SIEMENS

SINUMERIK

SINUMERIK 840D sl
Programování NC systémů

Programovací příručka

Úvod 1

Základní bezpečnostní upozornění 2

Základy 3

Pro pokročilé 4

Tabulky 5

Přílohy A

Platí pro:

Řídící systém
SINUMERIK 840D sl / 840DE sl

Software pro CNC systémy verze 4.95

07/2021
6FC5398-2EP40-0UA2

Промэлектроника, sales@prom-elec.com

Právní upozornění

Koncept výstražných upozornění

Tato příručka obsahuje pokyny, které musíte dodržovat z důvodu své osobní bezpečnosti a zamezení materiálním škodám. Upozornění ohledně Vaší osobní bezpečnosti jsou zvýrazněny výstražným trojúhelníkem, upozornění týkající se pouze materiálních škod jsou uvedeny bez výstražného trojúhelníku. Podle stupně ohrožení jsou výstražná upozornění zobrazena v sestupném pořadí následujícím způsobem.

⚠ **NEBEZPEČÍ**
znamená, že **nastane** smrt nebo těžké ublížení na zdraví, když se neučiní příslušná bezpečnostní opatření.

⚠ **VÝSTRAHA**
znamená, že **může** nastat smrt nebo těžké ublížení na zdraví, když se neučiní příslušná bezpečnostní opatření.

⚠ **POZOR**
znamená, že může nastat lehké ublížení na zdraví, když se neučiní příslušná bezpečnostní opatření.

UPOZORNĚNÍ
znamená, že mohou nastat materiální škody, když se neučiní příslušná bezpečnostní opatření.

Při výskytu více stupňů ohrožení bude vždy použito výstražné upozornění s nejvyšším stupněm. Je-li ve výstražném upozornění s výstražným trojúhelníkem výstraha před škodami na zdraví, pak může být v tomtéž výstražném upozornění ještě připojena výstraha před materiálnhími škodami.

Kvalifikovaný personál

Výrobek nebo systém, ke kterému náleží tato dokumentace, může obsluhovat pouze **personál s odpovídající kvalifikací**, který bude při provádění stanovených úkolů dodržovat pokyny uvedené v dokumentaci, zejména pak předpisy týkající se bezpečnosti práce. Kvalifikovaný personál je na základě svého vzdělání a zkušeností způsobilý odhalit rizika v souvislosti s obsluhou těchto výrobků či systémů a zabránit možnému ohrožení.

Používání výrobků Siemens v souladu s určením

Mějte na zřeteli následující:

⚠ **VÝSTRAHA**
Výrobky Siemens se smí používat pouze pro účely uvedené v katalogu a v příslušné technické dokumentaci. Pokud se používají cizí výrobky a komponenty, musí být doporučeny nebo schváleny firmou Siemens. Bezporuchový a bezpečný provoz předpokládá odbornou přepravu, skladování, ustavení, montáž, instalaci, uvedení do provozu, obsluhu a údržbu. Musí se dodržovat přípustné podmínky prostředí. Dodržovat se musí také pokyny v příslušné dokumentaci.

Známky

Všechny názvy označené ochrannou známkou ® jsou zapsané známky firmy Siemens AG. Ostatní názvy v této tiskovině mohou být značkami, jejichž používání třetími subjekty pro své účely může porušovat práva majitelů.

Vyloučení odpovědnosti

Zkontrolovali jsme obsah tiskoviny, zda je v souladu s popsaným hardwarem a softwarem. Přesto nelze vyloučit odchylky, takže nemůžeme převzít odpovědnost za kompletní shodu. Údaje v této tiskovině jsou pravidelně kontrolovány, potřebné opravy jsou uvedeny v následujících vydáních.

Siemens AG
Digital Industries
Postfach 48 48
90026 NÜRNBERG
NĚMECKO

Objednací číslo dokumentu: 6FC5398-2EP40-0UA2
Ⓟ 11/2021 Změny vyhrazeny

Copyright © Siemens AG 2019 - 2021.
Všechna práva vyhrazena

Obsah

1 Úvod ... 19

1.1 Prostřednictvím systému SINUMERIK ... 19

1.2 O této dokumentaci ... 19

1.3 Dokumentace na internetu ... 22
1.3.1 Přehled dokumentace systému SINUMERIK 840D sl ... 22
1.3.2 Přehled dokumentace k ovládacím prvkům systému SINUMERIK ... 22

1.4 Zpětná vazba na technickou dokumentaci ... 23

1.5 mySupport-Dokumentation ... 23

1.6 Servis a podpora ... 24

1.7 Důležité informace o produktu ... 25

2 Základní bezpečnostní upozornění ... 27

2.1 Všeobecná bezpečnostní upozornění ... 27

2.2 Záruční podmínky pro příklady aplikace ... 27

2.3 Bezpečnostní pokyny ... 27

3 Základy ... 29

3.1 Geometrické základy ... 29
3.1.1 Pozice na obrobku ... 29
3.1.1.1 Vztažný systém pro údaje poloh ... 29
3.1.1.2 Kartézské souřadnice ... 29
3.1.1.3 Polární souřadnice ... 31
3.1.1.4 Absolutní rozměry ... 32
3.1.1.5 Řetězové kótování ... 34
3.1.2 Pracovní roviny ... 35
3.1.3 Počátky (nuly) souřadného systému a vztažné body ... 36
3.1.4 Souřadné systémy ... 38
3.1.4.1 Souřadný systém stroje (MCS) ... 38
3.1.4.2 Základní souřadný systém (BCS) ... 40
3.1.4.3 Základní souřadný systém počátku (nuly) (BNS) ... 41
3.1.4.4 Nastavitelný souřadný systém (ENS) ... 41
3.1.4.5 Souřadný systém obrobku (WCS) ... 42
3.1.4.6 Jak spolu souvisejí různé souřadné systémy? ... 43

3.2 Základy programování NC systémů ... 43
3.2.1 Sestavování názvů NC programů ... 44
3.2.2 Struktura a obsah NC programu ... 45
3.2.2.1 Bloky a komponenty bloku ... 45
3.2.2.2 Pravidla pro sestavování bloku ... 47
3.2.2.3 Přiřazování hodnot ... 48
3.2.2.4 Komentáře ... 49
3.2.2.5 Přeskakování bloků ... 49

3.3 Založit NC program ... 51
3.3.1 Základní postup ... 51
3.3.2 Znaky, které jsou k dispozici ... 53
3.3.3 Hlavička programu ... 54
3.3.4 Příklady programování ... 55
3.3.4.1 Příklad 1: První kroky při programování ... 55
3.3.4.2 Příklad 2: NC program k soustružení ... 56
3.3.4.3 Příklad 3: NC program ke frézování ... 57

3.4 Výměna nástroje ... 60
3.4.1 Výměna nástroje, když je aktivní správa nástrojů ... 61
3.4.1.1 Výměna nástroje, když je aktivní aktivní správa nástrojů s funkcí T=místo ... 62
3.4.1.2 Výměna nástroje, když je aktivní aktivní správa nástrojů s příkazem M6 ... 63
3.4.2 Výměna nástroje, když správa nástrojů není aktivní ... 65
3.4.2.1 Výměna nástroje pomocí T-čísla, když správa nástrojů není aktivní ... 65
3.4.2.2 Výměna nástroje pomocí M6, když správa nástrojů není aktivní ... 66
3.4.3 Chování v případě nesprávně naprogramovaného T-čísla ... 67

3.5 Korekční parametry nástroje ... 67
3.5.1 Naprogramovaná kontura a dráha nástroje ... 67
3.5.2 Korekce délky nástroje ... 68
3.5.3 Korekce rádiusu nástroje ... 69
3.5.4 Paměť korekčních parametrů nástrojů ... 69
3.5.5 Typy nástrojů ... 71
3.5.5.1 Číslo typu nástroje a skupiny nástrojů ... 71
3.5.5.2 frézovací nástroje ... 71
3.5.5.3 Vrták ... 73
3.5.5.4 Brusné nástroje ... 74
3.5.5.5 Soustružnické nástroje ... 76
3.5.5.6 Speciální nástroje ... 78
3.5.6 Aktivování/deaktivování korekčních parametrů nástroje (D, D0) ... 80
3.5.7 Potlačení korekcí nástroje (SUPD) ... 82
3.5.8 Programovatelný offset korekce nástroje (TOFFL, TOFF, TOFFR, TOFFLR) ... 84

3.6 pohyby vřetena ... 89
3.6.1 Otáčky vřetena (S), směr otáčení vřetena (M3, M4, M5) ... 89
3.6.2 Řezná rychlost nástroje (SVC) ... 93
3.6.3 Konstantní řezná rychlost (G96/G961/G962, G97/G971/G972, G973, LIMS, SCC) ... 99
3.6.4 Aktivování/deaktivování konstantní obvodové rychlosti brusného kotouče (GWPSON, GWPSOF) ... 104
3.6.5 Programovatelné omezení otáček vřetena (G25, G26) ... 105

3.7 Regulace posuvu ... 106
3.7.1 Posuv (G93, G94, G95, F, FGROUP, FL, FGREF) ... 106
3.7.2 Najíždění polohovacími osami (POS, POSA, POSP, FA, WAITP, WAITMC) ... 114
3.7.3 Vřeteno v režimu regulace polohy (SPCON, SPCOF) ... 117
3.7.4 Nastavování polohy vřetena (SPOS, SPOSA, M19, M70, WAITS) ... 118
3.7.5 Posuv pro polohovací osy/vřetena (FA, FPR, FPRAON, FPRAOF) ... 123
3.7.6 Programovatelná korekce posuvu (OVR, OVRRAP, OVRA) ... 126
3.7.7 Programovatelná korekce zrychlení (ACC) ... 127
3.7.8 Posuv s korekcí ručním kolečkem (FD, FDA) ... 129
3.7.9 Optimalizace posuvu na zakřivených úsecích dráhy (CFTCP, CFC, CFIN) ... 132
3.7.10 Větší počet hodnot posuvu v jednom bloku (F, ST, SR, FMA, STA, SRA) ... 135
3.7.11 Blokový posuv (FB) ... 138

3.7.12 Posuv na zub (G95 FZ) ... 139

3.8 Geometrická nastavení ... 144
3.8.1 Nastavitelná posunutí počátku (G54 ... G57, G505 ... G599, G53, G500, SUPA, G153) ... 144
3.8.2 Nastavitelná posunutí počátku (G54 ... G57, G505 ... G599, G53, G500, SUPA, G153): Další informace ... 146
3.8.3 Volba pracovní roviny (G17/G18/G19) ... 147
3.8.4 Údaje rozměrů ... 150
3.8.4.1 Zadávání absolutních rozměrů (G90, AC) ... 151
3.8.4.2 Zadávání inkrementálních rozměrů (G91, IC) ... 153
3.8.4.3 Zadávání absolutních a inkrementálních rozměrů při soustružení a frézování (G90/G91) ... 157
3.8.4.4 Zadávání absolutních rozměrů pro kruhové osy (DC, ACP, ACN) ... 158
3.8.4.5 Systém měřicích jednotek palce/metrické jednotky (G70/G71, G700/G710) ... 160
3.8.4.6 Programování rádiusů/průměrů ve specifickém kanálu (DIAMON, DIAM90, DIAMOF, DIAMCYCOF) ... 164
3.8.4.7 Programování rádiusů/průměrů pro specifickou osu (DIAMONA, DIAM90A, DIAMOFA, DIACYCOFA, DIAMCHANA, DIAMCHAN, DAC, DIC, RAC, RIC) ... 166
3.8.5 Poloha obrobku při soustružení ... 170

3.9 Příkazy dráhy ... 172
3.9.1 Všeobecné informace týkající se příkazů dráhy ... 172
3.9.2 Příkazy posuvu s kartézskými souřadnicemi (G0, G1, G2, G3, X..., Y..., Z...) ... 173
3.9.3 Příkazy posuvu s polárními souřadnicemi ... 175
3.9.3.1 Vztažný bod polárních souřadnic (G110, G111, G112) ... 175
3.9.3.2 Příkazy posuvů pomocí polárních souřadnic (G0, G1, G2, G3, AP, RP) ... 176
3.9.4 Pohyby rychlým posuvem ... 180
3.9.4.1 Aktivování rychlého posuvu (G0) ... 180
3.9.4.2 Aktivování/deaktivování lineární interpolace pro pohyby rychlým posuvem (RTLION, RTLIOF) ... 182
3.9.4.3 Přizpůsobení tolerancí pro pohyby rychlým posuvem (STOLF, CTOLG0, OTOLG0) ... 184
3.9.5 Přímková interpolace (G1) ... 188
3.9.6 Kruhová interpolace ... 190
3.9.6.1 Přehled ... 190
3.9.6.2 Kruhová interpolace se středem a koncovým bodem (G2/G3, X... Y... Z..., I... J... K...) ... 191
3.9.6.3 Kruhová interpolace s rádiusem a koncovým bodem (G2/G3, X... Y... Z..., CR) ... 194
3.9.6.4 Kruhová interpolace s úhlem kruhové výseče a koncovým bodem/středem (G2/G3, X... Y... Z... / I... J... K..., AR) ... 196
3.9.6.5 Kruhová interpolace pomocí polárních souřadnic (G2/G3, AP, RP) ... 198
3.9.6.6 Kruhová interpolace s vnitřním a koncovým bodem (CIP, X... Y... Z..., I1... J1... K1...) ... 199
3.9.6.7 Kruhová interpolace s tangenciálním přechodem (CT, X... Y... Z...) ... 202
3.9.7 Spirální interpolace (G2/G3, TURN) ... 206
3.9.8 Evolventní interpolace (INVCW, INVCCW) ... 208
3.9.9 Definice kontur ... 213
3.9.9.1 Programování konturové křivky ... 213
3.9.9.2 Definice kontur: Přímka ... 214
3.9.9.3 Definice kontur: Dvě přímky ... 216
3.9.9.4 Definice kontur: Tři přímky ... 219
3.9.9.5 Definice kontur: Programování koncového bodu pomocí úhlu ... 223
3.9.10 Řezání závitu ... 223
3.9.10.1 Řezání závitu s konstantním stoupáním (G33, SF) ... 223
3.9.10.2 Řezání závitů s narůstajícím nebo s klesajícím stoupáním (G34, G35) ... 230
3.9.10.3 Programovatelný náběh a výběh závitu pomocí příkazů G33, G34 a G35 (DITS, DITE) ... 231

3.9.10.4 Rychlý zpětný pohyb během řezání závitu (LFON, LFOF, DILF, ALF, LFTXT, LFWP, LFPOS, POLF, POLFMASK, POLFMLIN) ... 233
3.9.10.5 Sudovitý závit (G335, G336) ... 237
3.9.11 Vrtání závitů bez vyrovnávací hlavičky ... 242
3.9.11.1 Vrtání závitů bez vyrovnávací hlavičky a zpětného pohybu (G331, G332) ... 242
3.9.11.2 Příklad: Vrtání závitů pomocí příkazu G331 / G332 ... 244
3.9.11.3 Příklad: Výstup naprogramovaných otáček při vrtání s aktuálním převodovým stupněm ... 244
3.9.11.4 Příklad: Použití datového bloku druhého převodového stupně ... 245
3.9.11.5 Příklad: Žádné naprogramované otáčky, monitorování převodového stupně ... 245
3.9.11.6 Příklad: Změna převodového stupně není možná, monitorování převodového stupně ... 245
3.9.11.7 Příklad: Programování bez příkazu SPOS ... 246
3.9.12 Vrtání závitů s vyrovnávací hlavičkou ... 246
3.9.12.1 Vrtání závitů s vyrovnávací hlavičkou a se zpětným pohybem (G63) ... 246
3.9.13 Faseta, zaoblení (CHF, CHR, RND, RNDM, FRC, FRCM) ... 247

3.10 Korekce rádiusu nástroje ... 253
3.10.1 Korekce rádiusu nástroje (G40, G41, G42, OFFN) ... 253
3.10.2 Najíždění na konturu a odjíždění od ní (NORM, KONT, KONTC, KONTT) ... 263
3.10.3 Korekce na vnějších rozích (G450, G451, DISC) ... 270
3.10.4 Měkké najíždění a odjíždění ... 273
3.10.4.1 Najíždění a odjíždění (G140 až G143, G147, G148, G247, G248, G347, G348, G340, G341, DISR, DISCL, DISRP, FAD, PM, PR) ... 273
3.10.4.2 Najíždění a odjíždění s rozšířenými strategiemi odjíždění (G460, G461, G462) ... 284
3.10.5 Aktivování/deaktivování monitorování kolize („rozpoznání hrdla láhve") (CDON, CDOF, CDOF2) ... 287
3.10.6 2 1/2 D-korekce nástroje (CUT2D, CUT2DD, CUT2DF, CUT2DFD) ... 289
3.10.7 Udržení konstantní korekce rádiusu nástroje (CUTCONON, CUTCONOF) ... 292
3.10.8 Nástroje se specifickou polohou břitu ... 294

3.11 Chování při pohybu po dráze ... 296
3.11.1 Přesné najetí (G60, G9, G601, G602, G603) ... 296
3.11.2 Režim řízení pohybu po dráze (G64, G641, G642, G643, G644, G645, ADIS, ADISPOS) ... 298

3.12 Transformace souřadného systému (Frame) ... 307
3.12.1 Framy ... 307
3.12.2 Příkazy framů ... 309
3.12.3 Programovatelné posunutí počátku (TRANS, ATRANS) ... 312
3.12.4 Programovatelné posunutí počátku (G58, G59) ... 316
3.12.5 Programovatelné otočení (ROT, AROT, RPL) ... 318
3.12.6 Programové otočení framu o prostorový úhel (ROTS, AROTS, CROTS) ... 324
3.12.7 Programovatelná změna měřítka (SCALE, ASCALE) ... 327
3.12.8 Programovatelné zrcadlové převrácení (MIRROR, AMIRROR) ... 330
3.12.9 Generování framu v závislosti na orientaci nástroje (TOFRAME, TOROT, PAROT) ... 335
3.12.10 Deaktivování framu (G53, G153, SUPA, G500) ... 338
3.12.11 Programování: Deaktivování superponovaných pohybů pro specifickou osu (CORROF) ... 339
3.12.12 Deaktivování aditivních posunutí počátku (DRFOF) ... 341
3.12.13 Posunutí počátku specifická pro technologii broušení (GFRAME0, GFRAME1 ... GFRAME100) ... 343

3.13 Výstupy pomocných funkcí ... 344
3.13.1 M-funkce ... 346

3.14 Doplňkové příkazy ... 349
3.14.1 Výstup hlášení (MSG) ... 349
3.14.2 Zápis řetězce znaků do proměnné BTSS (WRTPR) ... 351

3.14.3 Ohraničení pracovního pole ... 352
3.14.3.1 Ohraničení pracovního pole v BCS (G25/G26, WALIMON, WALIMOF) ... 352
3.14.3.2 Ohraničení pracovního pole ve WCS/ENS (WALCS0 ... WALCS10) ... 355
3.14.4 Najíždění na referenční bod (G74) ... 358
3.14.5 Najíždění na pevný bod (G75) ... 359
3.14.6 Najíždění na pevný doraz (FXS, FXST, FXSW) ... 363
3.14.7 Doba prodlevy (G4) ... 368
3.14.8 Zastavení interního předběžného zpracování ... 370

3.15 Ostatní informace ... 370
3.15.1 Osy ... 370
3.15.1.1 Hlavní osy / geometrické osy ... 371
3.15.1.2 Pomocné (doplňkové) osy ... 372
3.15.1.3 Hlavní vřeteno, řídící vřeteno ... 373
3.15.1.4 Osy stroje ... 373
3.15.1.5 Kanálové osy ... 373
3.15.1.6 Dráhové osy ... 374
3.15.1.7 Polohovací osy ... 374
3.15.1.8 Synchronizované osy ... 375
3.15.1.9 Příkazové osy ... 375
3.15.1.10 Osy PLC ... 375
3.15.1.11 Spřažené osy ... 376
3.15.1.12 Řídící spřažené osy ... 377
3.15.2 Od příkazu pohybu k pohybu stroje ... 379
3.15.3 Výpočet dráhy ... 379
3.15.4 Adresy ... 380
3.15.5 Název ... 382
3.15.6 Konstanty ... 384
3.15.7 Operátory a matematické funkce ... 386

4 Pro pokročilé ... 389

4.1 Flexibilní programování NC systémů ... 389
4.1.1 Proměnné ... 389
4.1.1.1 Systémová data ... 390
4.1.1.2 Předdefinované uživatelské proměnné: Kanálový početní parametr (R) ... 392
4.1.1.3 Předdefinované uživatelské proměnné: Globální početní parametry (RG) ... 394
4.1.1.4 Předdefinované uživatelské proměnné: linkové proměnné ... 395
4.1.1.5 Definice uživatelských proměnných (DEF) ... 397
4.1.1.6 Opětovná definice systémových dat, uživatelských dat a příkazů NC jazyka (REDEF) ... 403
4.1.1.7 Atribut: Inicializační hodnota ... 406
4.1.1.8 Atribut: Mezní hodnoty (LLI, ULI) ... 409
4.1.1.9 Atribut: Fyzikální jednotka (PHU) ... 411
4.1.1.10 Atribut: Přístupová oprávnění (APR, APW, APRP, APWP, APRB, APWB) ... 413
4.1.1.11 Přehled definovatelných a znovu definovatelných atributů ... 417
4.1.1.12 Definice a inicializace proměnných typu pole (DEF, SET, REP) ... 419
4.1.1.13 Datové typy ... 424
4.1.1.14 Minimum, maximum a rozsah proměnných (MINVAL, MAXVAL, BOUND) ... 425
4.1.1.15 Kontrola existence proměnné (ISVAR) ... 426
4.1.1.16 Načítání hodnot atributu/datového typu (GETVARPHU, GETVARAP, GETVARLIM, GETVARDIM, GETVARDFT, GETVARTYP) ... 428
4.1.1.17 Možné převádění typů ... 433
4.1.2 Nepřímé programování ... 434
4.1.2.1 Nepřímé programování adres ... 434

