete Baugruppen/Bauelemente |
| EMV | Elektromagnetische Verträglichkeit |
| EN | Europäische Norm |
| ENC | Encoder: Istwertgeber |
| EnDat | Geberschnittstelle |
| EPROM | Erasable Programmable Read Only Memory: Löschbarer, elektrisch programmierbarer nur Lesespeicher |
| ePS Network Services | Dienste zur internetgestützten Maschinen-Fernwartung |
| EQN | Typbezeichnung eines Absolutwertgebers mit 2048 Sinussignalen/Umdrehung |
| ES | Engineering System |
| ESR | Erweitertes Stillsetzen und Rückziehen |
| ETC | ETC–Taste ">"; Erweiterung der Softkeyleiste im gleichen Menü |

| **F** | |
|---|---|
| FB | Funktionsbaustein (PLC) |
| FC | Function Call: Funktionsbaustein (PLC) |
| FEPROM | Flash–EPROM: Les– und schreibbarer Speicher |
| FIFO | First In First Out: Speicher, der ohne Adressangabe arbeitet und dessen Daten in derselben Reihenfolge gelesen werden, in der sie gespeichert wurden |
| FIPO | Feininterpolator |
| FPU | Floating Point Unit: Gleitpunkteinheit |
| FRK | Fräsradiuskorrektur |
| FST | Feed Stop: Vorschub Halt |
| FUP | Funktionsplan (Programmiermethode für PLC) |
| FW | Firmware |

| **G** | |
|---|---|
| GC | Global Control (PROFIBUS: Broadcast-Telegramm) |
| GDIR | Globaler Teileprogrammspeicher |
| GEO | Geometrie, z.B. Geometrieachse |
| GIA | Gear Interpolation Data: Getriebeinterpolationsdaten |
| GND | Signal Ground |
| GP | Grundprogramm (PLC) |
| GS | Getriebestufe |
| GSD | Gerätestammdatei zur Beschreibung eines PROFIBUS Slaves |

| **G** | |
|---|---|
| GSDML | Generic Station Description Markup Language: XML-basierte Beschreibungs-sprache zur Erstellung einer GSD-Datei |
| GUD | Global User Data: Globale Anwenderdaten |

| **H** | |
|---|---|
| HEX | Kurzbezeichnung für hexadezimale Zahl |
| HiFu | Hilfsfunktion |
| HLA | Hydraulischer Linearantrieb |
| HMI | Human Machine Interface: SINUMERIK-Bedienoberfläche |
| HSA | Hauptspindelantrieb |
| HW | Hardware |

| **I** | |
|---|---|
| IBN | Inbetriebnahme |
| IKA | Interpolatorische Kompensation |
| IM | Interface-Modul: Anschaltungsbaugruppe |
| IMR | Interface-Modul Receive: Anschaltungsbaugruppe für Empfangsbetrieb |
| IMS | Interface-Modul Send: Anschaltungsbaugruppe für Sendebetrieb |
| INC | Increment: Schrittmaß |
| INI | Initializing Data: Initialisierungsdaten |
| IPO | Interpolator |
| ISA | International Standard Architecture |
| ISO | International Standard Organization |

| **J** | |
|---|---|
| JOG | Jogging: Einrichtbetrieb |

| **K** | |
|---|---|
| $K_V$ | Verstärkungsfaktor des Regelkreises |
| $K_P$ | Proportionalverstärkung |
| $K_Ü$ | Übersetzungsverhältnis |
| KOP | Kontaktplan (Programmiermethode für PLC) |

| **L** | |
|---|---|
| LAI | Logic Machine Axis Image: Logisches Maschinenachsen-Abbild |
| LAN | Local Area Network |
| LCD | Liquid-Crystal Display: Flüssigkristallanzeige |
| LED | Light Emitting Diode: Leuchtdiode |
| LF | Line Feed |

| **L** | |
|---|---|
| LMS | Lagemesssystem |
| LR | Lageregler |
| LSB | Least Significant Bit: Niederwertigstes Bit |
| LUD | Local User Data: Anwenderdaten (lokal) |

| **M** | |
|---|---|
| MAC | Media Access Control |
| MAIN | Main program: Hauptprogramm (OB1, PLC) |
| MB | Megabyte |
| MCI | Motion Control Interface |
| MCIS | Motion–Control–Information–System |
| MCP | Machine Control Panel: Maschinensteuertafel |
| MD | Maschinendatum bzw. Maschinendaten |
| MDA | Manual Data Automatic: Handeingabe |
| MDS | Motor Data Set: Motordatensatz |
| MELDW | Meldungswort |
| MKS | Maschinenkoordinatensystem |
| MM | Motor Module |
| MPF | Main Program File: Hauptprogramm (NC) |
| MSTT | Maschinensteuertafel |

| **N** | |
|---|---|
| NC | Numerical Control: Numerische Steuerung mit Satzaufbereitung, Verfahrbereich usw. |
| NCU | Numerical Control Unit: Hardware-Einheit des NC |
| NRK | Bezeichnung des Betriebssystems des NC |
| NST | Nahtstellensignal |
| NURBS | Non-Uniform Rational B-Spline |
| NV | Nullpunktverschiebung |
| NX | Numerical Extension: Achserweiterungsbaugruppe |

| **O** | |
|---|---|
| OB | Organisationsbaustein in der PLC |
| OEM | Original Equipment Manufacturer |
| OP | Operation Panel: Bedieneinrichtung |
| OPI | Operation Panel Interface: Bedientafel-Anschaltung |
| OPT | Options: Optionen |
| OLP | Optical Link Plug: Busstecker für Lichtleiter |
| OSI | Open Systems Interconnection: Normung für Rechnerkommunikation |

| **P** | |
|---|---|
| PAA | Prozessabbild der Ausgänge |
| PAE | Prozessabbild der Eingänge |
| PC | Personal Computer |
| PCIN | Name der SW für den Datenaustausch mit der Steuerung |
| PCMCIA | Personal Computer Memory Card International Association: Speichersteckkarten-Normierung |
| PCU | PC Unit: PC-Box (Rechnereinheit) |
| PG | Programmiergerät |
| PKE | Parameterkennung: Teil eines PKW |
| PKW | Parameterkennung: Wert (Parametrierteil eines PPO) |
| PLC | Programmable Logic Control: Anpass-Steuerung |
| PN | PROFINET |
| PNO | PROFIBUS-Nutzerorganisation |
| PO | POWER ON |
| POE | Programmorganisationseinheit |
| POS | Position/Positionieren |
| POSMO A | Positioning Motor Actuator: Positioniermotor |
| POSMO CA | Positioning Motor Compact AC: Komplette Antriebseinheit mit integrierter Leistungs– und Reglungsbaugruppe sowie Positioniereinheit und Programmspeicher; Wechselstrom–Einspeisung |
| POSMO CD | Positioning Motor Compact DC: wie CA, jedoch Gleichstromeinspeisung |
| POSMO SI | Positioning Motor Servo Integrated: Positioniermotor; Gleichstromeinspeisung |
| PPO | Parameter Prozessdaten Objekt ; Zyklisches Datentelegramm bei der Übertragung mit PROFIBUS–DP und Profil "Drehzahlveränderbare Antriebe" |
| PPU | Panel Processing Unit (zentrale Hardware einer Panel-basierten CNC-Steuerung z.B. SINUMERIK 828D) |
| PROFIBUS | Process Field Bus: Serieller Datenbus |
| PRT | Programmtest |
| PSW | Programmsteuerwort |
| PTP | Point to Point: Punkt zu Punkt |
| PUD | Program Global User Data: Programmglobale Anwendervariable |
| PZD | Prozessdaten: Prozessdatenteil eines PPO |

| **Q** | |
|---|---|
| QFK | Quadrantenfehler Kompensation |

| **R** | |
|---|---|
| RAM | Random Access Memory: Schreib-/Lese-Speicher |
| REF | Funktion Referenzpunkt anfahren |
| REPOS | Funktion Repositionieren |
| RISC | Reduced Instruction Set Computer: Prozessortyp mit kleinem Befehlssatz und schnellem Befehlsdurchsatz |