4.1.2.2 Nepřímé programování příkazů G-funkcí ... 437
4.1.2.3 Nepřímé programování atributů polohy (GP) ... 438
4.1.2.4 Nepřímé programování řádků výrobního programu (EXECSTRING) ... 440
4.1.3 Operace ... 441
4.1.3.1 Matematické funkce ... 441
4.1.3.2 Porovnávací a logické operace ... 443
4.1.3.3 Priorita operací ... 445
4.1.3.4 Korekce přesnosti při příkazech porovnávání (TRUNC) ... 446
4.1.3.5 Zaokrouhlení (ROUNDUP) ... 447
4.1.4 Operace s řetězci ... 448
4.1.4.1 Převod do datového typu STRING (AXSTRING) ... 449
4.1.4.2 Převod z datového typu STRING (NUMBER, ISNUMBER, AXNAME) ... 449
4.1.4.3 Zřetězení proměnných typu String (<<) ... 450
4.1.4.4 Převádění na malá/velká písmena (TOLOWER, TOUPPER) ... 452
4.1.4.5 Zjištění délky řetězce (STRLEN) ... 452
4.1.4.6 Vyhledávání znaků/řetězců v řetězci (INDEX, RINDEX, MINDEX, MATCH) ... 453
4.1.4.7 Vybírání dílčího řetězce (SUBSTR) ... 454
4.1.4.8 Načítání a zápis jednotlivých znaků ... 455
4.1.4.9 Formátování řetězce (SPRINT) ... 456
4.1.5 Programové skoky a větvení ... 464
4.1.5.1 Skok zpátky na začátek programu (GOTOS) ... 464
4.1.5.2 Programové skoky na návěští skoků (GOTOB, GOTOF, GOTO, GOTOC) ... 465
4.1.5.3 Větvení programu (CASE ... OF ... DEFAULT ...) ... 468
4.1.6 Opakování části programu (REPEAT, REPEATB, ENDLABEL, P) ... 470
4.1.7 Řídící struktury ... 476
4.1.7.1 Podmíněné příkazy a větvení (IF, ELSE, ENDIF) ... 478
4.1.7.2 Nekonečná programová smyčka (LOOP, ENDLOOP) ... 480
4.1.7.3 Smyčka s počítadlem (FOR ... TO ..., ENDFOR) ... 481
4.1.7.4 Programová smyčka s podmínkou na začátku smyčky (WHILE, ENDWHILE) ... 482
4.1.7.5 Programová smyčka s podmínkou na konci smyčky (REPEAT, UNTIL) ... 483
4.1.7.6 Příklad programu se vnořenými řídícími strukturami ... 483
4.1.8 Koordinace programů přesahující hranice kanálu (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) ... 484
4.1.9 Technika maker (DEFINE ... AS) ... 490

4.2 Technika podprogramů ... 492
4.2.1 Všeobecně ... 492
4.2.1.1 Podprogram ... 492
4.2.1.2 Názvy podprogramů ... 494
4.2.1.3 Vnoření podprogramů ... 495
4.2.1.4 Cesta pro vyhledávání ... 496
4.2.1.5 Formální a skutečný parametr ... 497
4.2.1.6 Předávání parametrů ... 497
4.2.2 Definice podprogramu ... 499
4.2.2.1 Podprogram bez předávání parametrů ... 499
4.2.2.2 Podprogram s předáváním parametrů Call-by-Value (PROC) ... 500
4.2.2.3 Podprogram s předáváním parametrů Call-by-Reference (PROC, VAR) ... 502
4.2.2.4 Ukládání modálních G-funkcí (SAVE) ... 504
4.2.2.5 Potlačení zpracovávání blok po bloku (SBLOF, SBLON) ... 505
4.2.2.6 Potlačení vypisování aktuálního bloku (DISPLOF, DISPLON, ACTBLOCNO) ... 510
4.2.2.7 Označení podprogramů s přípravou (PREPRO) ... 513
4.2.2.8 Návrat zpět z podprogramu - M17 ... 513
4.2.2.9 Návrat z podprogramu pomocí příkazu RET ... 514

4.2.2.10 Návrat z podprogramu s nastavitelnými parametry (RET ...) ... 515
4.2.2.11 Návrat z podprogramu s nastavitelnými parametry (RETB ...): ... 522
4.2.3 Volání podprogramu ... 526
4.2.3.1 Volání podprogramu bez předávání parametrů ... 526
4.2.3.2 Volání podprogramu s předáváním parametrů (EXTERN) ... 528
4.2.3.3 Počet opakování programu (P) ... 531
4.2.3.4 Modální volání podprogramu (MCALL) ... 532
4.2.3.5 Nepřímé volání podprogramu (CALL) ... 534
4.2.3.6 Nepřímé volání podprogramu se zadáním úseku programu, který á být zpracován (CALL BLOCK ... TO ...) ... 535
4.2.3.7 Nepřímé volání programu naprogramovaného v jazyce ISO (ISOCALL) ... 537
4.2.3.8 Volání podprogramu s udáním cesty a parametrů (PCALL) ... 537
4.2.3.9 Rozšíření cesty pro vyhledávání při volání podprogramu (CALLPATH) ... 538
4.2.3.10 Zpracovávání externího podprogramu (EXTCALL) ... 539

4.3 Rutiny přerušení (ASUP) ... 544
4.3.1 Funkce rutiny přerušení ... 544
4.3.2 Sestavování rutin přerušení ... 545
4.3.3 Přiřazování a spouštění rutin přerušení (SETINT, PRIO, BLSYNC) ... 545
4.3.4 Deaktivování/opětovné aktivování přiřazení rutiny přerušení (DISABLE, ENABLE) ... 548
4.3.5 Vymazání přiřazení rutiny přerušení (CLRINT) ... 548
4.3.6 Rychlé pozvednutí od kontury (SETINT LIFTFAST, ALF) ... 549
4.3.7 Směr pohybu při rychlém pozvednutí od kontury ... 551
4.3.8 Posloupnost pohybů při rutinách přerušení ... 554

4.4 Správa souborů a programů ... 555
4.4.1 Programová paměť ... 555
4.4.1.1 Paměť programů v NCK ... 555
4.4.1.2 Externí programová paměť ... 558
4.4.1.3 Adresování souborů v paměti pro programy ... 559
4.4.1.4 Cesta pro vyhledávání při volání podprogramu ... 564
4.4.1.5 Zjištění cesty a názvu souboru ... 566
4.4.2 Pracovní paměť (CHANDATA, COMPLETE, INITIAL) ... 567

4.5 Manipulace se soubory ... 570
4.5.1 Zápis do souboru (WRITE) ... 570
4.5.2 Vymazání souboru (DELETE) ... 573
4.5.3 Čtení řádků v souboru (READ) ... 574
4.5.4 Kontrola existence souboru (ISFILE) ... 576
4.5.5 Zjišťování informací o souboru (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO) ... 577

4.6 Chráněné oblasti ... 580
4.6.1 Definice chráněných oblastí (CPROTDEF, NPROTDEF) ... 580
4.6.2 Aktivování/deaktivování chráněné oblasti (CPROT, NPROT) ... 583
4.6.3 Kontrola narušení chráněné oblasti, ohraničení pracovního pole a softwarových koncových spínačů (CALCPOSI) ... 587

4.7 Speciální příkazy dráhy ... 597
4.7.1 Najíždění na kódované pozice (CAC, CIC, CDC, CACP, CACN) ... 597
4.7.2 Splinová interpolace (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) ... 598
4.7.3 Splinový svazek (SPLINEPATH) ... 609
4.7.4 Aktivování/deaktivování komprese NC bloků (COMPON, COMPCURV, COMPCAD, COMPSURF, COMPOF) ... 610
4.7.5 Polynomická interpolace (POLY, POLYPATH, PO, PL) ... 611

4.7.6 Nastavitelný vztah k dráze (SPATH, UPATH) ... 616
4.7.7 Měření ve specifickém kanálu (MEAS, MEAW) ... 618
4.7.8 Měření pomocí specifické osy (MEASA, MEAWA, MEAC) (volitelný doplněk) ... 621
4.7.9 Speciální funkce pro uživatele OEM (OMA1 ... OMA5, OEMIPO1, OEMIPO2, G810 ... G829) ... 629
4.7.10 Snížení posuvu se zpožděním v rozích (FENDNORM, G62, G621) ... 630
4.7.11 Programovatelné kritérium konce pohybu (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA) ... 631

4.8 Transformace souřadného systému (Frame) ... 634
4.8.1 Transformace souřadnic pomocí proměnné typu frame ... 634
4.8.1.1 Předdefinované proměnné typu frame ($P_CHBFRAME, $P_IFRAME, $P_PFRAME, $P_ACTFRAME) ... 636
4.8.2 Přiřazování hodnoty framu ... 639
4.8.2.1 Přímé přiřazení hodnoty (hodnota osy, úhlu, měřítko) ... 639
4.8.2.2 Načítání a editace složek framu (TR, FI, RT, SC, MI) ... 641
4.8.2.3 Výpočety s framy ... 643
4.8.2.4 Definice proměnných typu frame (DEF FRAME) ... 644
4.8.3 Hrubé a jemné posunutí (CTRANS, CFINE) ... 645
4.8.4 Externí posunutí počátku ($AA_ETRANS) ... 646
4.8.5 Nastavení skutečné hodnoty se ztrátou stavu najetí na referenční bod (PRESETON) ... 648
4.8.6 Nastavení skutečné hodnoty bez ztráty stavu najetí na referenční bod (PRESETONS) ... 649
4.8.7 Výpočet framu na základě 3 změřených bodů v prostoru (MEAFRAME) ... 651
4.8.8 Globální framy ... 654
4.8.8.1 Specifické kanálové framy ($P_CHBFR, $P_UBFR) ... 655
4.8.8.2 Framy aktivní v kanálu ... 656

4.9 Transformace ... 660
4.9.1 Všeobecné programování různých druhů transformace ... 660
4.9.1.1 Pohyby provádějící změnu orientace při transformacích ... 663
4.9.1.2 Přehled transformace orientace TRAORI ... 667
4.9.2 Transformace ve třech, čtyřech a pěti osách (TRAORI) ... 669
4.9.2.1 Všeobecné souvislosti v případě kardanové nástrojové hlavičky ... 669
4.9.2.2 Transformace ve třech, čtyřech a pěti osách (TRAORI) ... 671
4.9.2.3 Varianty programování orientace a základního nastavení (ORIRESET) ... 673
4.9.2.4 Programování orientace nástroje (A..., B..., C..., LEAD, TILT) ... 674
4.9.2.5 Frézování na čelní ploše (A4, B4, C4, A5, B5, C5) ... 680
4.9.2.6 Vztah orientačních os (ORIWKS, ORIMKS) ... 681
4.9.2.7 Programování orientačních os (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2) ... 683
4.9.2.8 Programování orientace podél kuželové plášťové plochy (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) ... 685
4.9.2.9 Zadání orientace pomocí druhého styčného bodu (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=) ... 688
4.9.3 Polynomický popis orientace (PO[úhel], PO[souřadnice]) ... 690
4.9.4 Otáčení orientace nástroje (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) ... 692
4.9.5 Orientace vzhledem k dráze ... 694
4.9.5.1 Druhy orientace vztažené k dráze ... 694
4.9.5.2 Otáčení orientace nástroje vzhledem k dráze (ORIPATH, ORIPATHS, úhel otočení) ... 695
4.9.5.3 Interpolace otočení nástroje vzhledem k dráze (ORIROTC, THETA) ... 697
4.9.5.4 Vyhlazení průběhu orientace (ORIPATHS A8=, B8=, C8=) ... 699
4.9.6 Komprimování orientace (COMPON, COMPCURV, COMPCAD, COMPSURF) ... 700
4.9.7 Aktivování/deaktivování vyhlazování charakteristiky orientace (ORISON, ORISOF): ... 703
4.9.8 Kinematická transformace ... 704

4.9.8.1 Aktivování transformace čelní plochy (TRANSMIT) ... 704
4.9.8.2 Aktivování transformace válcového pláště (TRACYL) ... 704
4.9.8.3 Aktivování transformace šikmého úhlu s naprogramovatelným úhlem (TRAANG) ... 707
4.9.8.4 Šikmé zapichování na bruskách (G5, G7) ... 709
4.9.9 Aktivování zřetězené transformace (TRACON) ... 710
4.9.10 Posuv PTP v kartézských souřadnicích ... 712
4.9.10.1 Aktivování/deaktivování posuvu PTP v kartézských souřadnicích (PTP, PTPG0, PTPWOC, CP) ... 712
4.9.10.2 Zadávání polohy kloubu (STAT) ... 713
4.9.10.3 Zadání znaménka úhlu osy (TU) ... 717
4.9.10.4 Příklad 1: Pohyb PTP robota se 6 osami pomocí transformace ROBX ... 720
4.9.10.5 Příklad 2: Najíždění PTP při obecné 5-osé transformaci ... 721
4.9.10.6 Příklad 3: PTPG0 a TRANSMIT ... 722
4.9.11 Okrajové podmínky při aktivování transformace ... 723
4.9.12 Deaktivování transformace (TRAFOOF) ... 724

4.10 Kinematické řetězce ... 725
4.10.1 Mazání komponent (DELOBJ) ... 725
4.10.2 Zjišťování indexu podle názvu (NAMETOINT) ... 728

4.11 Zabránění vzniku kolizí u kinematických řetězců ... 729
4.11.1 Kontrola dvojice oblastí chráněných kvůli kolizi (COLLPAIR) ... 729
4.11.2 Vyvolání nového výpočtu modelu stroje pro prevenci kolizí (PROTA) ... 729
4.11.3 Nastavení stavu chráněné oblasti (PROTS) ... 730
4.11.4 Stanovení vzdálenosti mezi dvěma chráněnými oblastmi (PROTD) ... 731

4.12 Transformace pomocí kinematických řetězců ... 733
4.12.1 Aktivování transformace (TRAFOON) ... 733
4.12.2 Modifikace transformace orientace po měření stroje (CORRTRAFO) ... 734

4.13 Korekční parametry nástroje ... 741
4.13.1 Paměť korekcí ... 741
4.13.2 Aditivní korekce ... 744
4.13.2.1 Aktivování aditivních korekcí (DL) ... 744
4.13.2.2 Definice opotřebení a seřizovacích hodnot ($TC_SCPxy[t,d], $TC_ECPxy[t,d]) ... 745
4.13.2.3 Vymazání aditivních korekcí (DELDL) ... 745
4.13.3 Korekce nástroje - speciální zacházení ... 746
4.13.3.1 Zrcadlové převrácení délek nástroje ... 748
4.13.3.2 Vyhodnocování znaménka opotřebení ... 748
4.13.3.3 Souřadný systém aktivního obrábění (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS) ... 749
4.13.3.4 Délka nástroje a změna roviny ... 752
4.13.4 On-line korekce nástroje ... 753
4.13.4.1 Definice polynomické funkce (FCTDEF) ... 753
4.13.4.2 Zápis on-line korekce nástroje, kontinuálně (PUTFTOCF) ... 755
4.13.4.3 Diskrétní zápis on-line korekce nástroje (PUTFTOC) ... 756
4.13.4.4 Aktivování/deaktivování on-line korekce nástroje (FTOCON/FTOCOF) ... 756
4.13.5 3D korekce rádiusu nástroje ... 757
4.13.5.1 Aktivování 3D korekce rádiusu nástroje pro 3D obvodové frézování (CUT3DC, CUT3DCD, ISD) ... 757
4.13.5.2 Aktivování 3D korekce rádiusu nástroje pro 3D frézování na čelní ploše (CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD) ... 761
4.13.5.3 3D obvodové frézování se zohledněním hraniční plochy (CUT3DCC, CUT3DCCD) ... 767
4.13.6 Orientace nástroje (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST) ... 772
4.13.7 Volné zadávání D-čísel, číslo břitu ... 778

4.13.7.1 Volné zadávání D-čísel, číslo břitu (adresa CE) ... 778
4.13.7.2 Volné zadávání D-čísel: Kontrola D-čísla (CHKDNO) ... 778
4.13.7.3 Volné zadávání D-čísel: Přejmenovávání D-čísel (GETDNO, SETDNO) ... 778
4.13.7.4 Volné zadávání D-čísel: Zjišťování T-čísla k zadanému D-číslu (GETACTTD) ... 779
4.13.7.5 Volné zadávání D-čísel: Nastavení D-čísla jako neplatného (DZERO) ... 780
4.13.8 Kinematika držáku nástroje ... 780
4.13.9 Délková korekce nástroje pro orientovatelný držák nástroje (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ) ... 786
4.13.10 Modifikace orientovatelného držáku nástroje po měření stroje (CORRTC) ... 790
4.13.11 On-line korekce délky nástroje (TOFFON, TOFFOF) ... 793
4.13.12 Modifikace korekčních parametrů u točivých nástrojů ... 796
4.13.12.1 Výpočet orientace (ORISOLH) ... 796
4.13.12.2 Aktivování modifikací korekčních parametrů u točivých nástrojů (CUTMOD, CUTMODK) ... 806
4.13.13 Práce s nástrojovým prostředím ... 813
4.13.13.1 Uložení nástrojového prostředí (TOOLENV) ... 813
4.13.13.2 Vymazání nástrojového prostředí (DELTOOLENV) ... 816
4.13.13.3 Načtení T-čísla, D-čísla a DL-čísla (GETTENV) ... 817
4.13.13.4 Načítání informací týkajících se uložených nástrojových prostředí ($P_TOOLENVN, ($P_TOOLENV) ... 818
4.13.13.5 Načtení délek nástroje, příp. složek délky nástroje (GETTCOR) ... 818
4.13.13.6 Změna komponent nástroje (SETTCOR) ... 824
4.13.14 Načítání přiřazení délek nástroje L1, L2, L3 osám souřadného systému (LENTOAX) ... 836

4.14 Chování při pohybu po dráze ... 839
4.14.1 Průběh charakteristiky posuvu (FNORM, FLIN, FCUB, FPO) ... 839
4.14.2 Chování zrychlení: ... 843
4.14.2.1 Způsoby chování při změnách zrychlení (BRISK, BRISKA, SOFT, SOFTA, DRIVE, DRIVEA) ... 843
4.14.2.2 Ovlivňování chování zrychlení u vlečných os (VELOLIMA, ACCLIMA, JERKLIMA) ... 845
4.14.2.3 Aktivování specifických technologických hodnot dynamiky (DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN, DYNFINISH), DYNPREC ... 847
4.14.3 Najíždění s dopřednou regulací (FFWON, FFWOF) ... 849
4.14.4 Programování tolerance kontury/orientace (CTOL, OTOL, ATOL) ... 849
4.14.5 Aktivování/deaktivování programovatelné přesnosti kontury (CPRECON, CPRECOF) ... 854
4.14.6 Aktivování/deaktivování automatického přepínání filtru (AFISON, AFISOF) ... 856
4.14.7 Zpracování programu s pamětí předběžného zpracování (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE) ... 859
4.14.8 Definice oblasti odloženého zastavení (DELAYFSTON, DELAYFSTOF) ... 861
4.14.9 Zabránění pozice programu pro SERUPRO (IPTRLOCK, IPTRUNLOCK) ... 863
4.14.10 Opětovné najíždění na konturu (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL) ... 866
4.14.11 Ovlivňování vedení pohybu ... 874
4.14.11.1 Přizpůsobení maximální rychlosti osy nebo otáček vřetena (VELOLIM) ... 874
4.14.11.2 Přizpůsobení maximálního osového ryvu (JERKLIM) ... 876
4.14.11.3 Přizpůsobení maximální rychlosti pohybu po dráze (FLIM) ... 877
4.14.11.4 Přizpůsobení maximálního zrychlení při pohybu po dráze (PACCLIM) ... 879
4.14.12 Chování při přechodu na další blok, když je aktivní vazba (CPBC) ... 880

4.15 Funkce pro osy ... 881
4.15.1 Výměna osy, výměna vřetena (RELEASE, GET, GETD) ... 881
4.15.2 Předávání osy jinému kanálu (AXTOCHAN) ... 885
4.15.3 Osové funkce (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) ... 887
4.15.4 Přepínatelné geometrické osy (GEOAX) ... 889
4.15.5 Osový zásobník (AXCTSWE, AXCTSWED, AXCTSWEC) ... 893

4.15.6 Čekání na platnou pozici osy (WAITENC) ... 895
4.15.7 Programovatelné přepnutí sady parametrů (SCPARA) ... 896
4.15.8 Aktivování/deaktivování přizpůsobování (CADAPTON, CADAPTOF) ... 897
4.15.9 Přizpůsobení filtru ryvu FIR režimu dynamiky (CALCFIR) ... 900
4.15.10 Čtení/zápis do parametrů pohonu ve výrobním programu (DRVPRD, DRVPWR) ... 902

4.16 Spojení os vazbou ... 905
4.16.1 Vlečení (TRAILON, TRAILOF) ... 905
4.16.2 Tabulky křivek (CTAB) ... 909
4.16.2.1 Definice tabulek křivek (CTABDEF, CATBEND) ... 909
4.16.2.2 Kontrola existence tabulky křivky (CTABEXISTS) ... 915
4.16.2.3 Mazání tabulek křivek (CTABDEL) ... 915
4.16.2.4 Blokování tabulek křivek proti mazání a přepisování (CTABLOCK, CTABUNLOCK) ... 916
4.16.2.5 Tabulky křivek: Zjišťování vlastností tabulek (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD) ... 918
4.16.2.6 Čtení hodnot z tabulky křivky (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) ... 919
4.16.2.7 Tabulky křivek: Kontrola využití systémových zdrojů (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) ... 923
4.16.3 Osová vazba řídící hodnotou (LEADON, LEADOF) ... 925
4.16.4 Elektronická převodovka (EG) ... 930
4.16.4.1 Definice elektronické převodovky (EGDEF) ... 931
4.16.4.2 Zapnutí elektronické převodovky (EGON, EGONSYN, EGONSYNE) ... 932
4.16.4.3 Vypnutí elektronické převodovky (EGOFS, EGOFC) ... 935
4.16.4.4 Vymazání definice elektronické převodovky (EGDEL) ... 936
4.16.4.5 Otáčkový posuv (G95) / elektronická převodovka (FPR) ... 936
4.16.5 Synchronní vřeteno ... 936
4.16.5.1 Synchronní vřeteno: Programové příkazy (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC) ... 937
4.16.6 Generická vazba (CP...) ... 947
4.16.7 Tangenciální řízení ... 955
4.16.7.1 Definice vazby (TANG) ... 955
4.16.7.2 Aktivování vytváření pomocných bloků (TLIFT) ... 956
4.16.7.3 Aktivování vazby (TANGON) ... 957
4.16.7.4 Deaktivování vazby (TANGOF) ... 959
4.16.7.5 Vymazání vazby (TANGDEL) ... 959
4.16.8 Vazba Master/Slave (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS): ... 961

4.17 Synchronní akce ... 964
4.17.1 Definice synchronní akce ... 964

4.18 Pohyb tam a zpět ... 964
4.18.1 Asynchronní oscilační pohyby (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) ... 964
4.18.2 Oscilační pohyby řízené prostřednictvím synchronních akcí (OSCILL) ... 969

4.19 Lisování a prostřihování ... 976
4.19.1 Aktivování/deaktivování ... 976
4.19.1.1 Aktivování/deaktivování lisování a prostřihování (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) ... 976
4.19.2 Automatické rozdělení dráhy ... 981
4.19.2.1 Rozdělení cesty u dráhových os ... 983
4.19.2.2 Rozdělení cesty u jednotlivých os ... 985

4.20 Broušení ... 986

4.20.1 Aktivování/deaktivování speciálního monitorování pro brusné nástroje (TMON, TMOF) 986

4.21 Rozšířené zastavování a odjíždění (ESR) 987
4.21.1 ESR řízené NC systémem 988
4.21.1.1 Odjíždění řízené NC systémem (POLF, POLFA, POLFMASK, POLFMLIN) 988
4.21.1.2 Zastavení NC systémem 992
4.21.2 ESR nezávislé na pohonu 992
4.21.2.1 Konfigurace zastavování nezávislého na pohonu (ESRS) 992
4.21.2.2 Konfigurace odjíždění nezávislého na pohonu (ESRR) 993

4.22 Doba zpracování programu / počítadlo obrobků 994
4.22.1 Doba zpracování programu 994
4.22.2 Počítadlo obrobků 998

4.23 Simulace programu 999
4.23.1 Funkce 999
4.23.2 Simulace programu ve spojení s cykly překladače 999

4.24 Další funkce 1000
4.24.1 Aktivování strojních parametrů (NEWCONF) 1000
4.24.2 Kontrola existujícího rozsahu NC jazyka (STRINGIS) 1001
4.24.3 Interaktivní vyvolávání oken z výrobního programu (MMC) 1005
4.24.4 Process DataShare - Výstup do externího zařízení/souboru (EXTOPEN, WRITE, EXTCLOSE). 1010
4.24.5 Alarmy (SETAL) 1014
4.24.6 Definice surového obrobku (WORKPIECE) 1016
4.24.7 Přepnutí režimu jazyka (G290, G291) 1020

4.25 Vlastní programy pro oddělování třísky 1022
4.25.1 Podporované funkce pro oddělování třísky 1022
4.25.2 Sestavování kontury (CONTPRON) 1023
4.25.3 Sestavování kódované tabulky kontury (CONTPRON) 1028
4.25.4 Zjištění průsečíku mezi dvěma konturovými prvky (INTERSEC) 1032
4.25.5 Pohyb po konturových prvcích v tabulce blok po bloku (EXECTAB) 1033
4.25.6 Výpočet parametrů kruhu (CALCDAT) 1034
4.25.7 Ukončení přípravy kontury (EXECUTE) 1036

4.26 Programování externích cyklů 1036
4.26.1 Technologické cykly 1036
4.26.1.1 Úvod 1036
4.26.1.2 Přehled pro jednotlivé technologie 1038
4.26.1.3 HOLES1 - Polohovací vzor Přímka 1039
4.26.1.4 HOLES2 - Polohovací vzor Kružnice nebo Kruhový oblouk 1040
4.26.1.5 POCKET3 - Pravoúhlá kapsa 1042
4.26.1.6 POCKET4 - Kruhová kapsa 1045
4.26.1.7 SLOT1 - Podélná drážka 1048
4.26.1.8 SLOT2 - Kruhová drážka 1050
4.26.1.9 LONGHOLE - Podlouhlá díra 1053
4.26.1.10 CYCLE60 - Gravírování 1055
4.26.1.11 CYCLE61 - Rovinné frézování 1057
4.26.1.12 CYCLE62 - volání kontury 1059
4.26.1.13 CYCLE63 - Frézování konturové kapsy / odstranění zbytkového materiálu z konturové kapsy / frézování konturového čepu / odstranění zbytkového materiálu z konturového čepu 1060
4.26.1.14 CYCLE64 - Předvrtání konturové kapsy 1062
4.26.1.15 CYCLE70 - Frézování závitu 1064

4.26.1.16 CYCLE72 - Frézování po dráze ... 1066
4.26.1.17 CYCLE76 - Pravoúhlý čep ... 1070
4.26.1.18 CYCLE77 - Kruhový čep ... 1072
4.26.1.19 CYCLE78 - Frézování vrtaných závitů ... 1074
4.26.1.20 CYCLE79 - Mnohohran ... 1076
4.26.1.21 CYCLE81 - Vrtání, navrtávání středicích důlků ... 1078
4.26.1.22 CYCLE82 - Vrtání, čelní zahlubování ... 1079
4.26.1.23 CYCLE83 - Vrtání hlubokých děr 1 ... 1082
4.26.1.24 CYCLE84 - Vrtání závitů bez vyrovnávací hlavičky ... 1085
4.26.1.25 CYCLE85 - Vystružování ... 1088
4.26.1.26 CYCLE86 - Vyvrtávání ... 1089
4.26.1.27 CYCLE92 - Upichování ... 1091
4.26.1.28 CYCLE95 - Oddělování třísky od kontury ... 1092
4.26.1.29 CYCLE98 - Řetězec závitů ... 1094
4.26.1.30 CYCLE99 - Soustružení závitu ... 1098
4.26.1.31 CYCLE435 - Nastavení souřadného systému orovnávače ... 1103
4.26.1.32 CYCLE495 - Profilování ... 1103
4.26.1.33 CYCLE782 - Přizpůsobení zatížení ... 1105
4.26.1.34 CYCLE800 - Naklápění roviny / naklápění nástroje / nastavování orientace nástroje ... 1108
4.26.1.35 CYCLE801 - Polohovací vzor Mřížka nebo Obdélník ... 1111
4.26.1.36 CYCLE802 - Libovolné pozice ... 1113
4.26.1.37 CYCLE806 - Interpolační soustružení ... 1116
4.26.1.38 CYCLE830 - Vrtání hlubokých děr 2 ... 1117
4.26.1.39 CYCLE832 - Vysokorychlostní obrábění ... 1123
4.26.1.40 CYCLE840 - Vrtání závitů s vyrovnávací hlavičkou ... 1127
4.26.1.41 CYCLE899 - Otevřená drážka ... 1129
4.26.1.42 CYCLE930 - Zápich ... 1132
4.26.1.43 CYCLE940 - Odlehčovací zápich tvaru E a F / závitový zápich ... 1134
4.26.1.44 CYCLE951 - Oddělování třísky ... 1137
4.26.1.45 CYCLE952 - Oddělování třísky / odstraňování zbytkového materiálu oddělováním třísky / zapichování / zapichování zbytkového materiálu / soustružení zápichu / odstraňování zbytkového materiálu soustružením zápichu ... 1140
4.26.1.46 CYCLE4071 - Podélné broušení s přísuvem v bodu obratu ... 1146
4.26.1.47 CYCLE4072 - Podélné broušení s přísuvem v bodě obratu a se signálem pro přerušení ... 1148
4.26.1.48 CYCLE4073 - Podélné broušení se spojitým přísuvem ... 1152
4.26.1.49 CYCLE4074 - Podélné broušení se spojitým přísuvem a se signálem pro přerušení ... 1153
4.26.1.50 CYCLE4075 - Plošné broušení s přísuvem v bodě obratu ... 1156
4.26.1.51 CYCLE4077 - Plošné broušení s přísuvem na bod obratu a se signálem pro přerušení ... 1159
4.26.1.52 CYCLE4078 - Plošné broušení se spojitým přísuvem ... 1163
4.26.1.53 CYCLE4079 - Plošné broušení s přerušovaným přísuvem ... 1165
4.26.1.54 GROUP_BEGIN - Začátek programového bloku ... 1167
4.26.1.55 GROUP_END - konec programového bloku ... 1168
4.26.1.56 GROUP_ADDEND - konec doplňkového bloku najíždění ... 1168
4.26.1.57 Okrajové podmínky ... 1169
4.26.2 Měřicí cykly ... 1170
4.26.2.1 Přehled parametrů měřicích cyklů ... 1170

5 Tabulky ... 1213

5.1 Příkazy ... 1213

5.2 Adresy ... 1251
5.2.1 Adresová písmena ... 1251
5.2.2 Pevné adresy ... 1252

5.2.3 Nastavitelné adresy ... 1257

5.3 G-příkazy ... 1263
5.3.1 G-příkazy ... 1263
5.3.2 G-skupina 1: Příkazy pohybu s modální platností ... 1263
5.3.3 G-skupina 2: Pohyby s blokovou platností, doba prodlevy ... 1264
5.3.4 G-skupina 3: Programovatelné framy, omezení pracovního pole a programování pólu ... 1264
5.3.5 G-skupina 4: FIFO ... 1265
5.3.6 G-skupina 6: Volba roviny ... 1265
5.3.7 G-skupina 7: Korekce rádiusu nástroje ... 1266
5.3.8 G-skupina 8: Nastavitelné posunutí počátku ... 1266
5.3.9 G-skupina 9: Potlačení framových a nástrojových korekcí ... 1266
5.3.10 G-skupina 10 Přesné najetí – režim řízení pohybu po dráze ... 1267
5.3.11 G-skupina 11: Blokové přesné najetí ... 1267
5.3.12 G-skupina 12: Kritéria přechodu na další blok při přesném najetí (G60/G9) ... 1267
5.3.13 G-skupina 13: Kótování obrobku v palcích/metrických jednotkách ... 1267
5.3.14 G-skupina 14: Kótování obrobku absolutní/inkrementální ... 1268
5.3.15 G-skupina 15: Typ posuvu ... 1268
5.3.16 G-skupina 16: Korekce posuvu (override) na vnitřních a vnějších zakřiveních ... 1269
5.3.17 G-skupina 17: Chování při najíždění/odjíždění s korekcí rádiusu nástroje ... 1269
5.3.18 G-skupina 18: Chování korekce nástroje v rozích ... 1269
5.3.19 G-skupina 19: Křivkový přechod na začátku splinu ... 1269
5.3.20 G-skupina 20: Křivkový přechod na konci splinu ... 1270
5.3.21 G-skupina 21: Profil zrychlení ... 1270
5.3.22 G-skupina 22: Typ korekce nástroje ... 1270
5.3.23 G-skupina 23: Monitorování možnosti kolize na vnitřních konturách ... 1271
5.3.24 G-skupina 24: Dopředná regulace ... 1271
5.3.25 G-skupina 25: Reference orientace nástroje ... 1271
5.3.26 G-skupina 26: Režim opětovného najíždění pro funkci REPOS (s modální platností) ... 1272
5.3.27 G-skupina 27: Korekce nástroje při změnách orientace na vnějších rozích ... 1272
5.3.28 G-skupina 28: Ohraničení pracovního pole ... 1272
5.3.29 G-skupina 29: Programování rádiusů/průměrů ... 1272
5.3.30 G-skupina 30: Komprese NC bloků ... 1273
5.3.31 G-skupina 31: G-příkazy pro OEM ... 1273
5.3.32 G-skupina 32: G-příkazy pro OEM ... 1274
5.3.33 G-skupina 33: Nastavitelná jemná korekce nástroje ... 1274
5.3.34 G-skupina 34: Vyhlazení orientace nástroje ... 1274
5.3.35 G-skupina 35: Lisování a prostřihování ... 1275
5.3.36 G-skupina 36: Lisování se zpožděním ... 1275
5.3.37 G-skupina 37: Profil posuvu ... 1275
5.3.38 G-skupina 38: Přiřazení rychlých vstupů/výstupů pro lisování/prostřihování ... 1275
5.3.39 G-skupina 39: Programovatelná přesnost kontury ... 1276
5.3.40 G-skupina 40: Konstantní korekce rádiusu nástroje ... 1276
5.3.41 G-skupina 41: Možnost přerušení řezání závitu ... 1276
5.3.42 G-skupina 42: Držák nástroje ... 1276
5.3.43 G-skupina 43: Směr najíždění WAB ... 1277
5.3.44 G-skupina 44: Rozdělení dráhy WAB ... 1277
5.3.45 G-skupina 45: Vztah dráhy a os v FGROUP ... 1277
5.3.46 G-skupina 46: Volba roviny pro rychlé pozvednutí ... 1277
5.3.47 G-skupina 47: Přepínání režimu pro externí NC-kód ... 1278
5.3.48 G-skupina 48: Chování při najíždění/odjíždění s korekcí rádiusu nástroje ... 1278
5.3.49 G-skupina 49: Pohyb od bodu k bodu ... 1278
5.3.50 G-skupina 50: Programování orientace ... 1278

5.3.51 G-skupina 51: Programování druhu interpolace orientace ... 1279
5.3.52 G-skupina 52: Otáčení framu vztahující se na obrobek ... 1279
5.3.53 G-skupina 53: Otáčení framu vztahující se na nástroj ... 1280
5.3.54 G-skupina 54: Otáčení vektoru při programování polynomu ... 1280
5.3.55 G-skupina 55: Rychlý posuv s/bez lineární interpolace ... 1280
5.3.56 G-skupina 56: Započítávání hodnot opotřebení nástroje ... 1281
5.3.57 G-skupina 57: Zpoždění v rozích ... 1281
5.3.58 G-skupina 59: Režim dynamiky pro dráhovou interpolaci ... 1281
5.3.59 G-skupina 60: Ohraničení pracovního pole ... 1282
5.3.60 G-skupina 61: Vyhlazení orientace nástroje ... 1282
5.3.61 G-skupina 62: Režim opětovného najíždění pro funkci REPOS (s blokovou platností) ... 1282
5.3.62 G-skupina 64: Frame pro broušení ... 1283
5.4 Předem definované procedury ... 1284
5.5 Předdefinované procedury v synchronních akcích ... 1306
5.6 Předem definované funkce ... 1307
5.7 Aktuální jazyk v HMI ... 1320
A Přílohy ... 1321
A.1 Seznam zkratek ... 1321
Rejstřík ... 1327

Úvod 1

1.1 Prostřednictvím systému SINUMERIK

Od jednoduchých standardních CNC strojů, přes standardizované stroje až po modulární strojní koncepce nejvyšší kategorie - řídící CNC systémy SINUMERIK nabízejí pro každou koncepci stroje odpovídající řešení. Ať už jde o kusovou nebo hromadnou výrobu, jednoduché nebo složité obrobky - SINUMERIK je vysoce produktivní automatizační řešení vhodné pro všechny výrobní oblasti - od konstrukce vzorů a nástrojů, přes výrobu forem až po výrobu velkých sérií.

Budete-li potřebovat další informace, navštivte internetové stránky věnované systému SINUMERIK (https://www.siemens.com/sinumerik).

1.2 O této dokumentaci

Předkládaná dokumentace patří do skupiny programovacích příruček k systému SINUMERIK.

Programovací příručka „Programování NC systémů“

Programovací příručka „Programování NC systémů“ obsahuje všechny informace, které jsou důležité pro programování funkcí řídícího NC systému SINUMERIK.

Cílovou skupinou jsou programátoři a projektanti.

Pomocí této programovací příručky mohou pracovníci cílové skupiny vyvíjet, psát, testovat a odstraňovat chyby v programech a v obrazovkách uživatelského rozhraní.

Přehled obsahu

Z přehledu obsahu na titulní stránce je patrné, že programovací příručka „Programování NC systémů“ je rozdělena na dvě hlavní kapitoly:

1. **Základy**
   Hlavní kapitola "Základy" je určena pro zkušené kvalifikované pracovníky obsluhy stroje a předpokládá odpovídající znalosti pro operace vrtání, frézování a soustružení. Pro vysvětlení příkazů a výrazů, které jsou definovány rovněž podle normy DIN 66025, se používají jednoduché příklady programování.
2. **Pro pokročilé**
   Hlavní kapitola „Pro pokročilé“ slouží technologům, kteří disponují znalostmi o všech možnostech programování. Řídící systémy SINUMERIK umožňují pomocí speciálního programovacího jazyka vytváření programů pro výrobu složitých obrobků (např. modelované povrchy volných tvarů, koordinace kanálů atd.) a technologům výrazně usnadňuje programování složitých operací.

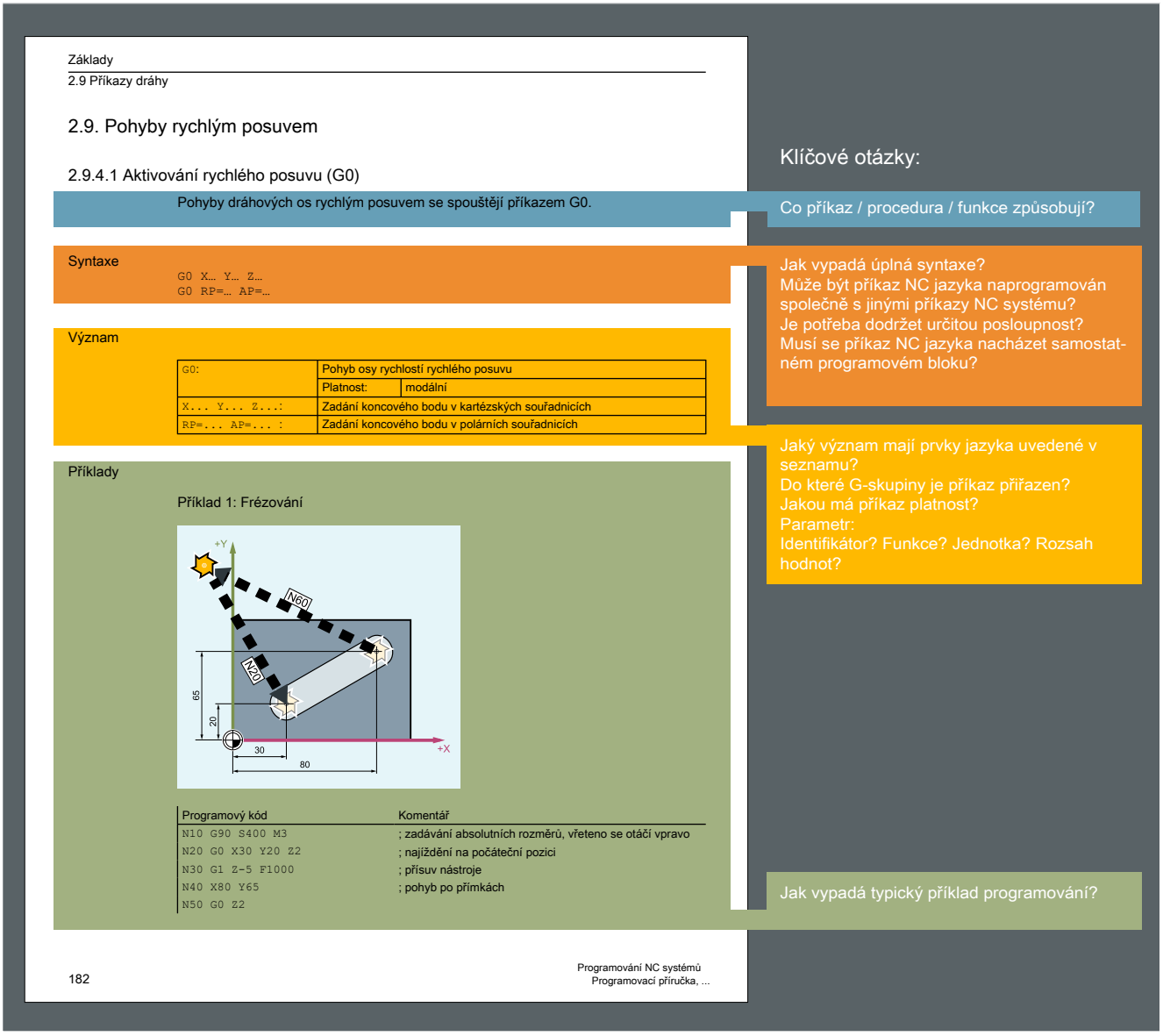
Platnost

Titulní strana obsahuje také všechny údaje týkající se platnosti dokumentů, tzn. pro který řídící systém SINUMERIK a pro kterou verzi programového vybavení toto vydání programovací příručky platí.

Struktura kapitol

Popisy prvků jazyka NC-systému (příkaz G-funkce, procedura, funkce, ...) mají jednotnou strukturu a při sestavování obsahu jsou dodržovány předem definované klíčové otázky.

Tato struktura je zřejmá z příkladu kapitoly na následujícím obrázku:



Základy
2.9 Příkazy dráhy

2.9. Pohyby rychlým posuvem

2.9.4.1 Aktivování rychlého posuvu (G0)

Pohyby dráhových os rychlým posuvem se spouštějí příkazem G0.

Syntaxe

```
G0 X... Y... Z...
G0 RP=... AP=...
```

Význam

G0:	Pohyb osy rychlostí rychlého posuvu	
	Platnost:	modální
X... Y... Z...:	Zadání koncového bodu v kartézských souřadnicích	
RP=... AP=... :	Zadání koncového bodu v polárních souřadnicích	

Příklady

Příklad 1: Frézování

Programový kód	Komentář
N10 G90 S400 M3	; zadávání absolutních rozměrů, vřeteno se otáčí vpravo
N20 G0 X30 Y20 Z2	; najíždění na počáteční pozici
N30 G1 Z-5 F1000	; přísuv nástroje
N40 X80 Y65	; pohyb po přímkách
N50 G0 Z2	

182

Programování NC systémů
Programovací příručka, ...

Klíčové otázky:

Co příkaz / procedura / funkce způsobují?

Jak vypadá úplná syntaxe?
Může být příkaz NC jazyka naprogramován společně s jinými příkazy NC systému?
Je potřeba dodržet určitou posloupnost?
Musí se příkaz NC jazyka nacházet samostatném programovém bloku?

Jaký význam mají prvky jazyka uvedené v seznamu?
Do které G-skupiny je příkaz přiřazen?
Jakou má příkaz platnost?
Parametr:
Identifikátor? Funkce? Jednotka? Rozsah hodnot?

Jak vypadá typický příklad programování?

Kromě v příkladu uvedených a vždy se vyskytujících částí Syntaxe, Význam a Příklad(y) se mohou v případě potřeby objevit ještě další položky:

Doplňkové položky	Použití
Předpoklady	Jestliže pro použití programovatelné funkce musí být splněny určité předpoklady (např. instalovaná licence k volitelnému doplňku).
Platnost	Jestliže je programovatelná funkce v platnosti pouze za určitých podmínek (např. jen v určitém provozním režimu).
Okrajové podmínky	Jestliže je potřeba mít při programování na paměti vzájemné ovlivňování s jinými funkcemi.
Další informace	Pokud jsou zapotřebí další vysvětlení, aby bylo programování funkce chápáno správně.

Standardní rozsah

V předkládané dokumentaci jsou popisovány funkce standardního rozsahu dodávky. Ten se může od rozsahu funkcí dodávaného systému lišit. Pro funkce dodaného systému jsou určující výlučně podklady v objednávce.

V systému se mohou vyskytovat i další funkce nepopsané v rámci této dokumentace, které lze spustit. S ohledem na tyto funkce však není možné vznést žádný nárok pro případ nové dodávky nebo servisního zásahu.

Z důvodů zachování přehlednosti nemůže tato dokumentace obsahovat všechny podrobné informace ke všem typům produktu. Kromě toho také nemůže tato dokumentace pokrýt veškeré možné případy, které se mohou v průběhu instalace, provozování a údržby vyskytnout.