| R | |
|---|---|
| ROV | Rapid Override: Eingangskorrektur |
| RP | R-Parameter, Rechenparameter, vordefinierte Anwendervariable |
| RPA | R-Parameter Active: Speicherbereich in NC für R-Parameternummern |
| RPY | Roll Pitch Yaw: Drehungsart eines Koordinatensystems |
| RTLI | Rapid Traverse Linear Interpolation: Lineare Interpolation bei Eilgangbewegung |
| RTS | Request To Send: Sendeteil einschalten, Steuersignal von seriellen Daten-Schnittstellen |
| RTCP | Real Time Control Protocol |

| S | |
|---|---|
| SA | Synchronaktion |
| SBC | Safe Brake Control: Sichere Bremsansteuerung |
| SBL | Single Block: Einzelsatz |
| SBR | Subroutine: Unterprogramm (PLC) |
| SD | Settingdatum bzw. Settingdaten |
| SDB | System Datenbaustein |
| SEA | Setting Data Active: Kennzeichnung (Dateityp) für Settingdaten |
| SERUPRO | Search-Run by Program Test: Satzsuchlauf via Programmtest |
| SFB | System Funktionsbaustein |
| SFC | System Function Call |
| SH | Sicherer Halt |
| SIM | Single in Line Module |
| SK | Softkey |
| SKP | Skip: Funktion zum Ausblenden eines Teileprogrammsatzes |
| SLM | Synchroner Linearmotor |
| SM | Schrittmotor |
| SMC | Sensor Module Cabinet Mounted |
| SME | Sensor Module Externally Mounted |
| SMI | Sensor Module Integrated |
| SPF | Sub Program File: Unterprogramm (NC) |
| SPS | Speicherprogrammierbare Steuerung = PLC |
| SRAM | Statischer Speicher (gepuffert) |
| SRK | Schneidenradiuskorrektur |
| SRM | Synchron rotatorischer Motor |
| SSFK | Spindelsteigungsfehlerkompensation |
| SSI | Serial Synchron Interface: Serielle synchrone Schnittstelle |
| SSL | Satzsuchlauf |
| STW | Steuerwort |
| SUG | Scheibenumfangsgeschwindigkeit |
| SW | Software |
| SYF | System Files: Systemdateien |
| SYNACT | Synchronized Action: Synchronaktion |

| T | |
|---|---|
| TB | Terminal Board (SINAMICS) |
| TCP | Tool Center Point: Werkzeugspitze |
| TCP/IP | Transport Control Protocol / Internet Protocol |
| TCU | Thin Client Unit |
| TEA | Testing Data Active: Kennung für Maschinendaten |
| TIA | Totally Integrated Automation |
| TM | Terminal Module (SINAMICS) |
| TO | Tool Offset: Werkzeugkorrektur |
| TOA | Tool Offset Active: Kennzeichnung (Dateityp) für Werkzeugkorrekturen |
| TRANSMIT | Transform Milling Into Turning: Koordinatentransformation für Fräsbearbeitungen an einer Drehmaschine |
| TTL | Transistor–Transistor–Logik (Schnittstellen–Typ) |
| TZ | Technologiezyklus |

| U | |
|---|---|
| UFR | User Frame: Nullpunktverschiebung |
| UP | Unterprogramm |
| USB | Universal Serial Bus |
| USV | Unterbrechungsfreie Stromversorgung |

| V | |
|---|---|
| VDI | Interne Kommunikationsschnittstelle zwischen NC und PLC |
| VDI | Verein Deutscher Ingenieure |
| VDE | Verband Deutscher Elektrotechniker |
| VI | Voltage Input |
| VO | Voltage Output |
| VSA | Vorschubantrieb |

| W | |
|---|---|
| WAB | Funktion Weiches An- und Abfahren |
| WKS | Werkstückkoordinatensystem |
| WKZ | Werkzeug |
| WLK | Werkzeuglängenkorrektur |
| WOP | Werkstatt-orientierte Programmierung |
| WPD | Work Piece Directory: Werkstückverzeichnis |
| WRK | Werkzeug-Radius-Korrektur |
| WZ | Werkzeug |
| WZK | Werkzeugkorrektur |
| WZV | Werkzeugverwaltung |
| WZW | Werkzeugwechsel |

| X | |
|---|---|
| XML | Extensible Markup Language |

| Z | |
|---|---|
| ZOA | Zero Offset Active: Kennung für Nullpunktverschiebungen |
| ZSW | Zustandswort (des Antriebs) |

# A.2 Dokumentationsübersicht

Eine umfangreiche Dokumentation zu den Funktionen von SINUMERIK 840D sl ab der Version 4.8 SP4 finden Sie unter Dokumentationsübersicht 840D sl (https://support.industry.siemens.com/cs/ww/de/view/109766213).

Sie haben die Möglichkeit, die Dokumente anzuzeigen oder im PDF- und HTML5-Format herunterzuladen.

Die Dokumentation ist in folgende Kategorien unterteilt:

- Anwender: Bedienen
- Anwender: Programmieren
- Hersteller/Service: Projektieren
- Hersteller/Service: Inbetriebnahme
- Hersteller/Service: Funktionen
- Safety Integrated
- SINUMERIK Integrate / MindApp
- Information und Training

# Index

## $


$A_PROBE, 623, 630
$A_PROBE_LIMITED, 630
$AA_ACC, 131
$AA_ATOL, 883
$AA_COUP_ACT
  bei axialer Leitwertkopplung, 933
  beim Mitschleppen, 911
$AA_ESR_ENABLE, 993
$AA_FGREF, 115
$AA_FGROUP, 115
$AA_G0MODE, 184
$AA_LEAD_SP, 933
$AA_LEAD_SV, 933
$AA_MM, 624
$AA_MM1...4, 631
$AA_MW, 624
$AA_MW1...4, 631
$AC_ACT_PROG_NET_TIME, 1001
$AC_ACTUAL_PARTS, 1004
$AC_AXCTSWA, 899
$AC_AXCTSWE, 900
$AC_CTOL, 883
$AC_CTOL_G0_ABS, 188
$AC_CUT_INV, 817
$AC_CUTMOD, 816
$AC_CUTMOD_ANG, 816
$AC_CUTMODK, 816
$AC_CUTTING_TIME, 1000
$AC_CYCLE_TIME, 1000
$AC_DELAYFST, 866
$AC_ESR_TRIGGER, 993
$AC_F_TYPE, 145
$AC_FGROUP_MASK, 115
$AC_FZ, 145
$AC_MEA, 623, 630
$AC_OLD_PROG_NET_TIME, 1001
$AC_OLD_PROG_NET_TIME_COUNT, 1001
$AC_OPERATING_TIME, 1000
$AC_OTOL, 883
$AC_OTOL_ G0_ABS, 188
$AC_PROG_NET_TIME_TRIGGER, 1001
$AC_REPOS_PATH_MODE, 875
$AC_REQUIRED_PARTS, 1003
$AC_S_TYPE, 99
$AC_SMAXVELO, 879
$AC_SMAXVELO_INFO, 879
$AC_SPECIAL_PARTS, 1004
$AC_STOLF, 188
$AC_SVC, 99
$AC_TOFF, 89
$AC_TOFFCR, 89
$AC_TOFFL, 89
$AC_TOFFR, 89
$AC_TOTAL_PARTS, 1003
$AC_TRAFO_CORR_ELEM_P, 747
$AC_TRAFO_CORR_ELEM_T, 747
$AC_TRAFO_ORIAX_LOC, 748
$AN_AXCTAS, 900
$AN_AXCTSWA, 899
$AN_ESR_TRIGGER, 993
$AN_LANGUAGE_ON_HMI, 1326
$AN_POWERON_TIME, 1000
$AN_SETUP_TIME, 1000
$NT_CLOSE_CHAIN_T, 748
$NT_CNTRL, 747
$NT_CORR_ELEM_P, 747
$NT_CORR_ELEM_T, 747
$NT_NAME, 740
$NT_ROT_AX_NAME, 812
$NT_TRAFO_INDEX, 740
$P_ACTBFRAME, 660
$P_AD, 816
$P_AEP, 284
$P_APDV, 284
$P_APR, 284
$P_BFRAME, 659
$P_CHBFRAME, 659
$P_CHBFRMASK, 660
$P_CTOL, 884
$P_CTOL_ G0_ABS, 188
$P_CUT_INV, 817
$P_CUTMOD, 816
$P_CUTMOD_ANG, 816
$P_CUTMOD_ERR, 818
$P_CUTMODK, 816
$P_DELAYFST, 866
$P_F_TYPE, 146
$P_FGROUP_MASK, 116
$P_FZ, 146
$P_GWPS, 106
$P_IFRAME, 661
$P_IS_EES_PATH, 570
$P_NCBFRAME, 659
$P_NCBFRMASK, 660
$P_ORI_DIFF, 807

$P_ORI_POS, 807
$P_ORI_SOL, 808
$P_ORI_STAT, 811
$P_OTOL, 884
$P_OTOL_ G0_ABS, 188
$P_PATH, 569
$P_PFRAME, 661
$P_PROG, 569
$P_PROGPATH, 569
$P_S_TYPE, 100
$P_SIM, 633
$P_STACK, 569
$P_STOLF, 188
$P_SUBPAR, 503
$P_SVC, 100
$P_TOFF, 89
$P_TOFFCR, 89
$P_TOFFL, 89
$P_TOFFR, 89
$P_TOOLENV, 826
$P_TOOLENVN, 826
$P_WORKAREA_CS_COORD_SYSTEM, 359
$P_WORKAREA_CS_LIMIT_MINUS, 360
$P_WORKAREA_CS_LIMIT_PLUS, 359
$P_WORKAREA_CS_MINUS_ENABLE, 359
$P_WORKAREA_CS_PLUS_ENABLE, 359
$PA_ATOL, 884
$PA_FGREF, 115
$PA_FGROUP, 116
$SA_LEAD_TYPE, 933
$SC_CONTPREC, 860
$SC_MINFEED, 861
$SC_PA_ACTIV_IMMED, 590
$SN_PA_ACTIV_IMMED, 590
$TC_CARR_CORR_ELEM, 799
$TC_CARR1...14, 789
$TC_CARR18...23, 789
$TC_CARR18[m], 793
$TC_DP1 ... 25, 749
$TC_ECPxy, 753
$TC_SCPxy, 753
$TC_TP_MAX_VELO, 96

*****

* (Rechenfunktion), 446

**/**

/ (Rechenfunktion), 446

**+**

+ (Rechenfunktion), 446

**<**

< (Vergleichsoperator), 448
<< (Verkettungsoperator), 455
<= ( Vergleichsoperator), 448
<> (Vergleichsoperator), 448

**=**

== (Vergleichsoperator), 448

**>**

> (Vergleichsoperator), 448
>= (Vergleichsoperator), 448

## 0

0-Zeichen, 453

## A

Abrichterkoordinatensystem setzen - CYCLE435
    extern programmieren, 1107
ABS, 446
Absolutmaß, 30
Abspanen
    unterstützende Funktionen, 1027
Abspanen - CYCLE951
    extern programmieren, 1140
Abspanen / Abspanen Rest / Stechen / Stechen Rest / Stechdrehen / Stechdrehen Rest - CYCLE952
    extern programmieren, 1143
Abstich - CYCLE92
    extern programmieren, 1094
AC, 153
ACC, 130
ACCLIMA, 856
Achs
    -Container, 380
    -tausch, 886
    -typen, 374
Achsen
    Bahn-, 377

Geometrie-, 374
Haupt-, 374
Kanal-, 377
Kommando-, 379
Lead-Linkachse, 381
Link-, 379
Maschinen-, 376
Mitschlepp-, 910
PLC-, 379
Positionier-, 377
Synchron-, 378
Zusatz-, 376
ACN, 160
ACOS, 446
ACP, 160
ACTBLOCNO, 514
ACTFRAME, 639
ADIS, 298
ADISPOS, 298
ADISPOSA, 634
Adressbuchstaben, 1255
Adresse
Wertzuweisung, 48
Adressen, 384
Adressierung, 563
Alarme
im NC-Programm setzen, 1018
ALF
für Schnellabheben von der Kontur, 554
für Schnellrückzug während Gewindeschneiden, 234
AMIRROR, 331
An Beladung anpassen - CYCLE782
extern programmieren, 1109
AND, 448
Anfahrpunkt/-winkel, 266
Anfang Programmblock - GROUP_BEGIN, 1171
ANG, 214
ANG1, 214
ANG2, 214
Anschlagpunkt, 34
Anweisung, 44
Anwender-XML, 1009
AP, 178
APR, 417
APRB, 417
APRP, 417
APW, 417
APWB, 417
APWP, 417
AR
Kreisprogrammierung, 197
Arbeitsebene, 33
Arbeitsfeldbegrenzung
im BKS, 355
Arbeitsspeicher, 570
AROT, 319
AROTS, 325
Array, 422
AS, 494
ASCALE, 328
ASIN, 446
A-Spline, 608
ASPLINE, 602
Asynchrones Pendeln, 968
ATAN2, 446
ATOL, 880
ATRANS, 313
Aufrunden, 451
Ausblendebenen, 50
Ausblendsätze, 49
Ausdrehen - CYCLE86
extern programmieren, 1093
Ausgabe
auf externes Gerät/Datei, 1014
Automatische Wegaufteilung, 985
Automatischer Unterbrechungszeiger, 868
AV, 940
AX, 892
AXCTSWE, 898
AXCTSWEC, 898
AXCTSWED, 898
Axiale Leitwertkopplung, 929
AXIS, 401
AXNAME, 454
AXSTRING, 892
AXTOCHAN, 890
AXTOSPI, 892


## B


B_AND, 448
B_NOT, 448
B_OR, 448
B_XOR, 448
Bahnachsen, 377
Bahnfräsen - CYCLE72
extern programmieren, 1070
Bahnrelative Werkzeugorientierung, 697
Bahnsteuerbetrieb, 298
Ballige Gewinde, 238
Basis-Koordinatensystem, 38
Basis-Nullpunktsystem, 39
Basisverschiebung, 40

BAUTO, 602
Bearbeitungszeit, 1000
Befehl, 44
Beliebige Positionen - CYCLE802
extern programmieren, 1117
Beschleunigungsmodus, 854
Bewegungsendekriterium
programmierbar, 634
Bezeichner, 43
Bezugspunkte, 34
Bezugsradius, 114
BFRAME, 639
Binär-Konstante, 389
BKS, 38
BLOCK, 539
BLSYNC, 549
BNAT, 602
BNS, 39
Bohren - CYCLE82
extern programmieren, 1083
Bohrer, 74
Bohrgewinde fräsen - CYCLE78
extern programmieren, 1078
BOOL, 401
BOUND, 429
BRISK, 854
BRISKA, 854
B-Spline, 609
BSPLINE, 602
BTAN, 602

## C

CAC, 601
CACN, 601
CACP, 601
CADAPTOF, 902
CADAPTON, 902
CALL, 538
CALLPATH, 541
CASE, 473
case-insensitiv, 562
CDC, 601
CDOF, 287
CDOF2, 287
CDON, 287
CFC, 135
CFIN, 135
CFINE, 648
CFTCP, 135
CHAN, 401
CHANDATA, 570
CHAR, 401
CHF, 248
CHKDNO, 786
CHR, 248
CIC, 601
CIP, 200
CLEARM, 488
CLRINT, 551
COARSE, 940
COARSEA, 634
COLLPAIR, 735
COMPCAD, 613
COMPCURV, 613
COMPLETE, 570
COMPOF, 613
COMPON, 613
COMPSURF, 613
CONTDCON, 1033
CONTPRON, 1027
CORROF, 340
CORRTC, 796
CORRTRAFO, 741
COS, 446
COUPDEF, 940
COUPDEL, 940
COUPOF, 940
COUPOFS, 940
COUPON, 940
COUPONC, 940
COUPRES, 940
CP, 718
CPBC, 952
CPDEF, 951
CPDEL, 951
CPFMOF, 954
CPFMON, 954
CPFMSON, 953
CPFPOS + CPOF, 954
CPFPOS + CPON, 952
CPFRS, 951
CPLA, 951
CPLCTID, 951
CPLDEF, 951
CPLDEL, 951
CPLDEN, 951
CPLINSC, 956
CPLINTR, 956
CPLNUM, 951
CPLOF, 951
CPLON, 951
CPLOUTSC, 956
CPLOUTTR, 956