Doplnění nebo změny produktu, které byly provedeny výrobcem stroje, jsou popsány v dokumentaci od tohoto výrobce stroje.

Internetové stránky třetích stran

Tento dokument může obsahovat odkazy na internetové stránky třetích stran. Firma Siemens nenese za obsah těchto webových stránek odpovědnost a ani nepovažuje tyto webové stránky a jejich obsah za vlastní. Firma Siemens informace na těchto webových stránkách nekontroluje a také za jejich obsah a informace není zodpovědná. Riziko vyplývající z jejich použití nese uživatel.

1.3 Dokumentace na internetu

1.3.1 Přehled dokumentace systému SINUMERIK 840D sl

Obsáhlou dokumentaci k funkcím systému SINUMERIK 840D sl od verze 4.8 SP4 naleznete prostřednictvím odkazu Přehled dokumentace systému 840D sl (https://support.industry.siemens.com/cs/ww/en/view/109766213).

Zde máte možnost si požadované dokumenty zobrazit nebo si je stáhnout ve formátu PDF a HTML5.

Dokumentace je rozčleněna do následujících kategorií:

- Uživatel: Obsluha
- Uživatel: Programování
- Výrobce/servis: Funkce
- Výrobce/servis: Hardware
- Výrobce/servis: Projekce/Uvedení do provozu
- Výrobce/servis: Safety Integrated
- Výrobce/servis: SINUMERIK Integrate / MindApp
- Informace a školení
- Výrobce/servis: SINAMICS

1.3.2 Přehled dokumentace k ovládacím prvkům systému SINUMERIK

Obsáhlou dokumentaci k ovládacím prvkům systému SINUMERIK naleznete prostřednictvím odkazu Přehled dokumentace k ovládacím prvkům systému SINUMERIK (https://support.industry.siemens.com/cs/document/109783841/technische-dokumentation-zu-sinumerik-bedienkomponenten?dti=0&lc=en-WW).

Zde máte možnost si požadované dokumenty zobrazit nebo si je stáhnout ve formátu PDF a HTML5.

Dokumentace je rozčleněna do následujících kategorií:

- Ovládací panely
- Ovládací panely stroje
- Panel s tlačítky pro ovládání stroje
- Jednotka ručního ovladače/malé ruční ovladače
- Další ovládací prvky

Přehled důležitých dokumentů, příspěvků a odkazy k tématu „SINUMERIK" naleznete na stránce SINUMERIK - stránka s přehledem témat (https://support.industry.siemens.com/cs/document/109766201/sinumerik-an-overview-of-the-most-important-documents-and-links?dti=0&lc=en-WW).

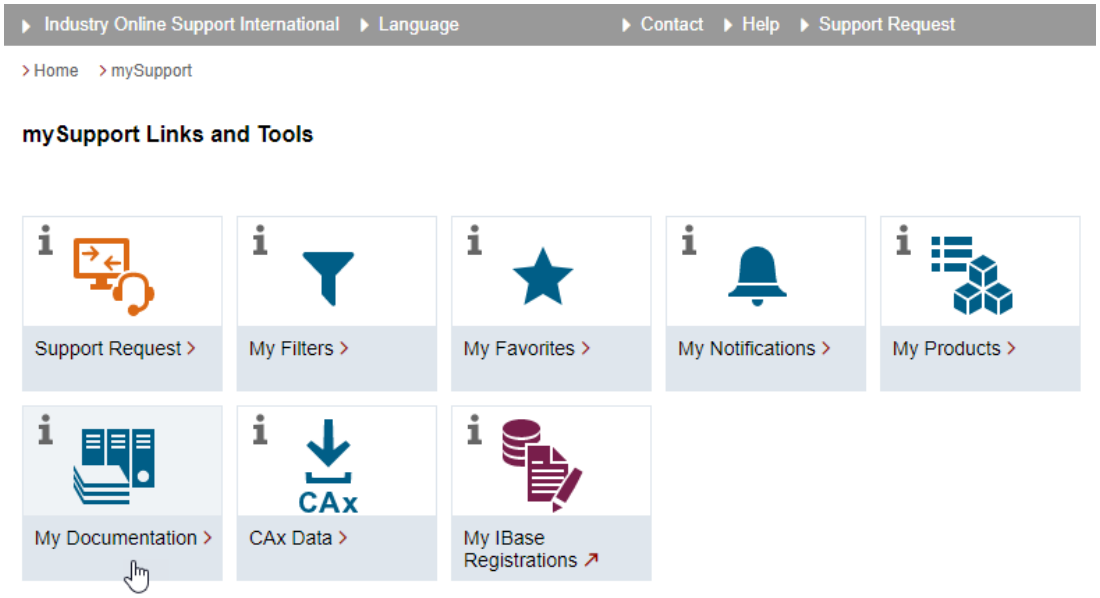
1.4 Zpětná vazba na technickou dokumentaci

Budete-li mít jakékoli dotazy, podněty nebo korekce týkající se technické dokumentace zveřejňované firmou Siemens Industry Online Support, použijte odkaz „Odeslat zpětnou vazbu" na konci příspěvku.

1.5 mySupport-Dokumentation

Pomocí webového systému „mySupport-Dokumentation" můžete svou dokumentaci individuálně sestavovat na základě dokumentů firmy Siemens a přizpůsobovat ji, aby odpovídala dokumentaci vašeho vlastního stroje.

Aplikaci spusťte pomocí tlačítka „Má dokumentace" na úvodní obrazovce portálu mySupport (https://support.industry.siemens.com/cs/ww/en/my):



Sestavené příručky je možno exportovat ve formátu RTF, PDF nebo XML.

Poznámka

Dokumenty od firmy Siemens, které podporují aplikaci mySupport-Dokumentation, poznáte podle toho, že je přítomen odkaz „Konfigurovat“.

1.6 Servis a podpora

Podpora pro produkt

Další informace o produktu naleznete na internetu:

Podpora pro produkt (https://support.industry.siemens.com/cs/ww/en/)

Na této adrese naleznete následující:

- Aktuální informace o produktu (oznámení vztahující se k produktu)
- FAQ (často kladené otázky)
- Příručky
- Ke stažení
- Bulletin s nejnovějšími informacemi k vašim produktům
- Fórum pro uživatele a odborníky z celého světa sloužící pro výměnu informací a zkušeností
- Kontaktní osoby pro danou oblast prostřednictvím databáze kontaktních osob (→ „Kontakt“)
- Informace o servisu, opravách a náhradních dílech pro danou oblast a mnoho dalšího (→ „Služby“)

Technická podpora

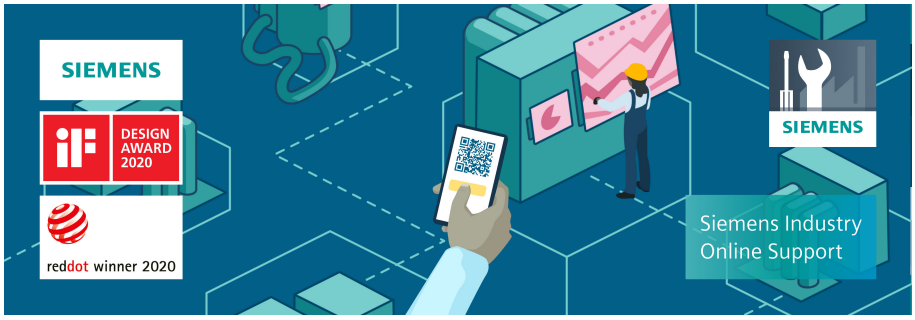
Specifická telefonní čísla na pracovníky technické podpory v dané zemi naleznete na internetu na adrese (https://support.industry.siemens.com/cs/ww/en/sc/4868) v oblasti „Kontakt“.

Jestliže máte nějaké technické dotazy, použijte on-line formulář v oblasti „Support Request“ (Žádost o podporu).

Vzdělávání

Na následující adrese (https://www.siemens.com/sitrain) naleznete informace o kurzech SITRAIN.
Portál SITRAIN představuje nabídku školení pro produkty, systémy a řešení firmy Siemens z oblasti pohonů a automatizační techniky.

On-line podpora firmy Siemens



Pomocí oceňované aplikace „Siemens Industry Online Support" máte kdykoli a kdekoli přístup k více než 300 000 dokumentů k produktům firmy Siemens Industry. Tato aplikace vám mimo jiné pomáhá v následujících oblastech:

- Řešení problémů při realizaci projektů
- Odstraňování chyb v případě poruchy
- Rozšíření nebo plánování nového zařízení

Kromě toho máte přístup k sekci Technické fórum a k dalším příspěvkům, které jsou pro vás připravovány našimi odborníky:

- Často kladené otázky
- Příklady použití
- Příručky
- Certifikáty
- Zprávy o produktech a mnoho dalšího

Aplikace „Siemens Industry Online Support" je k dispozici pro Apple iOS a Android.

QR kód s údaji na výrobním štítku

QR kód s údaji na výrobním štítku obsahuje specifické údaje o daném zařízení. Tento kód může být načten kterýmkoli chytrým telefonem a potom prostřednictvím mobilní aplikace „Industry Online Support" je možné zobrazit technické informace k odpovídajícímu zařízení.

1.7 Důležité informace o produktu

Použití OpenSSL

Tento produkt může obsahovat následující programové vybavení:

- Software, který byl vyvinut prostřednictvím projektu OpenSSL pro použití v rámci aplikace OpenSSL-Toolkit
- Kryptografický software, který vytvořil Eric Young
- Software, který vyvinul Eric Young

Další informace naleznete na internetu:

- OpenSSL (https://www.openssl.org)
- Cryptsoft (https://www.cryptsoft.com)

Dodržování hlavní směrnice pro ochranu dat

Firma Siemens dodržuje zásady ochrany dat, zejména nařízení týkající se minimalizace dat (privacy by design).

Pro tento produkt to znamená:

Produkt nezpracovává a ani neukládá žádná data vztahující se ke konkrétní osobě, pouze technická funkční data (jako je např. časová značka). Jestliže uživatel tato data spojí s jinými daty (např. s plány směn) nebo pokud uloží data vztahující se k nějaké osobě na stejné médium (např. na pevný disk) a vytvoří tak vztah s určitou osobou, musí sám zajistit, že právní předpisy pro ochranu dat nebudou porušeny.

Základní bezpečnostní upozornění 2

2.1 Všeobecná bezpečnostní upozornění

⚠ VÝSTRAHA

Ohrožení života v případě zanedbání bezpečnostních upozornění a zbývajících rizik

V případě zanedbání bezpečnostních upozornění a zbývajících rizik, která jsou uvedena v odpovídající dokumentaci k hardwaru, může dojít k nehodám s těžkými nebo i smrtelnými úrazy.

- Dodržujte bezpečnostní upozornění v dokumentaci k hardwaru.
- Mějte na paměti zbývající rizika, která byla zjištěna při jejich vyhodnocování.

⚠ VÝSTRAHA

Nesprávná funkce stroje v důsledku chybného nebo změněného nastavení parametrů

V důsledku chybného nebo změněného nastavení parametrů se může vyskytnout chybná funkce stroje, která může mít za následek úraz nebo i smrt.

- Chraňte nastavení parametrů před přístupem nepovolaných osob.
- Pro případ možné nesprávné funkce zajistěte vhodná opatření, např. spínače pro nouzové zastavení nebo nouzové vypínače.

2.2 Záruční podmínky pro příklady aplikace

Příklady aplikací jsou nezávazné a nekladou si žádné nároky na úplnost, ani pokud jde o konfiguraci a vybavení, ani pokud jde o všechny možné eventuality. Příklady aplikací nepřestavují žádná specifická zákaznická řešení, mají pouze nabízet určitou pomoc při řešení typických problémů.

Jako uživatel nesete odpovědnost za odborné zacházení s popisovanými produkty. Příklady aplikací vás nezbavují povinnosti zajistit bezpečné zacházení při použití, instalaci, provozu a údržbě.

2.3 Bezpečnostní pokyny

Siemens nabízí produkty a řešení s funkcemi Industrial Security, které podporují bezpečný provoz zařízení, systémů, strojů a sítí.

Pro zabezpečení zařízení, systémů, strojů a sítí proti kybernetickým útokům je třeba implementovat komplexní koncept Industrial Security (a následně udržovat), který odpovídá

aktuálnímu stavu techniky. Produkty a řešení společnosti Siemens představují jednu ze součástí takového konceptu.

Zákazníci odpovídají za to, aby zabránili nepovolanému přístupu ke svým zařízením, systémům, strojům a sítím. Tyto systémy, stroje a komponenty mají být připojeny do podnikové sítě nebo sítě Internet jen tehdy, je-li to nutné a pokud byla učiněna odpovídající ochranná opatření (např. firewally a/nebo segmentace sítě).

Další informace ohledně možných ochranných opatření v oblasti Industrial Security naleznete na:
https://www.siemens.com/industrialsecurity (https://www.siemens.com/industrialsecurity)

Produkty a řešení Siemens jsou neustále vyvíjeny, aby byly ještě bezpečnější. Siemens výslovně doporučuje používat produktové updaty, jakmile jsou k dispozici, a vždy používat aktuální verze programu. Používání zastaralých nebo již nepodporovaných verzí může zvyšovat riziko kybernetických útoků.

Abyste byli vždy informováni o aktualizacích produktů, objednejte si zprávy RSS Siemens o průmyslové bezpečnosti na:
https://www.siemens.com/industrialsecurity (https://new.siemens.com/global/en/products/services/cert.html#Subscriptions)

Další informace naleznete na internetu:

Příručka pro návrh konstrukce Industrial Security (https://support.industry.siemens.com/cs/ww/en/view/108862708)

⚠ VÝSTRAHA

Nejisté provozní stavy způsobené manipulací s programovým vybavením

Manipulace s programovým vybavením, např. viry, trojské koně nebo červi, mohou způsobit nejisté provozní stavy vašeho zařízení, které mohou mít za následek smrt, těžké ublížení na zdraví a materiální škody.

- Provádějte pravidelné aktualizace softwaru.
- Komponenty automatizačního a pohonového systému integrujte do celkové koncepce Industrial Security zařízení nebo stroje v souladu s nejmodernějšími technickými poznatky.
- Do své celkové koncepce Industrial Security zahrňte všechny použité produkty.
- Prostřednictvím odpovídajících ochranných opatření, např. antivirovým programem, chraňte soubory na výměnných paměťových médiích před škodlivým softwarem.
- Na závěr procesu uvádění do provozu zkontrolujte veškerá nastavení, která se vztahují k zabezpečení.

Základy

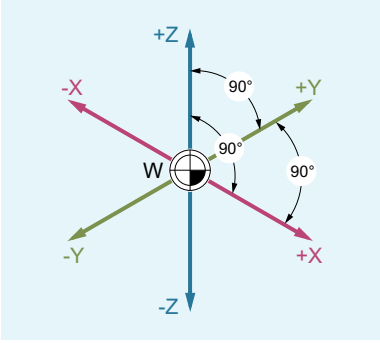
3

3.1 Geometrické základy

3.1.1 Pozice na obrobku

3.1.1.1 Vztažný systém pro údaje poloh

Aby stroj, případně řídící systém, mohl pracovat s pozicemi zadanými v NC-programu, musí být tyto údaje poloh uváděny ve vztažném systému, který je možno převést na směry pohybů jednotlivých os stroje. Pro tento účel se u obráběcích strojů používají pro popis obrobku kartézské, tzn. pravoúhlé pravotočivé souřadné systémy podle normy DIN 66217.



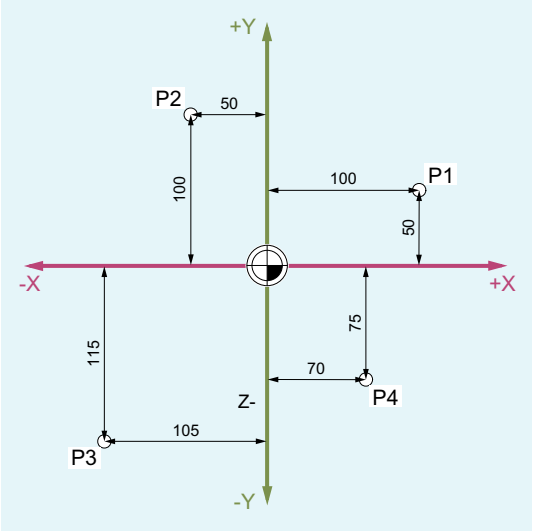
Počátek (nula) souřadného systému obrobku (W) odpovídá počátku souřadného systému obrobku.

3.1.1.2 Kartézské souřadnice

Osám v souřadném systému jsou přiřazeny rozměrové jednotky. Díky tomu můžete každý bod v souřadném systému a tedy i každou polohu na obrobku jednoznačně popsat směrem (X, Y a Z) a číselnou hodnotou. Počátek souřadné soustavy obrobku má vždy souřadnice X0, Y0 a Z0.

Údaje polohy ve formě kartézských souřadnic

Pro zjednodušení v následujícím příkladu použijeme jen jednu rovinu souřadného systému, např. rovinu X/Y.

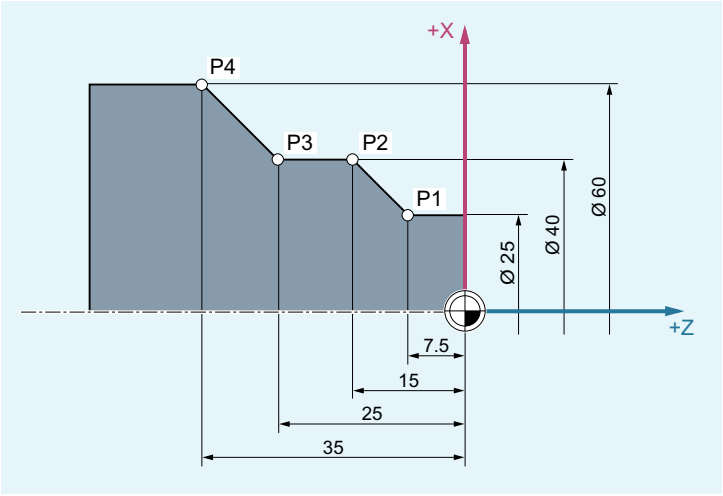


Body P1 až P4 mají následující souřadnice:

Pozice	souřadnice
P1	X100 Y50
P2	X-50 Y100
P3	X-105 Y-115
P4	X70 Y-75

Příklad: Polohy na obrobku při soustružení

U soustruhů postačuje pro popis kontury jedna rovina:

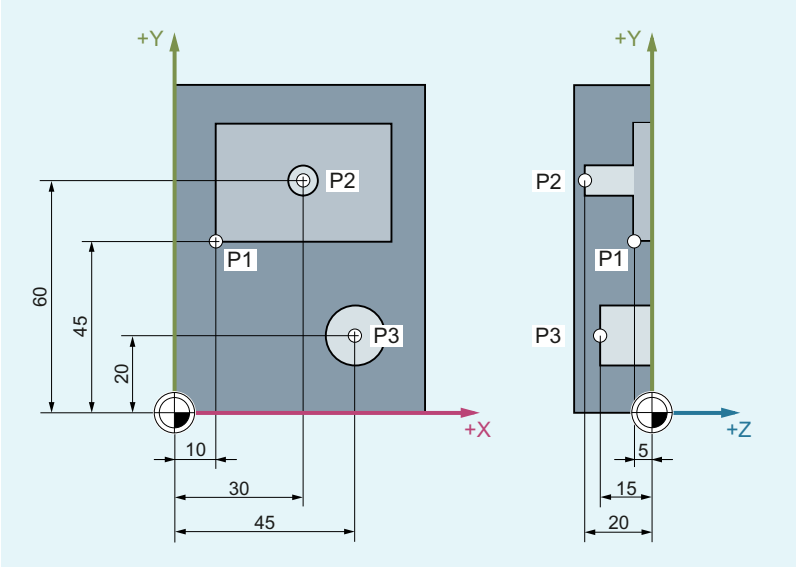


Body P1 až P4 mají následující souřadnice:

Pozice	souřadnice
P1	X25 Z-7.5
P2	X40 Z-15
P3	X40 Z-25
P4	X60 Z-35

Příklad: Polohy na obrobku při frézování

U frézovacích prací musí být popsána také přísuvná hloubka, tzn. je nutno přiřadit číselnou hodnotu také třetí souřadnici (v tomto případě souřadnici Z).



Body P1 až P3 mají následující souřadnice:

Pozice	souřadnice
P1	X10 Y45 Z-5
P2	X30 Y60 Z-20
P3	X45 Y20 Z-15

3.1.1.3 Polární souřadnice

Namísto kartézských souřadnic je možné pro popis poloh na obrobku používat také polární souřadnice. To má smysl tehdy, jestliže jsou obrobek nebo jeho části kótovány pomocí rádiusů a úhlů. Bod, od něhož kótování vychází, se nazývá "pól".

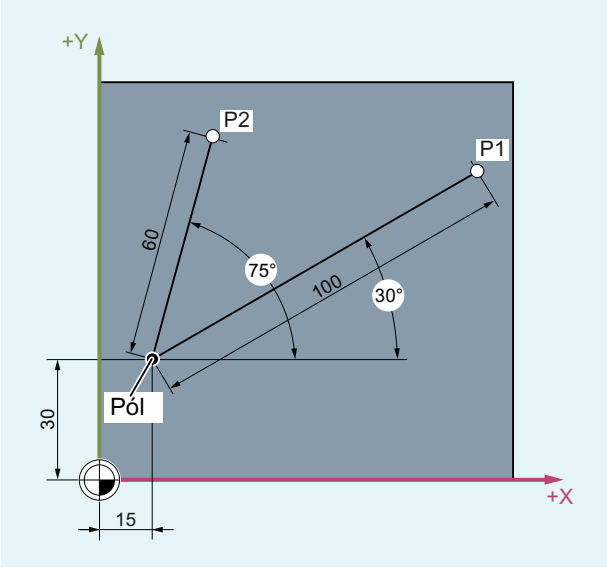
Údaje polohy ve formě polárních souřadnic

Polární souřadnice se skládají z **polárního rádiusu** a **polárního úhlu**.

Polární rádius je vzdálenost mezi pólem a danou polohou.

Polární úhel je úhel, který svírá polární rádius s vodorovnou osou pracovní roviny. Záporný polární úhel se měří ve směru hodinových ručiček, kladný úhel proti směru hodinových ručiček.

Příklad:



Body P1 a P2 je možné vzhledem k pólu popsat následujícím způsobem:

Pozice	Polární souřadnice
P1	RP=100 AP=30
P2	RP=60 AP=75
RP: Polární rádius AP: Polární úhel	

3.1.1.4 Absolutní rozměry

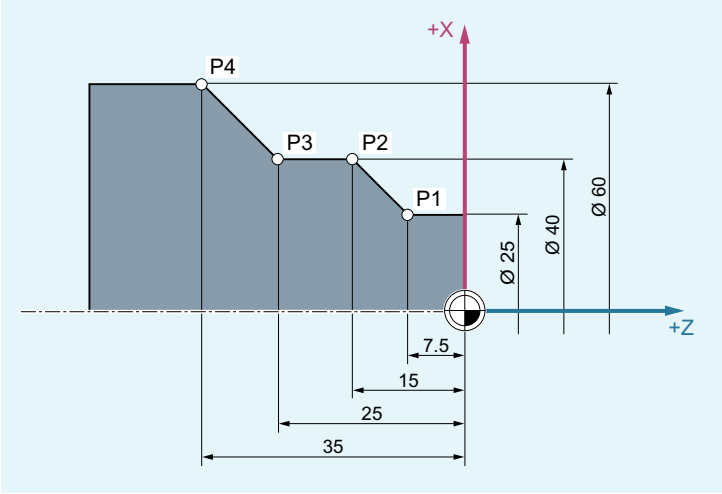
Údaje poloh při zadávání absolutních rozměrů

Při zadávání absolutních rozměrů jsou všechny údaje poloh vztaženy vždy na právě platný počátek.

S ohledem na pohyby nástroje to znamená:

Absolutní údaj polohy popisuje místo, na které má nástroj najet.

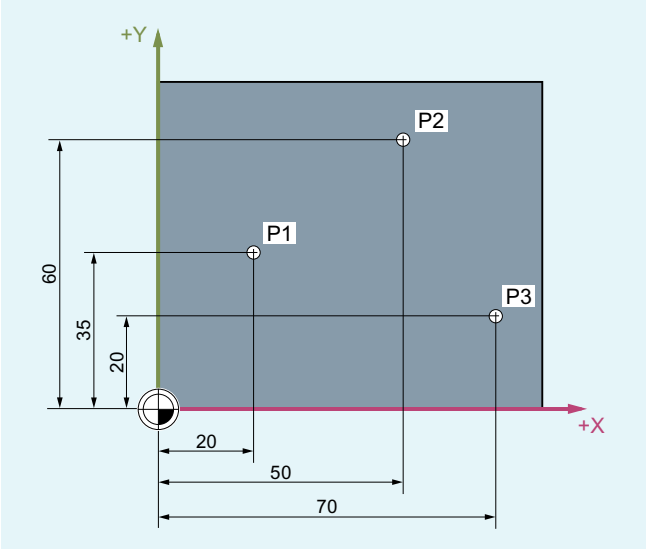
Příklad: Soustružení



V absolutních rozměrech vyplývají pro body P1 až P4 následující údaje poloh:

Pozice	Údaje poloh při zadávání absolutních rozměrů
P1	X25 Z-7,5
P2	X40 Z-15
P3	X40 Z-25
P4	X60 Z-35

Příklad: Frézování



V absolutních rozměrech vyplývají pro body P1 až P3 následující údaje poloh:

Pozice	Údaje poloh při zadávání absolutních rozměrů
P1	X20 Y35
P2	X50 Y60
P3	X70 Y20

3.1.1.5 Řetězové kótování

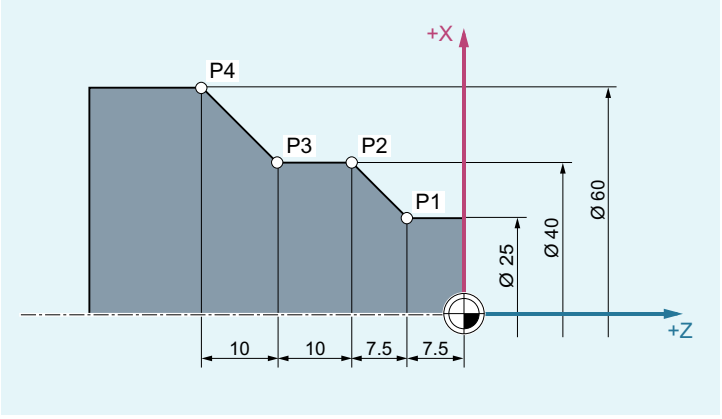
Údaje poloh v řetězových kótách (inkrementální kótování)

Ve výrobních výkresech se často stává, že rozměr není vztažen na počátek souřadného systému, nýbrž k jinému bodu na obrobku. Abyste nemuseli tyto rozměry přepočítávat, existuje možnost zadávání řetězových kót nebo inkrementálních rozměrů. Při tomto druhu zadávání rozměrů je údaj polohy vždy vztažen na předcházející bod.

S ohledem na pohyby nástroje to znamená:

Inkrementální rozměr udává, o kolik se má nástroj posunout.

Příklad: Soustružení



V inkrementálních rozměrech vyplývají pro body P2 až P4 následující údaje poloh:

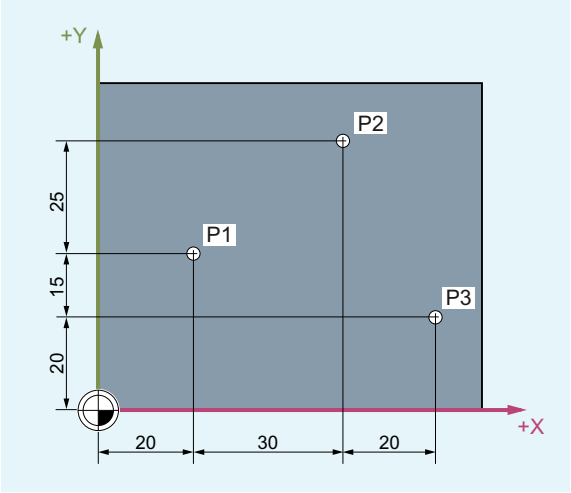
Pozice	Údaje poloh při zadávání inkrementálních rozměrů	Údaje jsou vztaženy na bod:
P2	X15 Z-7,5	P1
P3	Z-10	P2
P4	X20 Z-10	P3

Poznámka

Jestliže jsou aktivní funkce DIAMOF nebo DIAM90 (Strana 164), při zadávání inkrementálních rozměrů (G91) je požadovaný bod dráhy naprogramován jako údaj rádiusu.

Příklad: Frézování

Údaje poloh pro body P1 až P3 v případě řetězových kót znějí:



V inkrementálních rozměrech vyplývají pro body P1 až P3 následující údaje poloh:

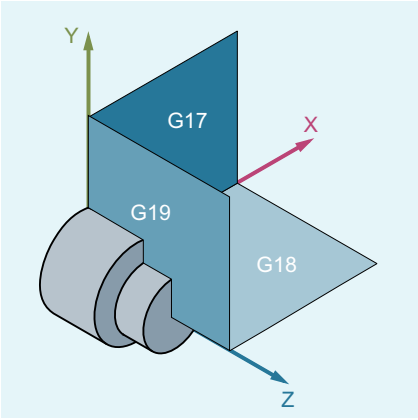
Pozice	Údaje poloh při zadávání inkrementálních rozměrů	Údaje jsou vztaženy na bod:
P1	X20 Y35	Nula (počátek souřadného systému)
P2	X30 Y20	P1
P3	X20 Y-35	P2

3.1.2 Pracovní roviny

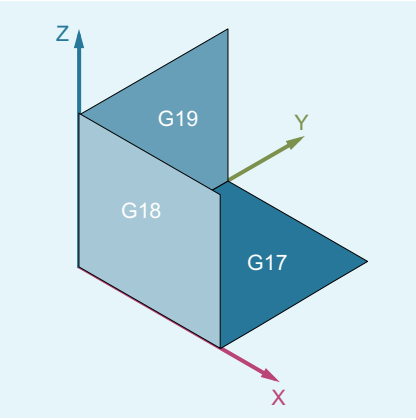
NC program potřebuje informaci, ve které rovině se má obrábění uskutečňovat. Pouze tehdy může řídící systém např. správně započítat hodnoty korekčních parametrů nástroje. Určení pracovní roviny je také nezbytné pro určité druhy programování kruhových drah a u polárních souřadnic.

Pracovní rovina je v základním kartézském souřadném systému obrobku definována prostřednictvím dvou souřadných os. Třetí souřadná osa je na tuto pracovní rovinu kolmá a určuje směr přísuvu nástroje (např. při 2D-obrábění).

Pracovní roviny pří soustružení / frézování



Pracovní roviny pří soustružení / frézování



Pracovní roviny pří frézování

Aktivování pracovní roviny

Pracovní roviny jsou v NC programu aktivovány pomocí příkazů G17, G18 a G19. Jejich vzájemná souvislost je definována následujícím způsobem:

Příkaz G-funkce	Pracovní rovina	abscisa	ordináta	Aplikáta ≙ směr přísuvu nástroje
G17	X/Y	X	Y	Z
G18	Z/X	Z	X	Y
G19	Y/Z	Y	Z	X

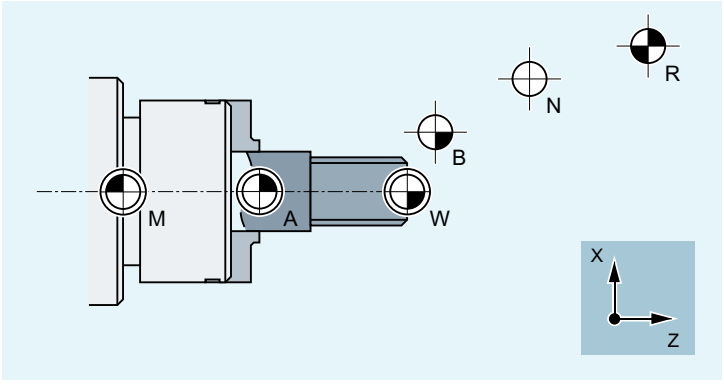
3.1.3 Počátky (nuly) souřadného systému a vztažné body

Na stroji s NC systémem jsou definovány různé počátky souřadných systémů (nuly) a vztažné body:

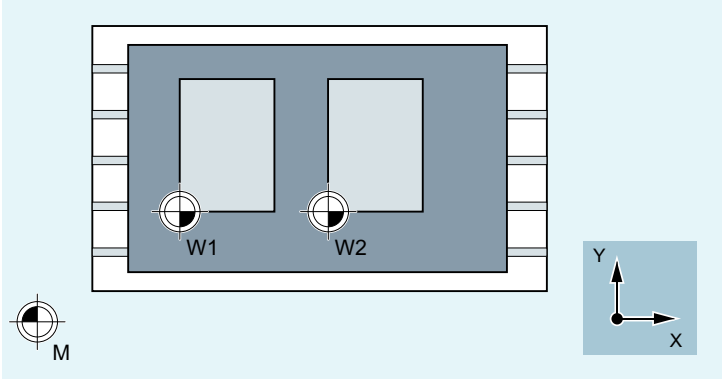
Počátky souřadného systému		
	M	Počátek souřadného systému stroje Prostřednictvím počátku souřadného systému stroje (nuly) je určen souřadný systém stroje (MCS). Na počátek souřadného systému stroje jsou vztaženy všechny ostatní vztažné body.
	W	Počátek souřadné soustavy obrobku = vztažný bod programu. Počátek (nula) souřadného systému obrobku definuje souřadný systém obrobku vzhledem k počátku souřadného systému stroje.
	A	Doraz Může se krýt s počátkem souřadné soustavy obrobku (jen u soustruhů).

Vztažné body		
	R	Referenční bod Poloha definovaná vačkami a měřicím systémem. Musí být známa vzdálenost tohoto bodu a počátku souřadné soustavy stroje **M**, aby poloha osy v tomto bodě mohla být nastavena přesně na tuto hodnotu.
	B	Počáteční bod Může být definován programem. Zde začíná pohyb 1. nástroje při obrábění.
	T	Vztažný bod držáku nástroje Nachází se držák se sklíčidlem nástroje. Prostřednictvím zadaných délek nástroje vypočítává řídící systém vzdálenost špičky nástroje od vztažného bodu držáku nástroje.
	N	Bod pro výměnu nástroje

Počátky (nuly) souřadného systému a vztažné body při soustružení



Počátky souřadného systému (nuly) při frézování



3.1.4 Souřadné systémy

Jsou rozlišovány následující souřadné systémy:

- Souřadný systém stroje (MCS) (Strana 38) s počátkem (nulou) stroje **M**
- Základní souřadný systém (BCS) (Strana 40)
- Základní souřadný systém počátku (nuly) (Strana 41)
- Nastavitelný souřadný systém (ENS) (Strana 41)
- Souřadný systém obrobku (WCS) (Strana 42) s počátkem (nulou) obrobku **W**

3.1.4.1 Souřadný systém stroje (MCS)

Souřadný systém stroje se skládá ze všech os, které na stroji fyzicky existují.

V souřadném systému stroje jsou definovány referenční body a body pro výměnu nástroje a palety (pevně stanovené body stroje).

Jestliže se programování uskutečňuje přímo v souřadném systému stroje (což je u některých příkazů G-funkcí možné), jsou přímo ovládány fyzické osy stroje. Přitom se vůbec nebere ohled na eventuálně existující parametry upnutí obrobku.

Poznámka

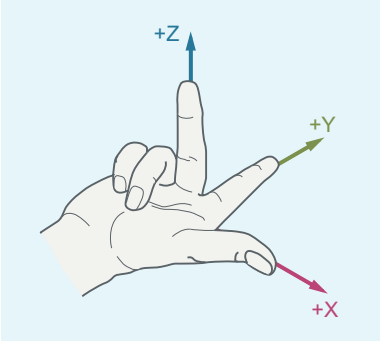
Jestliže existují různé souřadné systémy stroje (např. 5-osá transformace), potom se prostřednictvím interní transformace kinematiky stroje vytváří matematické zobrazení do souřadného systému, v němž se programuje.

Pravidlo tří prstů

To, jak je souřadný systém definován vůči stroji, závisí na typu příslušného stroje. Směry os se řídí tak zvaným "pravidlem tří prstů" **pravé** ruky (podle normy DIN 66217).

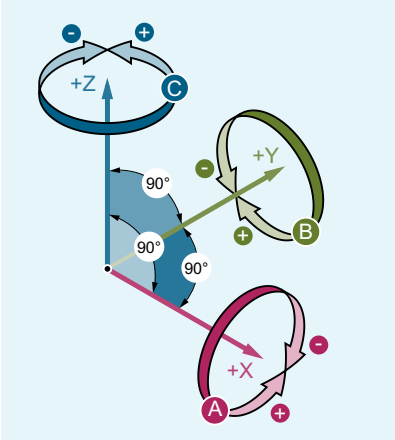
Jestliže člověk stojí před strojem tak, aby prostředníček jeho pravé ruky ukazoval proti směru přísuvu hlavního vřetena, potom je přiřazení následující:

- Palec ukazuje směr +X
- Ukazováček ukazuje směr +Y
- Prostředníček ukazuje směr +Z



Otáčivé pohyby okolo souřadných os X, Y a Z jsou označovány pomocí A, B a C. Směr otáčení je považován za kladný tehdy, pokud se při pohledu v kladném směru souřadné osy uskutečňuje tento otáčivý pohyb ve směru hodinových ručiček:

Směr otáčivého pohybu	**Smysl otáčení**
ve směru hodinových ručiček	kladný
proti směru hodinových ručiček	záporný

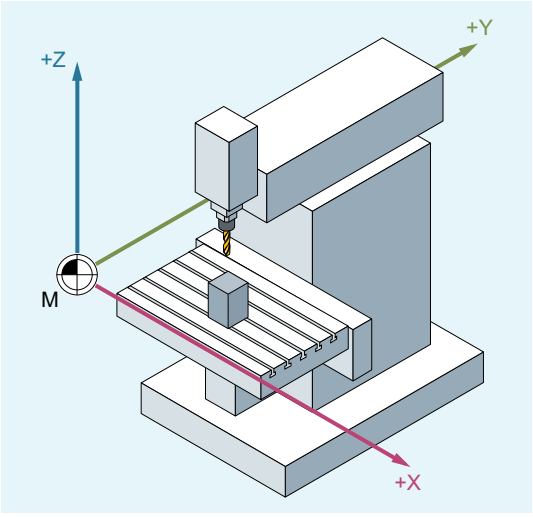


X, Y, Z Souřadné osy nacházející se kolmo vůči sobě

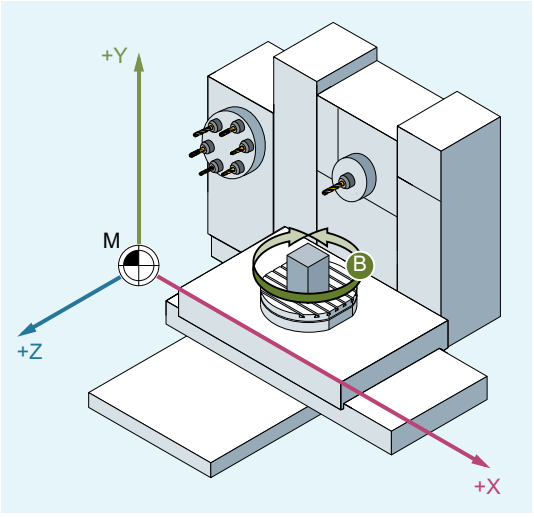
A, B, C Kruhové osy otáčející se okolo os X, Y, Z

Poloha souřadného systému u různých typů obráběcích strojů

Polohy souřadných systémů, které vyplývají z "pravidla tří prstů", mohou mít u různých typů obráběcích strojů různou orientaci, jak ukazují oba následující příklady:



Vertikální frézka se 3 osami



Horizontální frézka se 4 osami

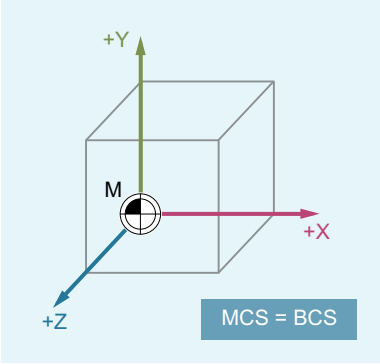
3.1.4.2 Základní souřadný systém (BCS)

Základní souřadný systém (BCS) se skládá ze tří os, které jsou vůči sobě v pravých úhlech (geometrické osy), ale i z dalších os (doplňkové osy) bez vzájemného geometrického vztahu.

Obráběcí stroje bez kinematické transformace

Jestliže se systém BCS může matematicky zobrazovat na MCS bez kinematické transformace (např. 5-osá transformace, TRANSMIT / TRACYL / TRAANG), jsou systémy BCS a MCS za všech okolností identické.

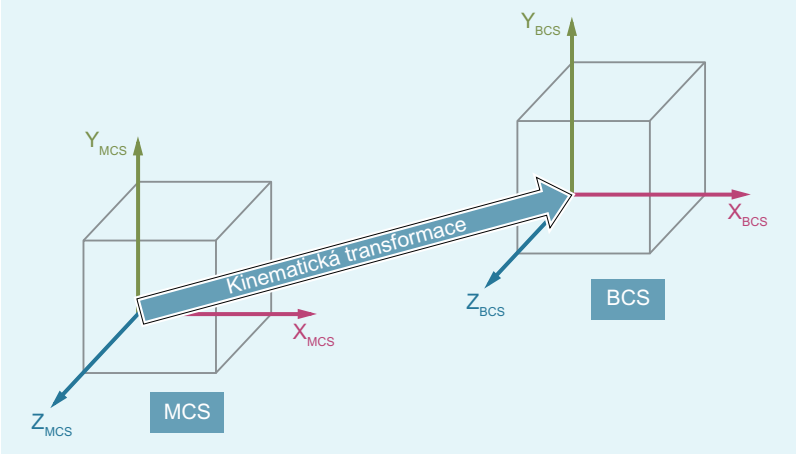
U těchto strojů mohou mít osy stroje a geometrické osy totožné názvy.



Obráběcí stroje s kinematickou transformací

Jestliže se systém BCS s kinematickou transformací (např. 5-osá transformace, TRANSMIT / TRACYL / TRAANG) matematicky zobrazuje na MCS, systémy BCS a MCS nejsou identické.

U těchto strojů musí mít osy stroje a geometrické osy odlišné názvy.

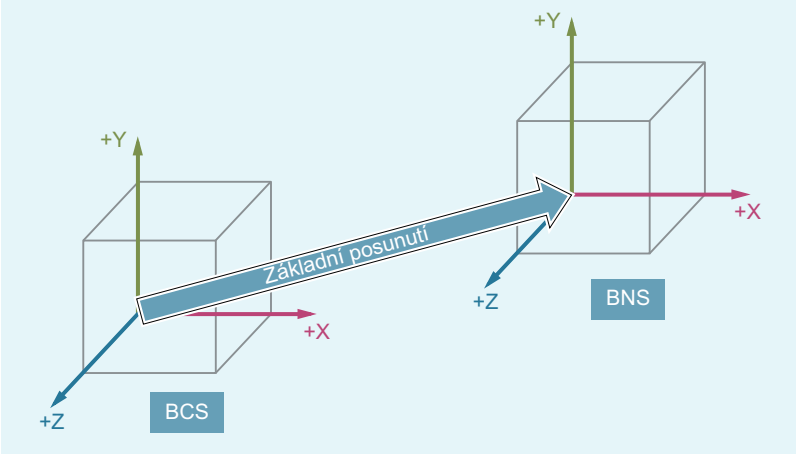


Kinematika stroje

Programování obrobku se vždy uskutečňuje ve dvou- nebo v trojrozměrném pravoúhlém souřadném systému (WCS). Pro výrobu těchto obrobků se ale stále častěji používají obráběcí stroje s kruhovými osami nebo lineární osy, které nejsou vůči sobě v pravém úhlu. Pro zobrazování souřadnic naprogramovaných ve WCS (pravoúhlý systém) do reálných pohybů os stroje slouží kinematické transformace.

3.1.4.3 Základní souřadný systém počátku (nuly) (BNS)

Základní souřadný systém počátku (nuly) vzniká složením základního souřadného systému a základního posunutí.



Základní posunutí

Základní posunutí popisuje transformaci souřadnic mezi systémy BCS a BNS. Jeho prostřednictvím může být definován např. nulový bod palety.