CPLPOS, 952
CPLSETVAL, 951
CPMALARM, 957
CPMBRAKE, 957
CPMPRT, 956
CPMRESET, 955
CPMSTART, 956
CPMVDI, 957
CPOF, 951
CPON, 951
CPRECOF, 860
CPRECON, 860
CPROT, 587
CPROTDEF, 584
CPSETTYPE, 957
CPSYNCOP, 956
CPSYNCOP2, 956
CPSYNCOV, 957
CPSYNFIP, 956
CPSYNFIP2, 957
CPSYNFIV, 957
CR, 195
CROTS, 325
C-Spline, 610
CSPLINE, 602
CT, 203
CTAB, 923
CTABDEF, 913
CTABDEL, 919
CTABEND, 913
CTABEXISTS, 919
CTABFNO, 927
CTABFPOL, 927
CTABFSEG, 927
CTABID, 921
CTABINV, 923
CTABISLOCK, 921
CTABLOCK, 920
CTABMEMTYP, 921
CTABMPOL, 927
CTABMSEG, 927
CTABNO, 927
CTABNOMEM, 927
CTABPERIOD, 921
CTABPOL, 927
CTABPOLID, 927
CTABSEG, 927
CTABSEGID, 927
CTABSEV, 923
CTABSSV, 923
CTABTEP, 923
CTABTEV, 923
CTABTMAX, 923
CTABTMIN, 923
CTABTSP, 923
CTABTSV, 923
CTABUNLOCK, 920
CTOL, 880
CTOLG0, 185
CTRANS, 648
CUT2D, 289
CUT2DD, 289
CUT2DF, 289
CUT2DFD, 289
CUT3DC, 765
CUT3DCC, 775
CUT3DCCD, 775
CUT3DCD, 765
CUT3DF, 769
CUT3DFD, 769
CUT3DFF, 769
CUT3DFS, 769
CUTCONOF, 291
CUTCONON, 291
CUTMOD, 813
CUTMODK, 813
CYCLE4071
  extern programmieren, 1149
CYCLE4072
  extern programmieren, 1151
CYCLE4073
  extern programmieren, 1155
CYCLE4074
  extern programmieren, 1156
CYCLE4075
  extern programmieren, 1159
CYCLE4077
  extern programmieren, 1162
CYCLE4078
  extern programmieren, 1166
CYCLE4079
  extern programmieren, 1168
CYCLE435 - Abrichterkoordinatensystem setzen
  extern programmieren, 1107
CYCLE495 - Profilieren
  extern programmieren, 1107
CYCLE60 - Gravur
  extern programmieren, 1059
CYCLE61 - Planfräsen
  extern programmieren, 1062
CYCLE62 - Konturaufruf
  extern programmieren, 1064

CYCLE63 - Konturtasche fräsen / Konturtasche Restmaterial / Konturzapfen fräsen / Konturzapfen Restmaterial
extern programmieren, 1064
CYCLE64 - Konturtasche vorbohren
extern programmieren, 1067
CYCLE70 - Gewindefräsen
extern programmieren, 1069
CYCLE72 - Bahnfräsen
extern programmieren, 1070
CYCLE76 - Rechteckzapfen
extern programmieren, 1073
CYCLE77 - Kreiszapfen
extern programmieren, 1076
CYCLE78 - Bohrgewinde fräsen
extern programmieren, 1078
CYCLE782 - An Beladung anpassen
extern programmieren, 1109
CYCLE79 - Mehrkant
extern programmieren, 1080
CYCLE800 - Schwenken Ebene / Schwenken Werkzeug / Ausrichten Werkzeug
extern programmieren, 1112
CYCLE801 - Positionsmuster Gitter oder Rahmen
extern programmieren, 1115
CYCLE802 - Beliebige Positionen
extern programmieren, 1117
CYCLE81 - Zentrieren
extern programmieren, 1082
CYCLE82 - Bohren
extern programmieren, 1083
CYCLE83 - Tieflochbohren 1
extern programmieren, 1086
CYCLE830 - Tieflochbohren 2
extern programmieren, 1120
CYCLE832 - High Speed Settings
extern programmieren, 1126
CYCLE84 - Gewindebohren ohne Ausgleichsfutter
extern programmieren, 1089
CYCLE840 - Gewindebohren mit Ausgleichsfutter
extern programmieren, 1130
CYCLE85 - Reiben
extern programmieren, 1092
CYCLE86 - Ausdrehen
extern programmieren, 1093
CYCLE899 - Offene Nut
extern programmieren, 1132
CYCLE92 - Abstich
extern programmieren, 1094
CYCLE930 - Einstich
extern programmieren, 1135
CYCLE940 - Freistich Form E und F / Freistich Gewinde
extern programmieren, 1138
CYCLE95 - Konturabspanen
extern programmieren, 1096
CYCLE951 - Abspanen
extern programmieren, 1140
CYCLE952 - Abspanen / Abspanen Rest / Stechen / Stechen Rest / Stechdrehen / Stechdrehen Rest
extern programmieren, 1143
CYCLE98 - Gewindekette
extern programmieren, 1098
CYCLE99 - Gewindedrehen
extern programmieren, 1102

## D

D..., 81
D0, 81
DAC, 167
Datei
-informationen, 582
Dateiname, 567
DC, 160
DEF, 401
DEFAULT, 473
DEFINE ... AS, 494
DELAYFSTOF, 864
DELAYFSTON, 864
DELDL, 753
DELETE, 578
DELOBJ, 731
DELTOOLENV, 824
Dezimal-Konstante, 388
DIACYCOFA, 167
DIAM90, 165
DIAM90A, 167
DIAMCHAN, 167
DIAMCHANA, 167
DIAMCYCOF, 165
DIAMOF, 165
DIAMOFA, 167
DIAMON, 165
DIAMONA, 167
DIC, 167
DILF, 234
DIN Unterprogrammname, 567
DIN 66217, 36
DISABLE, 551
DISC, 270
DISCL, 273
DISPLOF, 514

DISPLON, 514
DISPR, 869
DISR, 273
DISRP, 273
DITE, 232
DITS, 232
DIV, 446
DL, 752
D-Nummer
frei vergeben, 786
D-Nummern
prüfen, 786
umbenennen, 787
DO, 967
Drehachsen
Abstandsvektoren, 789
Richtungsvektoren, 789
Verdrehwinkel, 789
Drehung
des Orientierungsvektors, 695
Programmierbare, 319
Drehwerkzeuge, 77
Drehzahlsynchronität, 940
Drei-Finger-Regel, 36
DRFOF, 342
DRIVE, 854
DRIVEA, 854
Durchmesser-Programmierung, 165
DV, 940
DYNFINISH, 858
DYNNORM, 858
DYNPOS, 858
DYNPREC, 858
DYNROUGH, 858
DYNSEMIFIN, 858

## E

Easy XML, 1009
EAUTO, 602
Eckenverzögerung an allen Ecken, 634
Eckenverzögerung an Innenecken, 634
EES, 561
EES-Notation, 563
EG
Elektronisches Getriebe, 934
EGDEF, 934
EGDEL, 939
EGOFC, 938
EGOFS, 938
EGON, 935
EGONSYN, 935
EGONSYNE, 935
Einrichtewert, 753
Einstich - CYCLE930
extern programmieren, 1135
Eintauchtiefe, 768
Einzelsatz
-unterdrückung, 509
Elektronisches Getriebe, 934
ELSE, 482, 967
ENABLE, 551
ENAT, 602
Ende Einfahrzusatz - GROUP_ADDEND
extern programmieren, 1172
Ende Programmblock - GROUP_END
extern programmieren, 1171
ENDFOR, 485
ENDIF, 482
ENDLABEL, 475
ENDLOOP, 484
Endlosschleife, 484
ENDWHILE, 486
ENS, 40
Erweiterte Adressschreibweise, 385
ESR, 992
ESRR, 998
ESRS, 997
ETAN, 602
Eulerwinkel, 680
EVERY, 967
Evolvente, 209
EXECSTRING, 444
EXECTAB, 1038
EXECUTE, 1040
EXP, 446
EXTCALL
für SINUMERIK 840D sl, 543
EXTCLOSE, 1014
EXTERN, 532
extern programmieren, 1171
Externe Nullpunktverschiebung, 649
EXTOPEN, 1014