Základní posunutí se skládá z následujících složek:

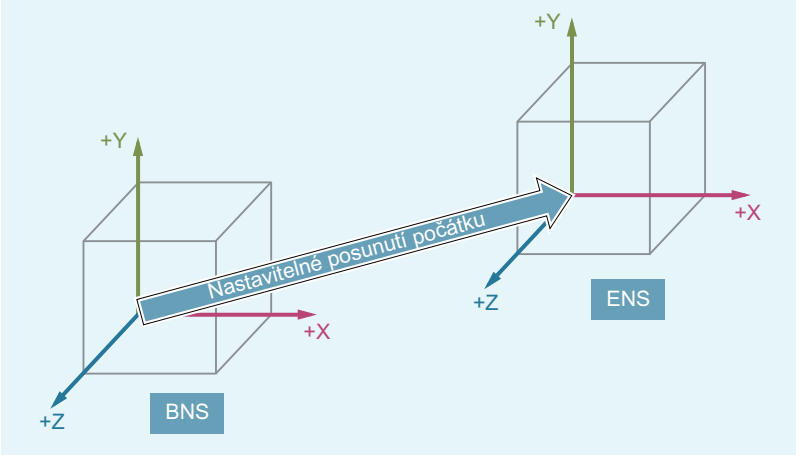
- Externí posunutí počátku
- Posunutí DRF
- Superponovaný pohyb
- Řetězec systémových framů
- Řetězec základních framů

3.1.4.4 Nastavitelný souřadný systém (ENS)

Nastavitelné posunutí počátku

Prostřednictvím nastavitelného posunutí počátku vyplývá ze základního souřadného systému počátku (nuly) (BNS) "nastavitelný souřadný systém" (ENS).

Nastavitelná posunutí počátku (nulového bodu) jsou v NC-programu aktivovány příkazy G54 ... G57 a G505 ... G599.



Jestliže žádné programovatelné transformace (framy) nejsou aktivní, potom se "nastavitelný souřadný systém" kryje se souřadným systémem obrobku (WCS).

Programovatelné transformace souřadného systému (framy)

Občas se jeví jako výhodné nebo dokonce nezbytné uvnitř jednoho NC-programu dříve zvolený počátek souřadného systému obrobku (příp. "Nastavitelný souřadný systém") posunout na jiné místo, případně systém pootočit, zrcadlově jej převrátit / změnit měřítko os. To se uskutečňuje prostřednictvím programovatelných transformací souřadného systému (framů) (Strana 307).

Poznámka

Programovatelné transformace souřadného systému (framy) se vždy vztahují na "nastavitelný souřadný systém".

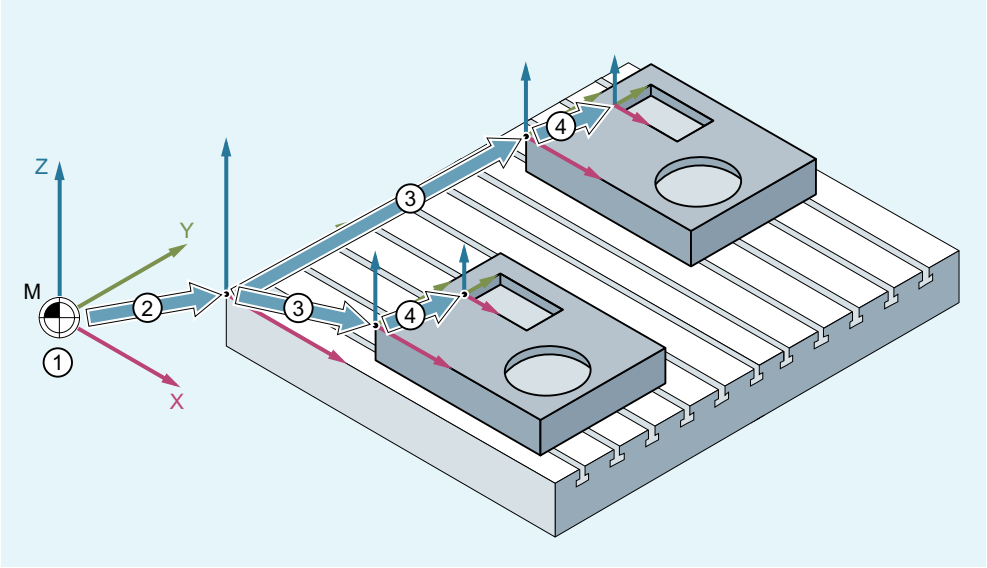
3.1.4.5 Souřadný systém obrobku (WCS)

V souřadném systému obrobku (WCS) je popisována geometrie obrobku. Nebo řečeno jinými slovy: Údaje v NC-programu se vztahují na souřadný systém obrobku.

Souřadný systém obrobku je vždy kartézským souřadným systémem, který je přiřazen jednomu určitému obrobku.

3.1.4.6 Jak spolu souvisejí různé souřadné systémy?

Následující příklad má vysvětlit souvislosti mezi různými souřadnými systémy:



① Kinematická transformace není aktivní. Tzn. souřadný systém stroje se kryje se základním souřadným systémem.

② Začleněním základního posunutí vzniká základní souřadný systém počátku (nuly) (BNS) s nulovým bodem palety.

③ Prostřednictvím nastavitelného posunutí počátku G54, příp. G55 je definován "nastavitelný souřadný systém" (ENS) pro obrobek 1, příp. pro obrobek 2.

④ Prostřednictvím programovatelné transformace souřadného systému vzniká souřadný systém obrobku (WCS).

3.2 Základy programování NC systémů

Poznámka

Směrnicí pro programování NC systémů je norma DIN 66025.

3.2.1 Sestavování názvů NC programů

Pravidla

Každému NC programu musí být při jeho vytvoření přiřazen název programu (identifikátor). Název programu může být libovolný, musí však být dodržena následující pravidla:

- Přípustné znaky:
  - Písmena: A ... Z, a ... z
  - Číslice: 0 ... 9
  - Znak podtržení: _
- První dva znaky by měly být dvě písmena nebo znak podtržení, za kterým následuje písmeno.

  Poznámka

  Jestliže jsou tyto podmínky splněny, potom může být tento NC program samotný vyvoláván jako podprogram z nějakého jiného programu zadáním tohoto názvu programu. Jestliže název programu proti tomu začíná číslicemi, potom je volání podprogramu možné jedině pomocí příkazu CALL.

- Maximální délka: 24 znaků

Poznámka

Velká/malá písmena

V jazyku NC systému SINUMERIK **nejsou** rozlišována velká a malá písmena.

Poznámka

Nepřípustné názvy programů

Aby se zabránilo problémům s aplikacemi Windows, následující názvy programů se **nesmějí** používat:

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

Pokud budete potřebovat informace o dalších omezeních, viz "Název (Strana 382)".

Interní přípony řídícího systému

Název programu zadaný při jeho sestavování je uvnitř řídícího systému rozšířen o předponu a o příponu.

- Počáteční řetězec: _N_
- Koncový řetězec:
  - Hlavní programy: _MPF
  - Podprogramy: _SPF

Soubory ve formátu děrné pásky

Soubory externě sestavených programů, které mají být načteny prostřednictvím rozhraní RS 232, musí existovat ve formátu děrné pásky.

Pro název programového souboru ve formátu děrné pásky platí ještě navíc následující pravidla:

- První znak: %
- Na konci se připojuje identifikace souboru v délce čtyř znaků: _xxx

Příklady:

- %_N_WELLE123_MPF
- %Priruba3_MPF

3.2.2 Struktura a obsah NC programu

3.2.2.1 Bloky a komponenty bloku

Bloky

NC program se skládá z posloupnosti NC-bloků. Každý blok obsahuje data pro uskutečnění jednoho kroku pracovního postupu při opracovávání obrobku.

Komponenty bloku

NC-bloky se skládají z následujících komponent:

- Příkazy (instrukce) podle normy DIN 66025
- Prvky vyššího jazyka NC systému

Příkazy podle normy DIN 66025

Příkazy se podle normy DIN 66025 skládají z adresového znaku a číslice, příp. posloupnosti číslic, které reprezentují aritmetickou hodnotu.

Adresový znak (adresa)

Adresový znak (obecně jedno písmeno) definuje význam příkazu.

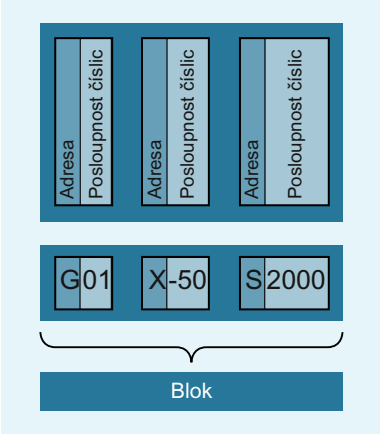
Příklady:

Adresové znaky	Význam
G	Příkaz G-funkce (podmínka dráhy)
X	Informace o dráze pro osu X
S	Otáčky vřetena

Posloupnost číslic

Posloupnost číslic je hodnota, která je přiřazena adresovému znaku. Posloupnost číslic může obsahovat znaménko a desetinnou tečku, přičemž znaménko se nachází vždy mezi adresovým

písmenem a posloupností číslic. Kladné znaménko (+) a nuly na začátku (0) není potřeba zapisovat.



Prvky vyššího jazyka NC systému

Protože sada příkazů podle normy DIN 66025 už není pro programování složitých pracovních postupů na moderních obráběcích strojích postačující, byla sada rozšířena o prvky vyššího programovacího jazyka NC systému.

K tomu mimo jiné patří:

- Příkazy vyššího jazyka NC systému
  Narozdíl od příkazů podle normy DIN 66025 se příkazy vyššího jazyka NC systému skládají z většího počtu adresových písmen, např.:
  - `OVR` pro korekci otáček (override)
  - `SPOS` pro nastavování vřetena do určité polohy
- Identifikátor (definovaný název) pro:
  - Systémové proměnné
  - Uživatelem definované proměnné
  - Podprogramy
  - Klíčová slova
  - Návěští skoků
  - Makra

  Poznámka

  Identifikátor musí být jedinečný a se nesmí používat pro různé objekty.
- Relační operátory
- Logické operátory
- Matematické funkce
- Řídící struktury

Platnost příkazů

Příkazy mohou mít modální nebo blokovou platnost:

- Modální
  Příkazy s modální platností zůstávají zachovány s naprogramovanou hodnotou (ve všech následujících blocích), dokud:
  - není stejným příkazem naprogramována stejná hodnota.
  - je naprogramován příkaz, který působení až dosud platného příkazu ruší.
- Bloková
  Příkazy s blokovou platností platí pouze v bloku, v němž jsou naprogramovány.

Konec programu

Poslední blok v posloupnosti opracování obsahuje speciální slovo pro konec programu: `M2`, `M17` příp. `M30`.

3.2.2.2 Pravidla pro sestavování bloku

Začátek bloku

NC bloky mohou být na svém začátku označeny číslem bloku. Tato čísla se skládají ze znaku "N“ a kladného celého čísla, např.:
`N40 ...`

Posloupnost čísel bloků je libovolná, doporučujeme Vám však používat narůstající posloupnost čísel bloků.

Poznámka

Čísla bloků musí být v rámci programu jednoznačná, aby při vyhledávání bloku bylo možné dosáhnout jednoznačného výsledku.

Koncový blok

Blok končí znakem LF (LINE FEED = nový řádek).

Poznámka

Znak LF nemusí být zapisován. Vkládá se automaticky při přechodu na další řádek.

Délka bloku

Blok může obsahovat maximálně **512 znaků** (včetně komentáře a znaku konce bloku LF).

Poznámka

V okně aktuálního bloku na obrazovce se za normálních okolností zobrazují tři bloky po 66 znacích. Vypisují se také komentáře. Hlášení se zobrazují v samostatném okně hlášení.

Posloupnost příkazů

Aby struktura bloku zůstala co možno nejpřehlednější, měly by být příkazy v bloku uspořádány v následující posloupnosti:
N... G... X... Y... Z... F... S... T... D... M... H...

Adresa	Význam
N	Adresa čísla bloku
G	Podmínka dráhy
X, Y, Z	Informace o dráze
F	Posuv
S	Otáčky
T	Nástroj
D	Číslo korekčních parametrů nástroje
M	Doplňková funkce
H	Pomocná funkce

Poznámka

Některé adresy mohou být v rámci jednoho bloku použity i několikrát, např.:

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

3.2.2.3 Přiřazování hodnot

Adresám mohou být přiřazovány hodnoty. Přitom platí následující pravidla:

- Znak "=“ mezi adresou a hodnotou se musí zapsat tehdy, pokud:
  - Adresa se skládá z více než jednoho písmena.
  - Přiřazovaná hodnota je více než konstantou.

  Znak "=“ je možné vypustit, jestliže adresa je jediným písmenem a jestliže hodnota je konstantou.
- Je možno používat také znaménka.
- Oddělovací znak za písmenem adresy je přípustný.

Příklady:

`X10`	Přiřazení hodnoty (10) adrese „X“, znak „=“ není zapotřebí.
`X1=10`	Přiřazení hodnoty (10) adrese „X“ a numerickým rozšířením (1), znak „=“ je nutný
`X=10*(5+SIN(37.5))`	Přiřazení hodnot pomocí numerického výrazu, znak "=“ je nutný

Poznámka

Po numerickém rozšíření musí vždy následovat některý ze zvláštních znaků „=“, „(“, „)“, „[“, „]“, „,“ nebo operátor, aby se adresa s numerickým rozšířením odlišila od adresového písmena s hodnotou.

3.2.2.4 Komentáře

Aby se zvýšila srozumitelnost NC programů, je možné k NC blokům připojovat komentáře.

Komentář se vyskytuje na konci bloku a od programové části NC bloku je oddělen středníkem (";").

Příklad 1:

Programový kód	**Komentář**
`N10 G1 F100 X10 Y20`	`; komentář obsahující vysvětlení obsahu NC bloku`

Příklad 2:

Programový kód	**Komentář**
`N10`	`; firma G&S, zakázka č. 12A71`
`N20`	`; program sestavil H. Müller, odd. TV 4, dne 21.11.94`
`N50`	`; díl č. 12, kryt pro ponorné čerpadlo, typ TP23A`

Poznámka

Komentáře se ukládají a vypisují při zobrazení právě zpracovávaného bloku.

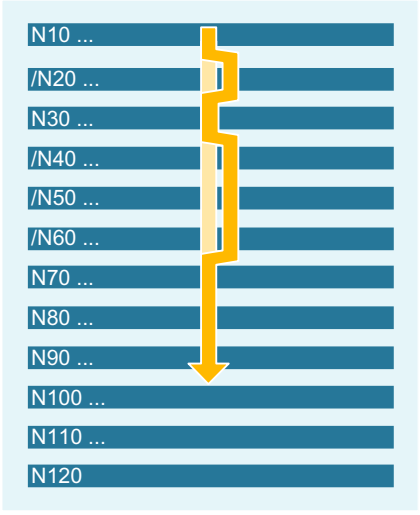
3.2.2.5 Přeskakování bloků

NC bloky, které se nemají uskutečňovat při každém průchodu programem, mohou být při jeho zpracovávání přeskakovány. Tato funkce nachází uplatnění např. při testování, příp. při zkouškách nového programu.

Přeskakované bloky

Bloky, které mají být přeskakovány, je třeba označit znakem „/“ (lomítko) před číslem bloku. Je možné přeskočit i několik bloků za sebou. Příkazy v přeskočených blocích se neprovedou, program bude pokračovat následujícím nepřeskakovaným blokem.

Příklad:



Programový kód	Komentář
N10 …	; se zpracuje
/N20 …	; je přeskočen
N30 …	; se zpracuje
/N40 …	; je přeskočen
/N50 …	; je přeskočen
/N60 …	; je přeskočen
N70 …	; se zpracuje
...	

Přeskakované úrovně

Bloky mohou být přiřazovány přeskakovaným úrovním (max. 10), které lze aktivovat prostřednictvím uživatelského rozhraní nebo uživatelského programu PLC.

Přiřazení se v NC programu uskutečňuje tak, že na začátku bloku se zapisuje lomítko, za nímž následuje číslo přeskakované úrovně. V každém bloku může být specifikována jen jedna přeskakovaná úroveň.

Příklad:

Programový kód	Komentář
/ ...	; blok se bude přeskakovat (1. přeskakovaná úroveň)
/0 ...	; blok se bude přeskakovat (1. přeskakovaná úroveň)
/1 N010...	; blok se bude přeskakovat (2. přeskakovaná úroveň)
/2 N020...	; blok se bude přeskakovat (3. přeskakovaná úroveň)

Programový kód	Komentář
...	
/7 N100...	; blok se bude přeskakovat (8. přeskakovaná úroveň)
/8 N080...	; blok se bude přeskakovat (9. přeskakovaná úroveň)
/9 N090...	; blok se bude přeskakovat (10. přeskakovaná úroveň)

Poznámka

Změna v přeskakovaných úrovních jsou možné jen tehdy, je-li řídící systém ve stavu STOP/Reset.

Poznámka

To, kolik úrovní přeskakování je k dispozici, závisí na strojním parametru pro zobrazování.

Poznámka

Přeskakování bloků je aktivní také při vyhledávání bloku.

Poznámka

Pro ovládání zpracování programu je možné používat také systémové a uživatelské proměnné v podmíněných skocích.

3.3 Založit NC program

3.3.1 Základní postup

Při sestavování NC programu je vlastní programování, tzn. převádění jednotlivých kroků pracovního postupu do NC jazyka, většinou jen menší částí práce při vývoji NC programu.

Před vlastním programováním by měl být velký důraz kladen na plánování a přípravu pracovního postupu. Čím přesněji si předem naplánujete, jak má být program strukturován a organizován, tím rychleji a snáze se Vám podaří vlastní hotový NC program napsat tak, aby byl přehledný a bez chyb. Přehledné programy jsou kromě toho zvláště výhodné tehdy, když později potřebujete provést v nich nějaké změny.

Jelikož se obrobky liší tvarem a formou, nemá smysl používat pro vytváření veškerých programů přesně stejné metody. Pro všeobecné případy se ale ukázal jako praktický následující postup.

Postup

1. **Příprava výrobního výkresu**
   - Stanovení počátku souřadného systému obrobku
   - Zakreslení souřadného systému
   - Přepočítání případných vadných souřadnic
2. **Stanovení postupu opracování**
   - Které nástroje budou použity pro obrábění jednotlivých kontur a kdy?
   - V jaké posloupnosti budou jednotlivé prvky obrobku vyráběny?
   - Jaké jednotlivé prvky se opakují (případně i v otočeném stavu) a měly by být uloženy do samostatného podprogramu?
   - Existují v jiném výrobním programu, příp. podprogramu dílčí kontury, které by bylo možné pro aktuální obrobek znovu použít?
   - Kde je účelné nebo nezbytné použití posunutí počátku, otočení, zrcadlového převrácení nebo změny měřítka (koncepce framů)?
3. **Sestavení pracovního postupu**
   Definujte jednotlivé kroky celého obráběcího postupu, např.:
   - Přesun rychlým posuvem na požadované místo
   - Výměna nástroje
   - Definice roviny opracování
   - Volný posuv pro dodatečné změření
   - Zapnutí/vypnutí vřetena, chladicí kapaliny
   - Vyvolání parametrů nástroje
   - Přísuv
   - Korekce dráhy
   - Najíždění na konturu
   - Odjíždění od kontury
   - atd.
4. **Převedení kroků obráběcího postupu do programovacího jazyka**
   - Každý jednotlivý krok se zapíše jako NC blok (příp. několik NC bloků).
5. **Sestavení všech jednotlivých kroků do NC programu**

3.3.2 Znaky, které jsou k dispozici

Pro sestavování NC programů jsou Vám k dispozici následující znaky:

- Velká písmena:
  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
- Malá písmena:
  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
- Číslice:
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Speciální znaky:
  Viz následující tabulka!

Speciální znaky	Význam
%	počáteční znak programu (pouze pro sestavování programů na externím PC)
(	závorka pro uzavření parametrů nebo výrazů
)	závorka pro uzavření parametrů nebo výrazů
[	závorka pro uzavření adres nebo indexů pole
]	závorka pro uzavření adres nebo indexů pole
<	je menší než
>	je větší než
:	hlavní blok, ukončení návěští, řetězcový operátor
=	přiřazení, součást rovnice
/	dělení, potlačení bloku
*	násobení
+	sečítání
-	odečítání, záporné znaménko
"	uvozovky, označení řetězce znaků
'	apostrof, označení speciální číselné hodnoty: hexadecimální, binární
$	identifikátor systémové proměnné
s_	znak podtržení, patří k písmenům
?	rezervováno
!	rezervováno
.	desetinná tečka
,	čárka, oddělení parametrů
;	středník, začátek komentáře
&	formátovací znak, stejný význam jako mezera
LF	Konec bloku
Tabulátor	oddělovací znak
mezera	oddělovací znak

Poznámka

Písmeno "O“ nesmí být zaměňováno za číslici nula "0“!

Poznámka

Malá a velká písmena nejsou rozlišována (výjimka: volání nástroje).

Poznámka

S netisknutelnými speciálními znaky se zachází stejně jako s mezerami.

3.3.3 Hlavička programu

NC bloky, které jsou uvedeny před vlastními pohybovými bloky sloužícími pro výrobu kontury obrobku, jsou označovány jako hlavička programu.

Hlavička programu obsahuje informace / příkazy, které se týkají následujících oblastí:

- Výměna nástroje
- Korekční parametry nástroje
- Pohyby vřetena
- Regulace posuvu
- Geometrické parametry (posunutí počátku, volba pracovní roviny)

Hlavička programu při soustružení

Následující příklad ukazuje, jakým způsobem je sestavena typická hlavička NC programu v případě soustružení:

Programový kód	**Komentář**
N10 G0 G153 X200 Z500 T0 D0	; stažení držáku nástroje před otočením revolverového držáku nástrojů
N20 T5	; otočení nástroje 5 do pracovní polohy
N30 D1	; aktivování datového bloku břitu nástroje
N40 G96 S300 LIMS=3000 M4 M8	; konstantní řezná rychlost (Vc) = 300 m/min, omezení max. otáček = 3000 ot/min, směr otáčení vlevo, chlazení aktivováno
N50 DIAMON	; pro osu X jsou programovány hodnoty průměrů
N60 G54 G18 G0 X82 Z0.2	; vyvolání posunutí počátku a pracovní roviny, najetí na počáteční pozici
...	

Hlavička programu při frézování

Následující příklad ukazuje, jakým způsobem je sestavena typická hlavička NC programu v případě frézování:

Programový kód	**Komentář**
N10 T="SF12"	; alternativa: T123
N20 M6	; spuštění výměny nástroje

Programový kód	Komentář
N30 D1	; aktivování datového bloku břitu nástroje
N40 G54 G17	; posunutí počátku a pracovní rovina
N50 G0 X0 Y0 Z2 S2000 M3 M8	; najížděcí pohyb k obrobku, zapnutí vřetena a chladicí kapaliny
...	

Jestliže se pracuje s orientací nástroje / s transformacemi souřadného systému, měly by být na začátku programu zrušeny případné stále ještě aktivní transformace:

Programový kód	Komentář
N10 CYCLE800()	; nastavení naklopené roviny do původní polohy
N20 TRAFOOF	; deaktivování transformací TRAORI, TRANSMIT, TRACYL, ...
...	

3.3.4 Příklady programování

3.3.4.1 Příklad 1: První kroky při programování

Příklad programu 1 by měl posloužit k prvnímu pracování a otestování programových kroků.

Postup

1. Vytvořte nový výrobní program (název)
2. Výrobní program editujte
3. Vyberte výrobní program
4. Aktivujte zpracování blok po bloku
5. Spusťte zpracování výrobního programu

Poznámka

Aby bylo možné program na obráběcím stroji zpracovat, musí být odpovídajícím způsobem nastaveny strojní parametry (--> viz dokumentace od výrobce stroje!).

Poznámka

Při testování programu se mohou vyskytnout alarmy, které je napřed nutno deaktivovat.

NC program

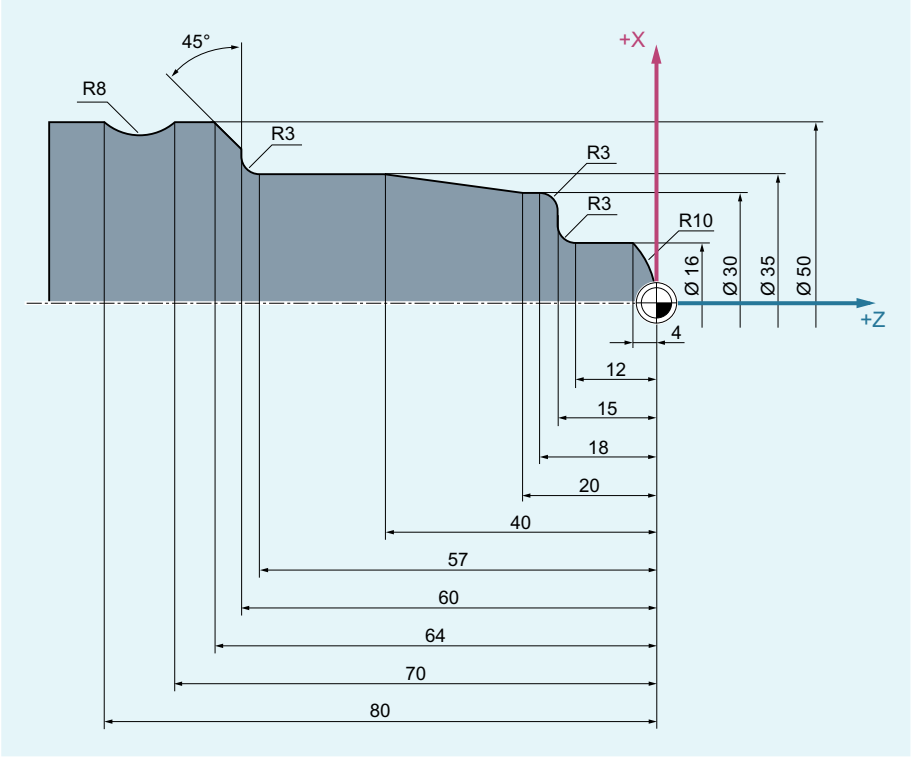
Programový kód	Komentář
N10 MSG("TO JE MUJ NC-PROGRAM")	; výpis hlášení „TO JE MUJ PROGRAM" na řádku alarmových hlášení

Programový kód	Komentář
N20 F200 S900 T1 D2 M3	; posuv, vřeteno, nástroj, korekce nástroje, vřeteno se otáčí vpravo
N30 G0 X100 Y100	; najíždění na pozici rychlým posuvem
N40 G1 X150	; obdélník pracovním posuvem, přímka ve směru X
N50 Y120	; přímka ve směru Y
N60 X100	; přímka ve směru X
N70 Y100	; přímka ve směru Y
N80 G0 X0 Y0	; návrat rychlým posuvem
N100 M30	; koncový blok

3.3.4.2 Příklad 2: NC program k soustružení

Příklad programování 2 je určen pro obrábění obrobku na soustruhu. Ukazuje programování rádiusů a korekci rádiusu nástroje.

Výkres rozměrů obrobku



NC program

Programový kód	Komentář
N5 G0 G53 X280 Z380 D0	; počáteční bod

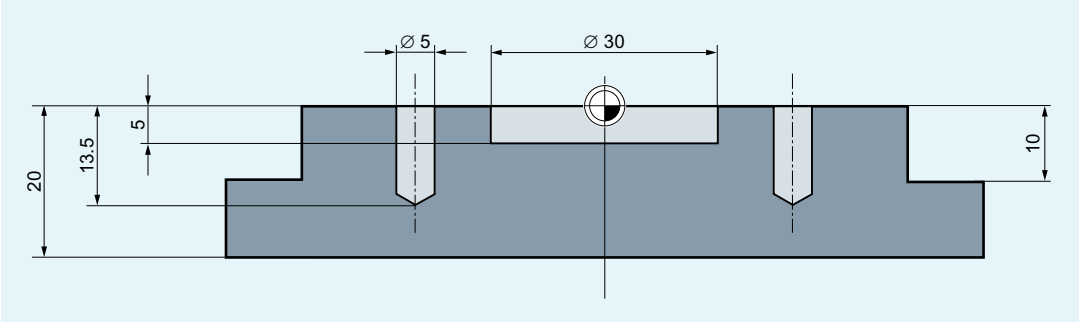
Programový kód	**Komentář**
N10 TRANS X0 Z250	; posunutí počátku
N15 LIMS=4000	; omezení otáček (G96)
N20 G96 S250 M3	; aktivování konstantní řezné rychlosti
N25 G90 T1 D1 M8	; volba nástroje a aktivování korekčních parametrů
N30 G0 G42 X-1.5 Z1	; nastavení nástroje do potřebné polohy s korekcí rádiusu nástroje
N35 G1 X0 Z0 F0.25	
N40 G3 X16 Z-4 I0 K-10	; soustružení rádiusu 10
N45 G1 Z-12	
N50 G2 X22 Z-15 CR=3	; soustružení rádiusu 3
N55 G1 X24	
N60 G3 X30 Z-18 I0 K-3	; soustružení rádiusu 3
N65 G1 Z-20	
N70 X35 Z-40	
N75 Z-57	
N80 G2 X41 Z-60 CR=3	; soustružení rádiusu 3
N85 G1 X46	
N90 X52 Z-63	
N95 G0 G40 G97 X100 Z50 M9	; deaktivování korekce rádiusu nástroje a najetí na bod pro výměnu nástroje
N100 T2 D2	; vyvolání nástroje a korekčních parametrů
N105 G96 S210 M3	; aktivování konstantní řezné rychlosti
N110 G0 G42 X50 Z-60 M8	; nastavení nástroje do potřebné polohy s korekcí rádiusu nástroje
N115 G1 Z-70 F0.12	; soustružení průměru 50
N120 G2 X50 Z-80 I6.245 K-5	; soustružení rádiusu 8
N125 G0 G40 X100 Z50 M9	; pozvednutí nástroje a deaktivování korekce rádiusu nástroje
N130 G0 G53 X280 Z380 D0 M5	; najetí na bod pro výměnu nástroje
N135 M30	; konec programu

3.3.4.3 Příklad 3: NC program ke frézování

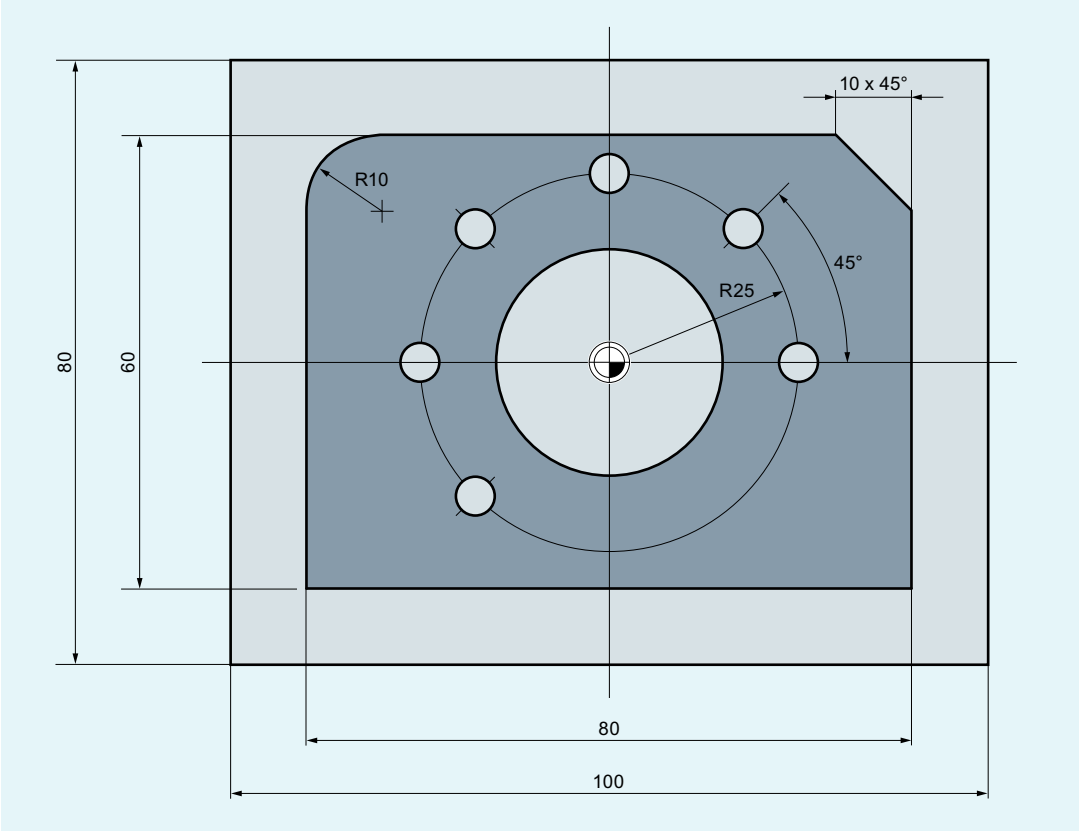
Příklad programování 3 je určen pro obrábění obrobku na vertikální frézce. Zahrnuje frézování pláště a čelní plochy, jakož i vrtání.

Výkres rozměrů obrobku

Boční pohled



Pohled shora



NC program

Programový kód	Komentář
N10 T="PF60"	; předvolba nástroje s názvem PF60.
N20 M6	; upnutí nástroje do vřetena
N30 S2000 M3 M8	; otáčky, směr otáčení, zapnutí chlazení

Programový kód	**Komentář**
N40 G90 G64 G54 G17 G0 X-72 Y-72	; základní nastavení geometrie a najíždění na počáteční bod
N50 G0 Z2	; osa Z na bezpečnostní vzdálenost
N60 G450 CFTCP	; chování při aktivním příkazu G41/G42.
N70 G1 Z-10 F3000	; frézování na hloubce záběru s posuvem = 3000 mm/min
N80 G1 G41 X-40	; zapnutí korekce rádiusu frézy
N90 G1 X-40 Y30 RND=10 F1200	; najíždění na konturu s posuvem = 1200 mm/min
N100 G1 X40 Y30 CHR=10	
N110 G1 X40 Y-30	
N120 G1 X-41 Y-30	
N130 G1 G40 Y-72 F3000	; vypnutí korekce rádiusu frézy
N140 G0 Z200 M5 M9	; pozvednutí frézy, vypnutí vřetena + chlazení
N150 T="SF10"	; předvolba nástroje s názvem SF10.
N160 M6	; upnutí nástroje do vřetena
N170 S2800 M3 M8	; otáčky, směr otáčení, zapnutí chlazení
N180 G90 G64 G54 G17 G0 X0 Y0	; základní nastavení pro geometrii a najíždění na počáteční bod
N190 G0 Z2	
N200 POCKET4(2,0,1,-5,15,0,0,0,0,0,800,1300,0,21,5,,,2,0.5)	; vyvolání cyklu pro frézování kapsy
N210 G0 Z200 M5 M9	; pozvednutí frézy, vypnutí vřetena + chlazení
N220 T="ZB6"	; vyvolání vrtáku pro navrtávání středicích důlků 6 mm
N230 M6	
N240 S5000 M3 M8	
N250 G90 G60 G54 G17 X25 Y0	; přesné najetí G60 kvůli přesnému polohování
N260 G0 Z2	
N270 MCALL CYCLE82(2,0,1,-2.6,,0)	; modální volání cyklu pro vrtání
N280 POSITION:	; značka pro skok kvůli opakování
N290 HOLES2(0,0,25,0,45,6)	; polohovací vzor pro vrtání
N300 ENDLABEL:	; značka konce kvůli opakování
N310 MCALL	; deaktivování modálního volání
N320 G0 Z200 M5 M9	
N330 T="SPB5"	; vyvolání spirálního vrtáku D 5 mm
N340 M6	
N350 S2600 M3 M8	
N360 G90 G60 G54 G17 X25 Y0	
N370 MCALL CYCLE82(2,0,1,-13.5,,0)	; modální volání cyklu pro vrtání
N380 REPEAT POSITION	; opakování popisu polohování z navrtávání středicích důlků
N390 MCALL	; deaktivování cyklu pro vrtání
N400 G0 Z200 M5 M9	

Programový kód	Komentář
N410 M30	; konec programu

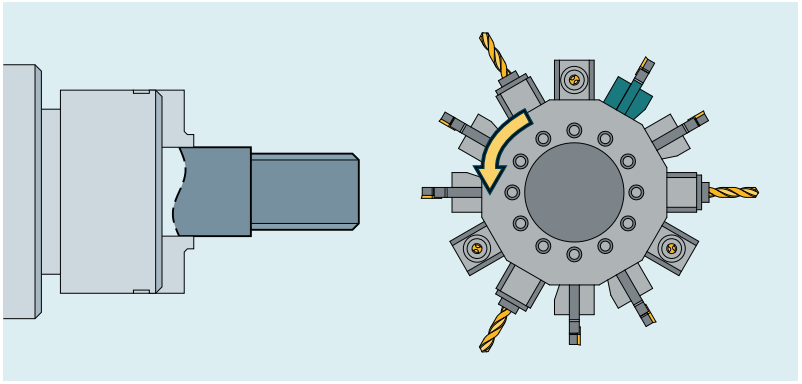
3.4 Výměna nástroje

Způsob výměny nástroje

Poznámka

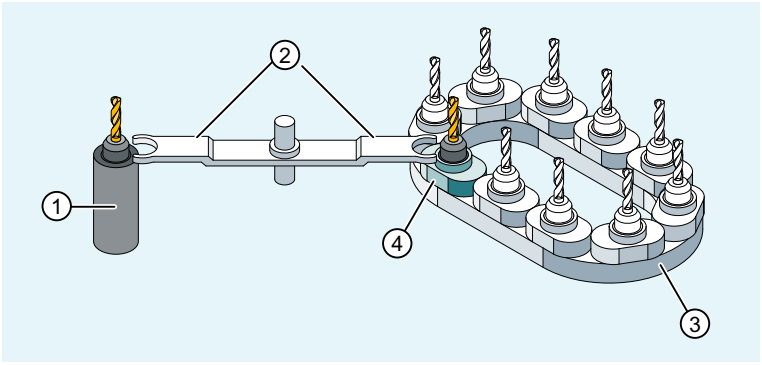
Způsob výměny nástroje je definován výrobcem stroje při jeho uvádění do provozu.

Výměna nástroje u soustruhů s revolverovým zásobníkem



V případě revolverových zásobníků na soustruzích se výměna nástroje, tedy jeho vyhledání a upnutí nástroje, vyvolává pouze příkazem T.

Výměna nástroje u obráběcích strojů s řetězovým, diskovým a plošným zásobníkem



① Vřeteno
② Podavač
③ Zásobník (zde: řetězový zásobník)
④ Místo pro výměnu pro vřeteno

V případě řetězových, diskových a plošných zásobníků se výměna nástroje za normálních okolností uskutečňuje ve dvou krocích:

1. Pomocí příkazu T se nástroj vyhledá v zásobníku.
2. Potom se prostřednictvím příkazu M vymění ve vřetenu.

Programování výměny nástroje s aktivní / neaktivní správou nástrojů

Programování výměny nástroje v případě aktivní správy nástrojů se liší od programování výměny nástroje u systémů, ve kterých je správa nástrojů neaktivní. Jsou proto popisovány obě varianty:

→ Výměna nástroje, když je aktivní správa nástrojů (Strana 61)

→ Výměna nástroje, když správa nástrojů není aktivní (Strana 65)

Programování pracovní roviny

Při výměně nástroje musí být naprogramována odpovídající pracovní rovina (Strana 35) (základní nastavení: G18). Tím je zaručeno, že korekce délky nástroje bude přiřazena správné ose.

Aktivování korekčních parametrů nástroje

Výměnou nástroje jsou aktivovány korekční parametry nástroje uložené do D-čísla (Strana 80).

3.4.1 Výměna nástroje, když je aktivní správa nástrojů

Správa nástrojů

Funkce "Správa nástrojů" zajišťuje, že na stroji je v každém okamžiku ten správný nástroj na správném místě a že data, která byla nástroji přiřazena, odpovídají aktuálnímu stavu. Kromě toho umožňuje tento modul rychlou výměnu nástrojů, zabraňuje výrobě zmetků díky monitorování doby používání nástroje, ale také v důsledku sledování náhradních nástrojů zkracuje doby, kdy je stroj mimo provoz.

Názvy nástrojů

Kvůli jednoznačné identifikaci je nutné na obráběcím stroji, na němž byla správa nástrojů aktivována, všechny nástroje opatřit názvy a čísly (např. "VRTAK", "3").

Nástroj je potom možné vyvolávat pomocí jeho názvu, např.:
`T="VRTAK"`

Poznámka

Název nástroje nesmí obsahovat žádné speciální znaky.

3.4.1.1 Výměna nástroje, když je aktivní aktivní správa nástrojů s funkcí T=místo

Naprogramováním funkce T=místo se uskuteční přímá výměna nástroje.

Použití

U soustruhů s revolverovým zásobníkem.

Syntaxe

Vyberte nástroj
```
T=<č.>
T=<název>
T<n>=<č.>
T<n>=<název>
```

Deaktivování nástroje
```
T0
```

Význam

`T=`:	Adresa pro aktivování nástroje, včetně výměny nástroje a aktivování korekčních parametrů nástroje Pokud jde o přiřazení hodnoty, existují následující možnosti:	
	`<č.>`:	Číslo místa v zásobníku
	`<název>`:	Název nástroje **Poznámka:** Při programování názvů nástrojů je nutno dbát na správný způsob zápisu (velká a malá písmena jsou rozlišována).
`<n>`:	Číslo vřetena jako rozšíření adresy **Poznámka:** Možnost programovat číslo vřetena jako rozšíření adresy závisí na konfiguraci stroje (--> věnujte prosím pozornost informacím od výrobce stroje).	
`T0`:	Deaktivování nástroje (místo v zásobníku není obsazeno)	

Poznámka

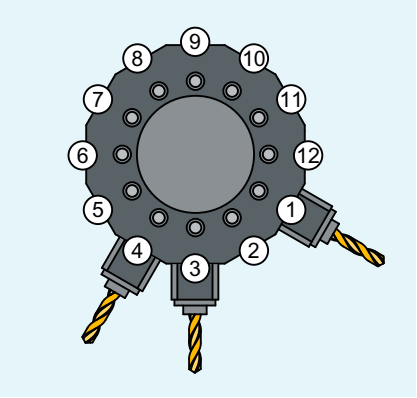
Jestliže zvolené místo v zásobníku nástrojů není obsazeno, má tento příkaz stejný účinek jako zadání příkazu T0. Zvolení neobsazených míst v zásobníku se může používat pro nastavování prázdných míst do požadované polohy.

Příklad

Revolverový zásobník má místa od 1 do 12 s následujícím obsazením nástroji:

Místo	Nástroj	Interní T-číslo	Stav
1	VRTAK, duplo č. = 1	T15	blokovaná
2	neobsazená		

Místo	Nástroj	Interní T-číslo	Stav
3	VRTAK, duplo č. = 2	T10	uvolněná
4	VRTAK, duplo č. = 3	T1	aktivní
5 ... 12	neobsazená		



① ... ⑫ Čísla míst v zásobníku

V NC programu je naprogramováno následující volání nástroje:

```
N10 T=1
```

Toto volání se zpracuje následujícím způsobem:

1. Místo č. 1 v zásobníku je zkontrolováno a přitom se zjistí identifikátor nástroje.
2. Správa nástrojů rozpozná, že je nástroj zablokován, a tudíž že jej není možné použít.
3. V souladu s nastavenou strategií vyhledávání se spustí hledávání nástroje podle T="VRTAK": "Najdi aktivní nástroj; jinak vyber nástroj s nejbližším vyšším duplo-číslem."
4. Použitelný nástroj je nalezen:
   "VRTAK" duplo-č. 3 (na místě 4 v zásobníku)
   Tím je volba nástroje ukončena, načež se spustí výměna nástroje.

Poznámka

Když je aktivní strategie vyhledávání „Vezmi první použitelný nástroj dané skupiny", není posloupnost nástrojů definována. V tomto případě se vybere nástroj T10 **nebo** T1.

Podle strategie vyhledávání nástroje „Vezmi první nástroj se stavem "aktivní" ze skupiny" se bude výměna týkat nástroje T1.

3.4.1.2 Výměna nástroje, když je aktivní aktivní správa nástrojů s příkazem M6

Naprogramováním příkazu T je nástroj vybrán. Aktivním se nástroj stane až s příkazem M6 (včetně korekčních parametrů nástroje).

Použití

U obráběcích strojů s řetězovým, diskovým a plošným zásobníkem.

Syntaxe

Vyberte nástroj
```
T=<č.>
T=<název>
T<n>=<č.>
T<n>=<název>
```

Výměna nástroje
```
M6
```

Deaktivování nástroje
```
T0
```

Význam

T=:	Příkaz pro volbu nástroje Pokud jde o přiřazení hodnoty, existují následující možnosti:	
	<č.>:	Číslo místa v zásobníku
	<název>:	Název nástroje **Poznámka:** Při programování názvů nástrojů je nutno dbát na správný způsob zápisu (velká a malá písmena jsou rozlišována).
<n>:	Číslo vřetena jako rozšíření adresy **Poznámka:** Možnost programovat číslo vřetena jako rozšíření adresy závisí na konfiguraci stroje (--> věnujte prosím pozornost informacím od výrobce stroje).	
M6:	M-funkce pro výměnu nástroje (podle normy DIN 66025) Prostřednictvím příkazu M6 se aktivují vybraný nástroj (T...) a korekční parametry nástroje (D...).	
T0:	Deaktivování nástroje (místo v zásobníku není obsazeno)	

Příklad

Programový kód	**Komentář**
N10 T=1 M6	; výměna nástroje z místa v zásobníku č. 1
N20 D1	; aktivování korekce délky nástroje
N30 G1 X10 ...	; pracuje se s nástrojem T=1
...	
N70 T="VRTAK"	; předvolba nástroje s názvem "VRTAK"
N80 ...	; pracuje se s nástrojem T=1
...	
N100 M6	; výměna nástroje T="VRTAK".