## F

F...
bei Geradeninterpolation, 189
bei Gewindeschneiden G34 G35, 231
bei Vorschub, 108
FA, 126
FAD, 273
Fahrbefehl, 173
FALSE, 401

Fase, 248
FB, 140
FCTDEF, 761
FCUB, 849
FD, 131
FDA, 131
Feld
-definition, 422
-element, 423
Feldindex, 425
FENDNORM, 634
Festanschlag, 366
Festpunkt
anfahren, 362
FFWOF, 859
FFWON, 859
FGREF, 108
FGROUP, 108
FIFOCTRL, 862
FILEDATE, 582
FILEINFO, 582
FILESIZE, 582
FILESTAT, 582
FILETIME, 582
FINE, 940
FINEA, 634
FL, 108
FLIN, 849
FMA, 137
FNORM, 849
Folgeachse
bei axialer Leitwertkopplung, 929
FOR, 485
FP, 362
FPO, 849
FPR, 126
FPRAOF, 126
FPRAON, 126
Frame
abwählen, 339
-Anweisungen, 310
aufrufen, 645
-Kettung, 663
-Skalierung, 328
-Spiegelung, 331
FRAME, 401
Framekomponente
FI, 644
MI, 644
RT, 644
SC, 644
TR, 644
Frames, 308
Frameketten, 647
globale, 657
Kanalspezifische, 658
System-, 659
zuweisen, 646
Framevariable
Aufruf von Koordinatentransformationen, 637
Vordefinierte Framevariable, 639
Werte zuweisen, 642
Fräserbearbeitungspunkt, 768
Fräserhilfspunkt, 768
Fräserspitze, 768
Fräswerkzeuge, 72
FRC, 248
FRCM, 248
Freistich Form E und F / Freistich Gewinde - CYCLE940
extern programmieren, 1138
FROM, 967
FTOCOF, 764
FTOCON, 764
Funktion
vordefinierte, 1312

## G

G0-Toleranzen, 185
G1, 189
G110, 176
G111, 176
G112, 176
G140, 273
G141, 273
G142, 273
G143, 273
G147, 273
G148, 273
G153
bei Frame abwählen, 339
bei Nullpunktverschiebung, 147
G17, 149
G18, 149
G19, 149
G2, 192
G247, 273
G248, 273
G25
Arbeitsfeldbegrenzung, 355
Spindeldrehzahlbegrenzung, 106

G26
Arbeitsfeldbegrenzung, 355
Spindeldrehzahlbegrenzung, 106
G290, 1024
G291, 1024
G3, 192
G33, 224
G335, 238
G336, 238
G34, 231
G340, 273
G341, 273
G347, 273
G348, 273
G35, 231
G4, 371
G40, 254
G41, 254
G42, 254
G450, 270
G451, 270
G460, 284
G461, 284
G462, 284
G5, 714
G500
bei Nullpunktverschiebung, 147
G505 ... G599, 147
G53
bei Frame abwählen, 339
bei Nullpunktverschiebung, 147
G54 ... G57, 147
G58, 317
G59, 317
G60, 296
G601, 296
G602, 296
G603, 296
G62, 634
G621, 634
G64, 298
G641, 298
G642, 298
G643, 298
G644, 298
G645, 298
G7, 714
G70, 162
G700, 162
G71, 162
G710, 162
G74, 361
G75, 362
G810 ... G819, 633
G820 ... G829, 633
G9, 296
G90, 153
G91, 155
G93, 108
G94, 108
G95, 108
G96, 100
G961, 100
G962, 100
G97, 100
G971, 100
G972, 100
G973, 100
G-Befehle, 1267
G-Code
indirekt programmieren, 441
Genauhalt, 296
GEOAX, 893
Geometrie
-achsen, 374
Geometrieachse
umschalten, 893
Geraden
-interpolation, 189
Geschwindigkeitskopplung, 943
GET, 886
GETACTTD, 788
GETD, 886
GETDNO, 787
GETTCOR, 826
GETTENV, 825
GETVARAP, 433
GETVARDFT, 435
GETVARDIM, 435
GETVARLIM, 434
GETVARPHU, 432
GETVARTYP, 437
Gewinde
-drehrichtung, 226
-kette, 225
-mehrgängig, 225
-schneiden G33, 224
-schneiden G34 G35, 231
-steigung, 231
Gewindebohren mit Ausgleichsfutter - CYCLE840
extern programmieren, 1130
Gewindebohren ohne Ausgleichsfutter - CYCLE84
extern programmieren, 1089

Gewindedrehen - CYCLE99
extern programmieren, 1102
Gewindefräsen - CYCLE70
extern programmieren, 1069
Gewindekette - CYCLE98
extern programmieren, 1098
GFRAME0 ... GFRAME100, 344
G-Gruppe
Technologie, 858
Glättung
des Orientierungsverlaufs, 708
Globaler Teileprogrammspeicher (GDIR), 561
GOTO, 470
GOTOB, 470
GOTOC, 470
GOTOF, 470
GOTOS, 469
GP, 442
Gravur - CYCLE60
extern programmieren, 1059
GROUP_ADDEND - Ende Einfahrzusatz
extern programmieren, 1172
GROUP_BEGIN - Anfang Programmblock
extern programmieren, 1171
GROUP_END - Ende Programmblock
extern programmieren, 1171
Grundstellung der Werkzeugorientierung
ORIRESET, 676
GUD, 402
GWPSOF, 105
GWPSON, 105

## H

Halt
am Zyklusende, 350
Programmierter, 350
Wahlweiser, 350
Haltesatz, 868
Handrad
-überlagerung, 131
Haupteintrag, 170
Helixinterpolation, 207
Hexadezimal-Konstante, 388
High Speed Settings - CYCLE832
extern programmieren, 1126
Hilfsfunktionsausgabe
Eigenschaften der Hilfsfunktionen, 345
im Bahnsteuerbetrieb, 348
Schnelle, 347
HOLES1 - Positionsmuster Reihe
extern programmieren, 1045
HOLES2 - Positionsmuster Kreis oder Teilkreis
extern programmieren, 1045

## I

I...
bei Gewindeschneiden G33, 224
bei Gewindeschneiden G34 G35, 231
bei Kreisinterpolation, 192
IC, 155
ID, 967
IDS, 967
IF, 482
IFRAME, 639
INDEX, 457
Indirekte Programmierung
von Adressen, 438
von G-Codes, 441
von Positionsattributen, 442
von Teileprogrammzeilen, 444
INICF, 401
INIPO, 401
INIRE, 401
INIT, 488
INITIAL, 570
INITIAL_INI, 570
Initialisierung
von Feldern, 423
Initialisierungsprogramm, 571
Inkrementalmaß, 32
INT, 401
INTEGER-Konstante, 388
Interner Vorlaufstopp, 373
Interpolation des Drehvektors, 695
Interruptroutine
Aus-/Einschalten, 551
Löschen, 551
Neu Zuordnen, 550
Programmierbare Verfahrrichtung, 554
Rückzugsbewegung, 554
Schnellabheben von der Kontur, 552
INTERSEC, 1036
INVCCW, 209
INVCW, 209
IPOBRKA, 634
IPOENDA, 634
IPOSTOP, 940
IPTRLOCK, 867
IPTRUNLOCK, 867
IR, 238
ISAXIS, 892
ISFILE, 581

ISNUMBER, 454
ISOCALL, 540
Istwertkopplung, 943
ISVAR, 430

## J

J...
  bei Gewindeschneiden G34 G35, 231
  bei Kreisinterpolation, 192
JERKLIM, 877
JERKLIMA, 856
JR, 238

## K

K...
  bei Gewindeschneiden G33, 224
  bei Gewindeschneiden G34 G35, 231
  bei Kreisinterpolation, 192
Kanal
  -achsen, 377
Kartesische Koordinaten, 28
Kartesisches PTP-Fahren, 717
Kegelgewinde, 230
Kennung
  für spezielle Zahlenwerte, 53
  für systemeigene Variablen, 53
  für Zeichenkette, 53
Kettenmaß, 32
Kettenmaßangabe, 155
Kinematik
  Aufgelöste, 793
Kinematiktyp, 793
Klemmoment
  -Festanschlag, 369
Kollisionsüberwachung, 287
Kommando
  -achsen, 379
Kommentare, 48
Konstante, 388
KONT, 263
KONTC, 263
Kontroll
  -strukturen, 481
KONTT, 263
Kontur
  anfahren/verlassen, 263
  -aufbereitung, 1027
  -codierung, 1033
  -element, 173
  -rechner, 215
  -tabelle, 1027
  wieder anfahren, 869
Konturabspanen - CYCLE95
  extern programmieren, 1096
Konturaufbereitung
  Fehlerrückmeldung, 1040
Konturaufruf - CYCLE62
  extern programmieren, 1064
Konturecke
  anfasen, 248
  verrunden, 248
Konturelement
  abfahren, 1038
Konturgenauigkeit
  programmierbare, 860
Konturtasche fräsen / Konturtasche Restmaterial / Konturzapfen fräsen / Konturzapfen Restmaterial – CYCLE63
  extern programmieren, 1064
Konturtasche vorbohren – CYCLE64
  extern programmieren, 1067
Konturzug-Programmierung, 214
Koordinaten
  Kartesische, 28
  Polar-, 29
  Zylinder-, 179
Koordinatensystem
  Basis-, 38
  Übersicht, 36
Koordinatentransformationen (Frames), 41
Koppelfaktor, 908
Kopplung
  Generische, 950
Kopplungsart, 943
Kopplungsstatus
  bei axialer Leitwertkopplung, 933
  beim Mitschleppen, 911
Korrektur
  -ebene, 291
  Werkzeuglängen-, 68
  Werkzeugradius-, 69
Korrekturspeicher, 749
KR, 238
Kreisdaten
  berechnen, 1039
Kreisinterpolation
  mit Zwischen- und Endpunkt, 200
  Schraubenlinieninterpolation, 207
Kreisnut - SLOT2
  extern programmieren, 1055

Kreisprogrammierung
Interpolationsarten, 192
mit Mittel- und Endpunkt, 192
mit Öffnungswinkel und Mittelpunkt, 197
mit Polarkoordinaten, 199
mit Radius und Endpunkt, 195
Kreistasche - POCKET4
extern programmieren, 1050
Kreiszapfen - CYCLE77
extern programmieren, 1076

## L

L..., 530
Label, 475
Lagesynchronität, 940
Lagesynchronität mit Winkelversatz, 940
Langloch - LONGHOLE
extern programmieren, 1057
Längsnut - SLOT1
extern programmieren, 1052
Laufwerksname, 564
Laufzeit
-verhalten von Kontrollstrukturen, 481
LEAD, 677
LEADOF, 929
LEADON, 929
Leitachse
bei axialer Leitwertkopplung, 929
Leitwertkopplung
Ist- und Sollwertkopplung, 932
Synchronisation Leit- und Folgeachse, 931
Leitwertsimulation, 933
LENTOAX, 845
LF, 47
LFOF, 234
LFON, 234
LFPOS, 234
LFTXT, 234
LFWP, 234
LIFTFAST, 552
LIMS, 100
LINE FEED, 47
Link
-Achsen, 379
Lead-Link-Achse, 381
-Variablen, 399
Linksgewinde, 226
LLI, 413
LN, 446
Lochstreifenformat, 44
Logische Operatoren, 448
LONGHOLE - Langloch
extern programmieren, 1057
LookAhead, 303
LOOP, 484
LUD, 402

## M

M..., 348
M0, 348
M1, 348
M17, 518
M19
bei Spindeln positionieren, 121
M-Funktionen, 348
M2, 348
M3, 91
M30, 518
M4, 91
M40, 348
M41, 348
M42, 348
M43, 348
M44, 348
M45, 348
M5, 91
M6, 66
M70, 121
Makro, 494
Maschinen
-achsen, 376
-Nullpunkt, 34
Maschinenkoordinatensystem, 36
MASLDEF, 964
MASLDEL, 964
MASLOF, 964
MASLOFS, 964
MASLON, 964
Maßangaben
für Rundachsen und Spindeln, 160
im Durchmesser, 165
im Radius, 165
Möglichkeiten, 153
Maßstabsfaktor, 328
Maßsystem, 162
Masterspindel, 376
MATCH, 457
MAXVAL, 429
MCALL, 535
MD10010, 488
MD10240, 163
MD10260, 162

MD10280, 488
MD10651, 239
MD10710, 234
MD15800, 399
MD18104, 823
MD18116, 824
MD18156, 399
MD20360, 831
MD24558, 833
MD24658, 833
MD9440, 83
MEAC, 624
MEAFRAME, 653
MEAS, 621
MEASA, 624
MEAW, 621
MEAWA, 624
Mehrkant - CYCLE79
extern programmieren, 1080
Meldungen, 352
Messzyklenparameter
CYCLE961, 1194
CYCLE971, 1210
CYCLE973, 1175
CYCLE974, 1177
CYCLE976, 1183
CYCLE977, 1190
CYCLE978, 1185
CYCLE979, 1196
CYCLE982, 1207
CYCLE994, 1180
CYCLE995, 1201
CYCLE996, 1203
CYCLE9960, 1205
CYCLE997, 1199
CYCLE998, 1188
M-Funktionen, 348
MINDEX, 457
MINVAL, 429
MIRROR, 331
Mitschleppachsen, 910
Mitschleppen, 908
Mitschleppverband, 908
MKS, 36
MMC, 1009
MOD, 446
Modal wirksam, 46
MODAXVAL, 892
MPF, 560
MSG, 352

## N

NAMETOINT, 734
NC-Hochsprache, 45
NCK, 401
NCK-Notation, 563
NC-Programm
erstellen, 51
NC-Programmierung
Zeichenvorrat, 52
Nenner-Polynom, 618
NEWCONF, 1005
Nibbeln, 980
NOC, 940
NORM, 263
NOT, 448
NPROT, 587
NPROTDEF, 584
Nullframe, 147
Nullpunkte
beim Drehen, 171
Nullpunktsystem
Basis-, 39
Einstellbares, 40
Nullpunktverschiebung
Einstellbare, 147
Externe, 649
Programmierbare, 313
NUMBER, 454
Numerische Erweiterung, 385
NUT, 688
Nutsäge, 80

## O

OEM-Adressen, 632
OEM-Funktionen, 632
OEMIPO1/2, 633
Offene Nut - CYCLE899
extern programmieren, 1132
OFFN, 254
Offset
Werkzeuglängen-, 85
Werkzeugradius-, 85
OMA1 ... OMA5, 632
Online-Werkzeuglängenkorrektur, 800
OR, 448
ORIAXES, 686
ORIC, 780
ORICONCCW, 688

ORICONCW, 688
ORICONIO, 688
ORICONTO, 688
ORICURVE, 691
ORID, 780
Orientierbare Werkzeugträger, 789
Orientierungsachsen, 686
Orientierungsprogrammierung, 686
Orientierungstransformation TRAORI
  Generische 5/6-Achs Transformation, 667
  Maschinenkinematik, 667
  Orientierungsbewegungen, 666
  Orientierungsprogrammieung, 676
  Varianten der Orientierungsprogrammieung, 676
Orientierungsvektor THETA, 695
ORIEULER, 686
ORIMKS, 684
ORIPATH, 698
ORIPATHS, 698
ORIPLANE, 688
ORIRESET(A, B, C), 676
ORIROTA, 695
ORIROTC
  bei Drehung der Werkzeugorientierung, 695
  bei Interpolation der Werkzeugdrehung, 700
ORIROTR, 695
ORIROTT, 695
ORIRPY, 686
ORIRPY2, 686
ORIS, 780
ORISOF, 708
ORISOLH, 802
ORISON, 708
ORIVECT, 686
ORIVIRT1, 686
ORIVIRT2, 686
ORIWKS, 684
OS, 968
OSB, 968
OSC, 780
OSCILL, 973
OSCTRL, 968
OSD, 780
OSE, 968
OSNSC, 968
OSOF, 780
OSP1, 968
OSP2, 968
OSS, 780
OSSE, 780
OST, 780
OST1, 968
OST2, 968
OTOL, 880
OTOLG0, 185
OVR, 129
OVRA, 129
OVRRAP, 129

## P

P..., 534
P_ACTFRAME, 661
Parameter
  Aktual-, 501
  Formal-, 501
  -übergabe bei Unterprogrammaufruf, 532
  -übergabe beim Unterprogrammaufruf, 501
  Werkzeug-, 749
PAROT, 336
PAROTOF, 336
PCALL, 541
PDELAYOF, 980
PDELAYON, 980
Pendelbewegung
  Umkehrbereich, 975
  Umkehrpunkt, 975
  Zustellung im Umkehrpunkt, 977
Pendeln
  Asynchrones, 968
  Asynchrones Pendeln, 968
  Synchrones Pendeln, 973
  Teilzustellung, 975
  über Synchronaktion steuern, 973
Pfadangabe, 564
PFRAME, 639
PHI
  bei Orientierung entlang einer Kegelmantelfläche, 688
  Orientierungspolynome, 694
PHU, 415
PL
  bei Polynom-Interpolation, 614
  bei Spline-Interpolation, 602
Planachse, 172
Planfräsen - CYCLE61
  extern programmieren, 1062
Plangewinde, 229
PLC
  -Achsen, 379
PM, 273
PO, 614
PO[PHI]
  bei Drehung der Werkzeugorientierung, 698

bei Orientierung entlang einer Kegelmantelfläche, 688
Orientierungspolynome, 694
PO[PSI]
bei Drehung der Werkzeugorientierung, 698
bei Orientierung entlang einer Kegelmantelfläche, 688
Orientierungspolynome, 694
PO[THT]
bei Drehung der Werkzeugorientierung, 698
Orientierungspolynome, 694
PO[XH]
bei Orienierungsvorgabe zweier Kontaktpunkte, 691
Orientierungspolynome, 694
PO[YH]
bei Orienierungsvorgabe zweier Kontaktpunkte, 691
Orientierungspolynome, 694
PO[ZH]
bei Orienierungsvorgabe zweier Kontaktpunkte, 691
Orientierungspolynome, 694
POCKET3 - Rechtecktasche
extern programmieren, 1047
POCKET4 - Kreistasche
extern programmieren, 1050
Pol, 176
Polarkoordinaten, 29
Polarradius, 30
Polar-Transformation, 668
Polarwinkel, 30
POLF
für NC-geführtes Rückziehen, 993
für Schnellrückzug während Gewindeschneiden, 234
POLFA, 993
POLFMASK
für NC-geführtes Rückziehen, 993
für Schnellrückzug während Gewindeschneiden, 234
POLFMLIN
für NC-geführtes Rückziehen, 993
für Schnellrückzug während Gewindeschneiden, 234
POLY, 614
Polynom-Interpolation, 614
Polynomkoeffizient, 615
POLYPATH, 614
PON, 988
PONS, 980
POS, 116
POSA, 116
POSFS, 940
Positionierachsen, 377
Positionsattribute
indirekt programmieren, 442
Positionsmuster Gitter oder Rahmen - CYCLE801
extern programmieren, 1115
Positionsmuster Kreis oder Teilkreis - HOLES2
extern programmieren, 1045
Positionsmuster Reihe - HOLES1
extern programmieren, 1045
POSP, 116
POT, 446
PR, 273
PREPRO, 517
PRESETON, 650
PRESETONS, 652
PRIO, 549
PRLOC, 401
Process DataShare, 1014
Profilieren - CYCLE495
extern programmieren, 1107
Programm
-adressierung, 563
-ende, 46
Initialisierungs-, 571
-kopf, 53
-laufzeiten, 1000
-name, 43
-speicher, 560
-sprünge, 470
-verzweigung, 473
-wiederholung, 534
Programmierter Halt, 350
Programmschleife
Endschleife, 484
IF-Schleife, 482
REPEAT-Schleife, 487
WHILE-Schleife, 486
Zählschleife, 485
Programmspeicher
Dateitypen, 560
Standard-Verzeichnisse, 560
Programmteil
-wiederholung, 475
Programmteilwiederholung
mit indirekter Programmierung CALL, 539
PROTA, 736
PROTD, 738
PROTS, 737
Prozedur
vordefinierte, 1288

PSI
bei Orientierung entlang einer Kegelmantelfläche, 688
Orientierungspolynome, 694
PTP, 718
PTPG0, 718
PTPWOC, 718
PUD, 402
PUNCHACC, 980
Punkt-zu-Punkt-Fahren, 717
PUTFTOC, 764
PUTFTOCF, 763
PW, 602

## Q

QU, 347

## R

RAC, 167
Radius
effektiver, 114
Radius-Programmierung, 165
Randbedingungen bei Transformationen, 728
READ, 579
REAL, 401
REAL-Konstante, 388
Rechenparameter
Globaler, 398
Kanalspezifische, 396
Rechtecktasche - POCKET3
extern programmieren, 1047
Rechteckzapfen - CYCLE76
extern programmieren, 1073
Rechtsgewinde, 226
REDEF, 407
Referenzpunkt, 35
Referenzpunktfahren, 361
Reiben - CYCLE85
extern programmieren, 1092
RELEASE, 886
REP, 422
REPEAT, 475
REPEATB, 475
REPOSA, 869
REPOSH, 869
REPOSHA, 869
REPOSL, 869
REPOSQ, 869
REPOSQA, 869
Restzeit
für ein Werkstück, 1002
RET, 519
RET (parametrierbar), 520
RETB (parametrierbar), 526
RG, 398
RIC, 167
Richtungsvektor, 681
RINDEX, 457
RMBBL, 869
RMEBL, 869
RMIBL, 869
RMNBL, 869
RND, 248
RNDM, 248
Rohteildefinition, 1020
ROT, 319
ROTS, 325
ROUND, 446
ROUNDUP, 451
RP, 178
RPL, 319
RPY-Winkel, 680
Ruck
-begrenzung, 854
-korrektur, 877
Rückziehen
antriebsautarkes, 998
NC-geführtes, 993
Rückzug
-richtung beim Gewindeschneiden, 235
Run MyScreens, 1009
Rundung, 248

## S

S, 91
Satz, 44
ausblenden, 49
-ende, 47
-ende LF, 53
-länge, 47
-nummer, 47
Reihenfolge der Anweisungen, 47
Satzanzeige
unterdrücken, 514
Satzweise wirksam, 46
SAVE, 508
SBLOF, 509
SBLON, 509
SCALE, 328
SCC, 100

Schachtelungstiefe
von Kontrollstrukturen, 481
Schiefwinkeltransformation (TRAANG)
mit programmierbarem Winkel, 713
Schleifwerkzeuge, 75
Schneiden
-anzahl von Konturwerkzeugen, 290
-bezugspunkt, 293
-lage, 70
-lage relevante, 293
-mittelpunkt, 70
-nummer, 82
-radius, 70
Schneidennummer, 786
Schnellabheben von der Kontur, 552
Schnellrückzug
Gewindeschneiden, 234
Schnittgeschwindikeit (konstante), 100
Schrägeinstechschleifen, 714
Schrittgeschwindigkeit, 94
Schutzbereiche, 584
Schwenken Ebene / Schwenken Werkzeug / Ausrichten Werkzeug – CYCLE800
extern programmieren, 1112
SCPARA, 901
SD, 602
SD41610, 748
SD41611, 748
SD42010, 233
SD42440, 156
SD42442, 156
SD42465, 304
SD42466, 304
SD42475, 704, 707
SD42476, 704, 707
SD42477, 704, 707
SD42900, 756
SD42910, 756
SD42920, 757
SD42930, 757
SD42935, 759
SD42940, 760
SD42984, 815
SD43240, 122
SD43250, 122
SET, 422
SETAL, 1018
SETDNO, 787
SETINT, 549
SETM, 488
SETMS, 91
SETTCOR, 833
SF, 224
SIN, 446
Singuläre Stellen, 685
SLOT1 - Längsnut
extern programmieren, 1052
SLOT2 - Kreisnut
extern programmieren, 1055
SOFT, 854
SOFTA, 854
Sollwertkopplung, 943
SON, 980
Sonderwerkzeuge, 79
Sonderzeichen, 52, 53
SONS, 980
SPCOF, 120
SPCON, 120
Speicher
Arbeits-, 570
Programm-, 559
Vorlauf-, 862
SPF, 560
SPI, 892
SPIF1, 980
SPIF2, 980
Spindel
-drehrichtung, 91
-drehzahl, 91
-drehzahlbegrenzung, 106
Haupt-, 376
M-Funktionen, 350
Positionieren, 121
-tausch, 886
Spline
-Interpolation, 602
-Typen, 608
SPLINEPATH, 612
Spline-Verbund, 612
SPN, 985
SPOF, 980
SPOS, 121
SPOSA, 121
SPP, 985
Sprachmodus, 1024
SPRINT, 461
Sprung
auf Programmanfang, 469
auf Sprungmarken, 470
Sprungmarke
bei Programmsprüngen, 471
bei Programmteilwiederholungen, 475
SQRT, 446
SR, 137

SRA, 137
ST, 137
STA, 137
Stanzen, 980
START, 488
STARTFIFO, 862
Startpunkt, 35
Startpunktversatz
beim Gewindeschneiden, 225
Startpunkt-Zielpunkt, 173
STAT, 718
Stillsetzen
antriebsautarkes, 997
NC-geführtes, 997
Stirnfräsen, 683
STOLF, 185
STOPFIFO, 862
STOPRE, 862
String
formatieren, 461
-länge, 457
-operationen, 452
-Verkettung, 455
STRING, 401
STRINGIS, 1006
STRLEN, 457
SUBSTR, 458
Suchpfad
bei Unterprogrammaufruf, 568
Programmierbarer Suchpfad, 541
Suchunfähige Bereiche erfassen und suchen, 868
SUG, 105
SUPA
bei Frame abwählen, 339
bei Nullpunktverschiebung, 147
SVC, 94
S-Wert
Interpretation, 93
Synchron
-achsen, 378
Synchrones Pendeln
Auswertung IPO-Takt, 978
Nächste Teilzustellung, 979
Synchronaktionen, 976
Zuordnung von Pendel- und Zustellachse, 976
Zustellbewegung, 977
Zustellung im Umkehrbereich, 977
Zustellungen festlegen, 976
Synchronlauf
fein, 943
grob, 943
Synchronspindel
Kopplung, 940
-paar festlegen, 945
SYNR, 401
SYNRW, 401
SYNW, 401
Systemframes, 659

## T

T0, 65
TAN, 446
TANG, 958
TANGDEL, 962
TANGOF, 962
TANGON, 960
TCARR, 794
TCOABS, 794
TCOFR, 794
TCOFRX, 794
TCOFRY, 794
TCOFRZ, 794
THETA
bei Drehung der Werkzeugorientierung, 695
bei Interpolation der Werkzeugdrehung, 700
Tieflochbohren 1 - CYCLE83
extern programmieren, 1086
Tieflochbohren 2 - CYCLE830
extern programmieren, 1120
TILT, 677
TLIFT, 959
TMOF, 991
TMON, 991
TOFF, 85
TOFFL, 85
TOFFLR, 85
TOFFOF, 800
TOFFON, 800
TOFFR, 85
TOFRAME, 336
TOFRAMEX, 336
TOFRAMEY, 336
TOFRAMEZ, 336
TOLOWER, 456
TOOLENV, 821
TOROT, 336
TOROTOF, 336
TOROTX, 336
TOROTY, 336
TOROTZ, 336
TOUPPER, 456
TOWBCS, 758

TOWKCS, 758
TOWMCS, 758
TOWSTD, 758
TOWTCS, 758
TOWWCS, 758
TRAANG
mit programmierbarem Winkel, 713
TRACON, 715
TRACYL, 710
TRAFOOF, 730
TRAFOON, 740
TRAILOF, 908
TRAILON, 908
TRANS, 313
Transformation mit schwenkbarer Linearachse, 673
Transformationen
Drei- Vier- und Fünf- Achstransformation, 674
Kinematikunabhängige Grundstellung der Werkzeugorientierung, 665
Kinematische Transformationen, 665
Orientierungstransformation, 664
verkettete, 715
Verkettete Transformationen, 666
Transformationsarten
Allgemeine Funktion, 664
TRANSMIT, 709
TRAORI, 674
TRUE, 401
TRUNC, 446
TU, 722
TURN, 207

## U

ULI, 413
Umschaltbare Geometrieachsen, 893
Unterprogramm
Anwendung, 497
-aufruf indirekt, 538
-aufruf mit Parameterübergabe, 532
-aufruf modal, 535
-aufruf ohne Parameterübergabe, 530
-name, 498
Programmierbarer Suchpfad, 541
-rücksprung parametrierbar (RET ...), 520
-rücksprung parametrierbar (RETB...), 526
-wiederholung, 534
Unterprogrammaufruf mit Pfadangabe und Parametern, 541
UNTIL, 487

## V

Variable
Typenkonvertierung, 437
Variablen
anwenderdefiniert, 401
Typkonvertierung, 453
VELOLIM, 878
VELOLIMA, 856
Vergleichsoperatoren, 448
Verkettung
von Strings, 455
Verschleißwert, 753
Verweilzeit, 371
Verzeichnispfad, 566
Vorlauf
-speicher, 862
Vorlaufstopp
Interner, 373
Vorschub
für Bahnachsen, 110
für Positionierachsen, 126
für Synchronachsen, 112
-geschwindigkeit, 189
-korrektur, 129
Maßeinheiten, 113
mit Handradüberlagerung, 131
-Override, 133
Regeln, 108
Zeitreziproker, 111

## W

WAB, 273
Wahlweiser Halt, 350
WAITC, 940
WAITE, 488
WAITENC, 900
WAITM, 488
WAITMC, 488
WAITP, 116
WAITS, 121
WALCS<n>, 358
WALCS0, 358
WALIMOF, 355
WALIMON, 355
Wegberechnung, 383
Werkstück
-Hauptverzeichnis, 560
-kontur, 174

-Nullpunkt, 34
-verzeichnisse, 560
-zähler, 1003
Werkstück-Kette, 746
Werkstückkoordinatensystem, 41
Werkzeug
-gruppe, 71
-Korrekturspeicher, 70
-längenkorrektur, 68
-orientierung, 780
-orientierung bei Framewechsel, 796
-parameter, 749
-radiuskorrektur, 69
-schneide, 81
-spitze, 70
-typ, 71
-typennummer, 71
-wechsel mit M6, 66
-wechsel mit T-Nummer, 65
-wechselpunkt, 35
Werkzeugdrehzahl
maximal, 96
Werkzeug-Kette, 746
Werkzeugkorrektur
Koordinatensystem für Verschleißwerte, 757
Korrekturspeicher, 749
-Offset, 85
Werkzeugkorrekturen
additive, 752
Werkzeugorientierung
bahnrelativ, 697
Werkzeugradiuskorrektur
an Außenecken, 270
CUT2DF, 291
Eckenverzögerung, 633
Werkzeugträger
-bezugspunkt, 35
-kinematik, 789
-Orientierbare, 794
Wertzuweisung, 48
WHEN, 967
WHEN-DO, 976
WHENEVER, 967
WHENEVER-DO, 976
WHILE, 486
WKS, 41
am Werkstück ausrichten, 336
WORKPIECE, 1020
WRITE, 574
WRTPR, 353

## X

X..., 175
XOR, 448

## Y

Y..., 175

## Z

Z..., 175
Zählschleife, 485
Zeichenvorrat, 52
Zentrieren - CYCLE81
extern programmieren, 1082
Zielpunkt, 173
Zusatzachsen, 376
Zylindergewinde, 229
Zylinderkoordinaten, 179
Zylindermanteltransformation, 668