Programový kód	Komentář
N140 D1 G1 X10 ...	; pracuje se s nástrojem T="VRTAK".
...	

3.4.2 Výměna nástroje, když správa nástrojů není aktivní

3.4.2.1 Výměna nástroje pomocí T-čísla, když správa nástrojů není aktivní

Naprogramováním T-čísla se uskuteční přímá výměna nástroje.

Použití

U soustruhů s revolverovým zásobníkem.

Syntaxe

Vyberte nástroj
```
T<č.>
T=<č.>
T<n>=<č.>
```

Deaktivování nástroje
```
T0
T0=<č.>
```

Význam

T:	Adresa pro aktivování nástroje, včetně výměny nástroje a aktivování korekčních parametrů nástroje	
<č.>:	Číslo nástroje	
	Rozsah hodnot:	0 ... 32000
<n>:	Číslo vřetena jako rozšíření adresy **Poznámka:** Možnost programovat číslo vřetena jako rozšíření adresy závisí na konfiguraci stroje (--> věnujte prosím pozornost informacím od výrobce stroje).	
T0:	Deaktivování aktivního nástroje	

Příklad

Programový kód	Komentář
N10 T1 D1	; výměna a upnutí nástroje T1 a aktivování korekčních parametrů nástroje D1.
...	
N70 T0	; deaktivování nástroje T1

Programový kód	Komentář
...	

3.4.2.2 Výměna nástroje pomocí M6, když správa nástrojů není aktivní

Naprogramováním příkazu T je nástroj vybrán. Aktivním se nástroj stane až s příkazem M6 (včetně korekčních parametrů nástroje).

Použití

U obráběcích strojů s řetězovým, diskovým a plošným zásobníkem.

Syntaxe

Vyberte nástroj
```
T<č.>
T=<č.>
T<n>=<č.>
```

Výměna nástroje
```
M6
```

Deaktivování nástroje
```
T0
T0=<č.>
```

Význam

T:	Příkaz pro volbu nástroje	
<č.>:	Číslo nástroje	
	Rozsah hodnot:	0 ... 32000
<n>:	Číslo vřetena jako rozšíření adresy **Poznámka:** Možnost programovat číslo vřetena jako rozšíření adresy závisí na konfiguraci stroje (--> věnujte prosím pozornost informacím od výrobce stroje).	
M6:	M-funkce pro výměnu nástroje (podle normy DIN 66025) Prostřednictvím příkazu M6 se aktivují vybraný nástroj (T...) a korekční parametry nástroje (D...).	
T0:	Deaktivování aktivního nástroje	

Příklad

Programový kód	Komentář
N10 T1 M6	; výměna a upnutí nástroje T1
N20 D1	; aktivování korekce délky nástroje
N30 G1 X10 ...	; pracuje se s nástrojem T1.

Programový kód	Komentář
...	
N70 T5	; předvolba nástroje T5.
N80 ...	; pracuje se s nástrojem T1.
...	
N100 M6	; výměna a upnutí nástroje T5.
N110 D1 G1 X10 ...	; pracuje se s nástrojem T5
...	

3.4.3 Chování v případě nesprávně naprogramovaného T-čísla

Chování případě nesprávně naprogramovaného T-čísla závisí na konfiguraci stroje:

MD22562 TOOL_CHANGE_ERROR_MODE		
Bit	**Hodnota**	**Význam**
7	0	Základní nastavení Při programování T-příkazů se okamžitě kontroluje, zda je T-číslo v NC systému známé. Pokud tomu tak není, aktivuje se alarm.
	1	Naprogramované T-číslo je kontrolováno až poté, co je zvoleno příslušné D-číslo. Pokud T-číslo není v NC systému známo, bude alarm aktivován až s volbou D-čísla. Toto chování je požadováno tehdy, když naprogramování T-čísla má např. ovlivňovat také polohování a pro tyto účely nemusí parametry nástroje existovat (revolverový zásobník).

3.5 Korekční parametry nástroje

3.5.1 Naprogramovaná kontura a dráha nástroje

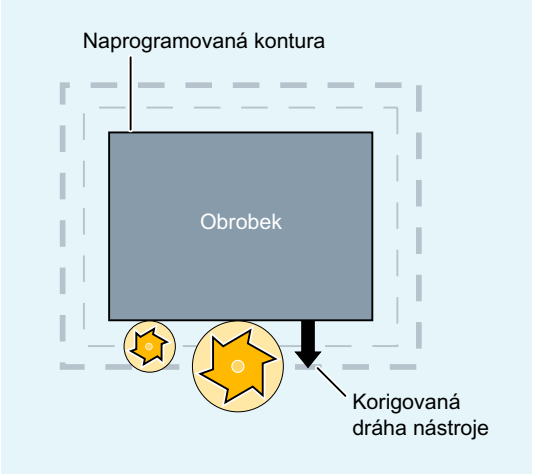
Do programu jsou přímo zadávány rozměry obrobku (např. podle výrobního výkresu). Na parametry nástroje, jako jsou průměr frézy, délka břitu soustružnického nože (levý/pravý nůž) a délka nástroje, se proto při sestavování programu vůbec nemusí brát v úvahu.

Řídící systém koriguje dráhu nástroje

Při výrobě obrobku jsou pohyby nástroje řízeny v závislosti na geometrii příslušného nástroje tak, aby s každým použitým nástrojem bylo možné naprogramovanou konturu vyrobit.

Aby mohl řídící systém vypočítat dráhu nástroje, musí být jeho parametry uloženy v paměti korekčních parametrů nástrojů v řídícím systému. Prostřednictvím NC programu jsou vyvolávány potřebný nástroj (T...) a k němu patřící blok korekčních parametrů nástroje (D...).

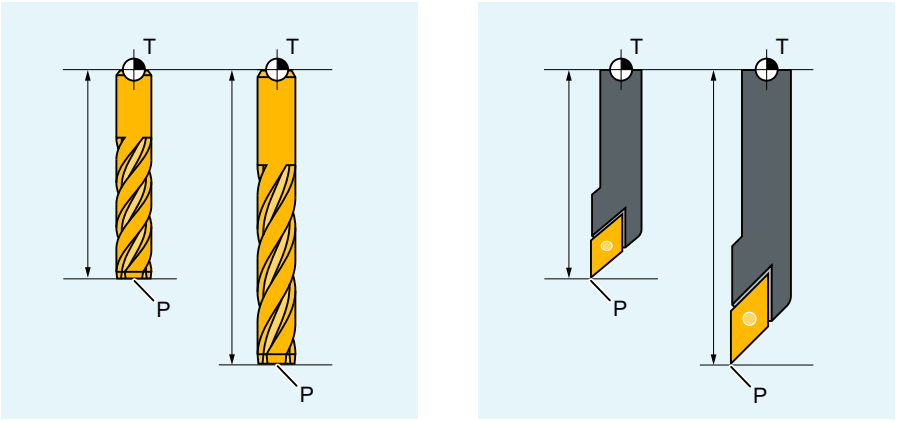
Řídící systém v průběhu zpracování programu vybírá potřebné korekční parametry z paměti korekčních parametrů nástrojů a v závislosti na různých charakteristikách nástrojů provádí individuální korekce dráhy nástroje.



3.5.2 Korekce délky nástroje

Pomocí korekce délky nástroje se vyrovnávají rozdíly v délce používaných nástrojů.

Za délku nástroje se považuje vzdálenost mezi vztažným bodem držáku pro upnutí nástroje a špičkou nástroje:



T Vztažný bod držáku nástroje

P Špička nástroje

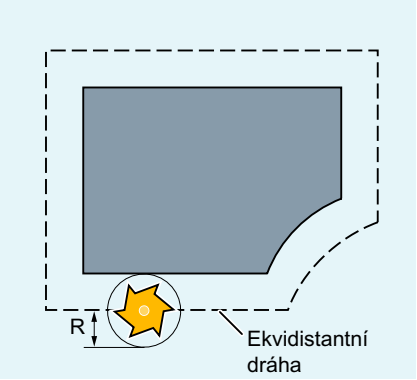
Tyto délky se měří a spolu s definovatelnými hodnotami opotřebení se ukládají do paměti korekčních parametrů nástrojů v řídícím systému. Řídící systém odtud vypočítává pohybové vzdálenosti ve směru přísuvu.

Poznámka

Hodnota korekce délky nástroje je závislá na jeho prostorové orientaci.

3.5.3 Korekce rádiusu nástroje

Kontura a dráha nástroje nejsou identické. Střed frézy, příp. střed břitu se musí v závislosti na rádiusu nástroje pohybovat po ekvidistantní dráze vedle kontury (dráha středu nástroje). Za tím účelem řídící systém v průběhu zpracovávání programu posouvá naprogramovanou dráhu středu nástroje podle rádiusu tohoto aktivního nástroje (paměť korekčních parametrů nástroje) tak, aby se jeho břit pohyboval přesně po naprogramované kontuře.



R Rádius nástroje

S Střed břitu

Korekce rádiusu nástroje je podrobně popsána v kapitole "Korekce rádiusu nástroje (Strana 253)".

Viz také

2 1/2 D-korekce nástroje (CUT2D, CUT2DD, CUT2DF, CUT2DFD) (Strana 289)

3.5.4 Paměť korekčních parametrů nástrojů

V paměti korekčních parametrů nástrojů v řídícím systému musí být pro každý břit nástroje k dispozici následující údaje:

- Typ nástroje
- Poloha břitu
- Geometrické parametry nástroje (délka, rádius)

Tato data se zadávají jako parametry nástroje (max. 25). To, které parametry jsou pro daný nástroj zapotřebí, závisí na typu tohoto nástroje. Do nepotřebných parametrů nástroje je nutno dosadit nulu (což odpovídá předdefinovanému systémovému nastavení).

Poznámka

Hodnota, která byla jednou uložena do paměti korekčních parametrů, se při každém vyvolání nástroje započítává.

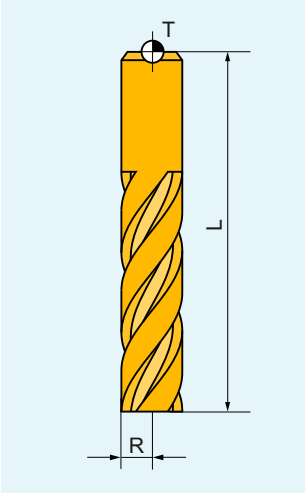
Typ nástroje

Typ nástroje (vrták, fréza nebo soustružnický nůž) určuje, které geometrické parametry jsou zapotřebí a jak budou započítávány.

Poloha břitu

Poloha břitu popisuje polohu špičky nástroje vzhledem ke středu břitu. Poloha břitu je spolu s rádiusem břitu zapotřebí pro výpočet korekce rádiusu nástroje u soustružnických nástrojů (typ nástroje 5xx) (Strana 76).

Geometrické parametry nástroje (délka, rádius)



T	Vztažný bod držáku nástroje
R	Rádius nástroje
L	Délka nástroje

Geometrické parametry nástroje se skládají z několika komponent (geometrie, opotřebení). Řídící systém tyto komponenty připočítává k jedné z výsledných veličin (např. celková délka 1, výsledný rádius). Příslušný výsledný rozměr je uplatňován při aktivování paměti korekčních parametrů.

Způsob, jakým se tyto hodnoty přepočítávají do jednotlivých os, je dán typem nástroje a právě zvolenou rovinou (G17/G18/G19).

3.5.5 Typy nástrojů

3.5.5.1 Číslo typu nástroje a skupiny nástrojů

Každému typu nástroje je přiřazeno jednoznačné 3-místné číslo. Prostřednictvím první číslice (místo stovek) se uskutečňuje přiřazení nástroje k jedné z následujících technologií, příp. ke skupině nástrojů:

Typ nástroje	Skupina nástrojů
1xy	Fréza (Strana 71)
2xy	Vrták (Strana 73)
3xy	rezervováno
4xy	Brusné nástroje (Strana 74)
5xy	Soustružnické nástroje (Strana 76)
6xy	rezervováno
7xy	Speciální nástroje (Strana 78)

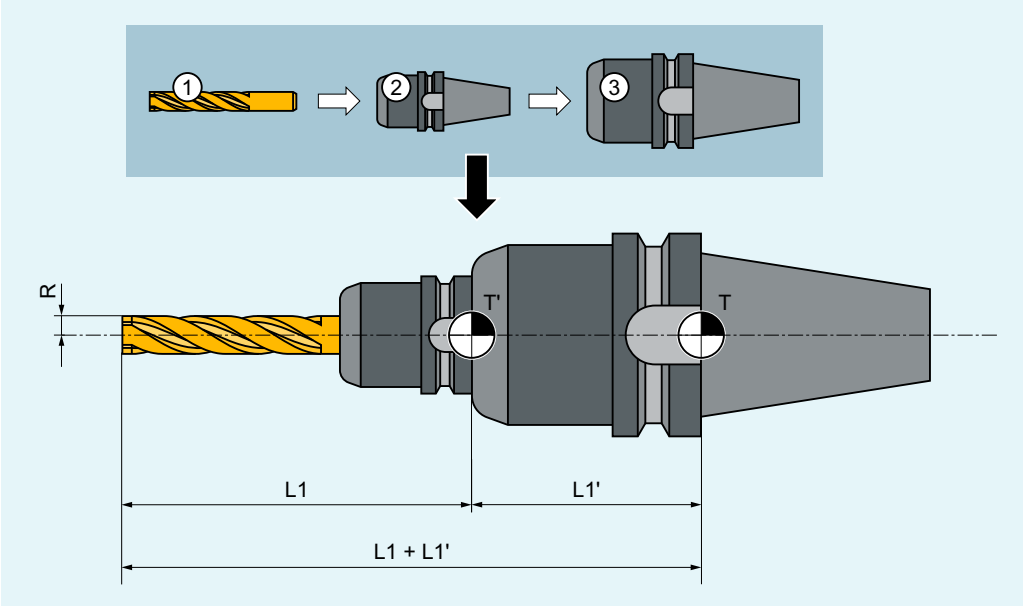
3.5.5.2 frézovací nástroje

V rámci skupiny nástrojů s názvem "Frézovací nástroje" existují následující typy nástrojů:

100	Frézovací nástroj podle CLDATA (Cutter Location Data)
110	Fréza s kulovou hlavou
111	Válcová zápustková fréza
120	Stopková fréza bez zaoblení rohů
121	Stopková fréza se zaoblením rohů
130	Fréza s úhlovou hlavou bez zaoblení rohů
131	Fréza s úhlovou hlavou se zaoblením rohů
140	Nástroj pro rovinné frézování
145	Závitová fréza
150	Kotoučová fréza
151	Pila
155	Fréza ve tvaru komolého kužele bez zaoblení rohů
156	Kuželová komolá fréza se zaoblením rohů
157	Kuželová zápustková fréza
160	Fréza na vrtání závitů

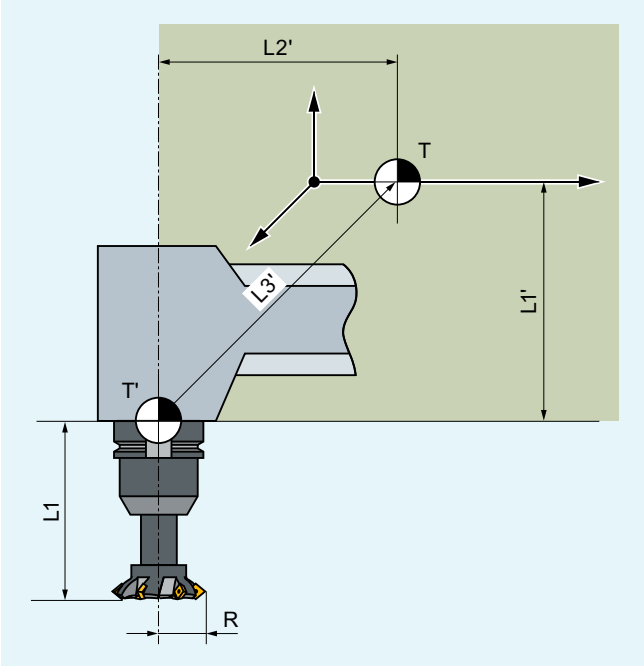
Parametry nástroje

Následující obrázky poskytují přehled o tom, které parametry nástroje se u frézovacích nástrojů ukládají do paměti korekčních parametrů:



①	Nástroj
②	Držák nástroje
③	Adaptér nástroje
T	Vztažný bod adaptéru (v případě upnutého nástroje = vztažný bod držáku nástroje)
T'	Vztažný bod držáku nástroje
L1	Geometrie - délka 1
L1'	Rozměr adaptéru - délka 1
L1 + L1'	Celková délka L1
R	Rádius

Parametry nástroje	Význam
$TC_DP1	Typ nástroje 1xy
$TC_DP3	Geometrie - délka 1
$TC_DP6	Geometrie - rádius
$TC_DP21	Rozměr adaptéru - délka 1
• Hodnoty opotřebení v souladu s požadavky. • Zbývající hodnoty je potřeba nastavit na 0.	



T Vztažný bod držáku nástroje
T' Vztažný bod držáku nástroje
L1 Geometrie - délka 1
R Rádius nástroje
L1' Základní rozměr - délka 1
L2' Základní rozměr - délka 2
L3' Základní rozměr - délka 3

Parametry nástroje	Význam
$TC_DP1	Typ nástroje
$TC_DP3	Geometrie - délka 1
$TC_DP6	Geometrie - rádius
$TC_DP21	Základní rozměr - délka 1
$TC_DP22	Základní rozměr - délka 2
$TC_DP23	Základní rozměr - délka 3
• Hodnoty opotřebení v souladu s požadavky. • Zbývající hodnoty je potřeba nastavit na 0.	

3.5.5.3 Vrták

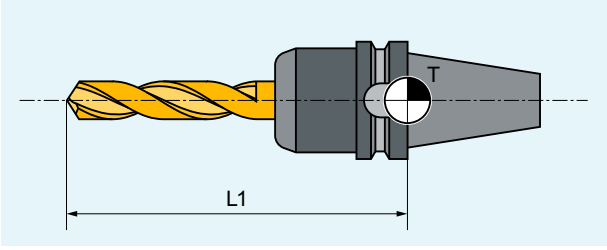
V rámci skupiny nástrojů s názvem "Vrták" existují následující typy nástrojů:

Č.	Typ nástroje
200	Spirální vrták
205	Vrták na vrtání zplna
210	Vrtací tyč

Č.	Typ nástroje
220	Středicí vrták
230	Kuželový záhlubník
231	Plochý záhlubník
240	Závitník pro normální závit
241	Závitník pro jemný závit
242	Závitník pro Withworthův závit
250	Výstružník

Parametry nástroje

Následující obrázek poskytuje přehled o tom, které parametry nástroje se u vrtáků ukládají do paměti korekčních parametrů:



T Vztažný bod držáku nástroje

L1 Délka 1

Parametry nástroje	Význam
$TC_DP1	Typ nástroje
$TC_DP3	Geometrie - délka 1
• Hodnoty opotřebení v souladu s požadavky. • Zbývající hodnoty je potřeba nastavit na 0.	

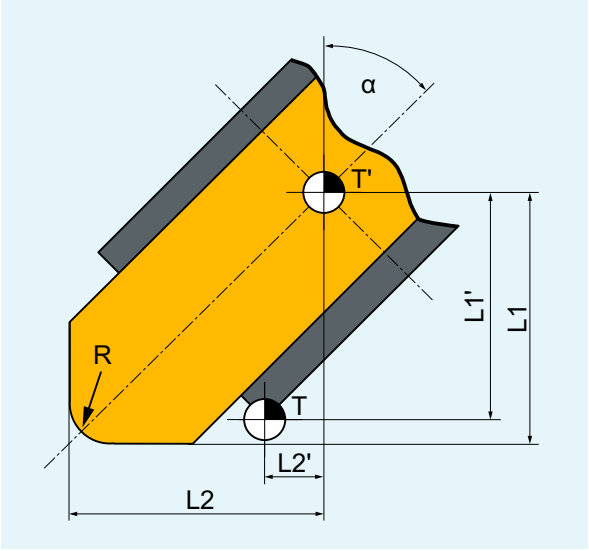
3.5.5.4 Brusné nástroje

V rámci skupiny nástrojů s názvem "Brusné nástroje" existují následující typy nástrojů:

400	Obvodový brusný kotouč
401	Obvodový brusný kotouč s monitorováním
402	Obvodový brusný kotouč bez monitorování a bez rozměru základny
403	Obvodový brusný kotouč s monitorováním a bez rozměru základny pro obvodovou rychlost brusného kotouče (GWPS)
410	Čelní brusný kotouč
411	Čelní brusný kotouč s monitorováním
412	Čelní brusný kotouč bez monitorování
413	Čelní brusný kotouč s monitorováním a bez rozměru základny pro obvodovou rychlost brusného kotouče (GWPS)
490	Orovnávač

Parametry nástroje

Následující obrázek poskytuje přehled o tom, které parametry nástroje se u brusných nástrojů ukládají do paměti korekčních parametrů:



T	Vztažný bod držáku nástroje
T'	Vztažný bod držáku nástroje
L1	Geometrie - délka 1
L1'	Základní rozměr - délka 1
L2	Geometrie - délka 2
L2'	Základní rozměr - délka 2
R	Rádius
α	Úhel šikmého kotouče

Specifické parametry břitů	Význam
$TC_DP1	Typ nástroje 4xy
$TC_DP2	Poloha břitu
$TC_DP3	Geometrie - délka 1
$TC_DP4	Geometrie - délka 2
$TC_DP6	Rádius
$TC_DP21	Základní rozměr - délka 1
$TC_DP22	Základní rozměr - délka 2
• Hodnoty opotřebení v souladu s požadavky. • Zbývající hodnoty je potřeba nastavit na 0.	

Parametry specifické pro daný nástroj	Význam
$TC_TPG1	Číslo vřetena
$TC_TPG2	Pravidlo pro zřetězení [1)]
$TC_TPG3	Minimální rádius kotouče

Parametry specifické pro daný nástroj	Význam
$TC_TPG4	Minimální šířka kotouče
$TC_TPG5	Momentální šířka kotouče
$TC_TPG6	Maximální otáčky
$TC_TPG7	Maximální obvodová rychlost
$TC_TPG8	Úhel šikmého kotouče
$TC_TPG9	Číslo parametru pro výpočet rádiusu
$TC_TPG_DRSPATH	Cesta do adresáře, v němž je program pro orovnávání
$TC_TPG_DRSPROG	Název programu pro orovnávání

[1] Délkové korekční parametry geometrie, opotřebení a základní rozměr mohou být zřetězeny pro levou a pravou korekci kotouče, tzn. pokud se změní korekce délky pro levý břit, budou tyto hodnoty automaticky zaznamenány také pro pravou stranu a naopak.

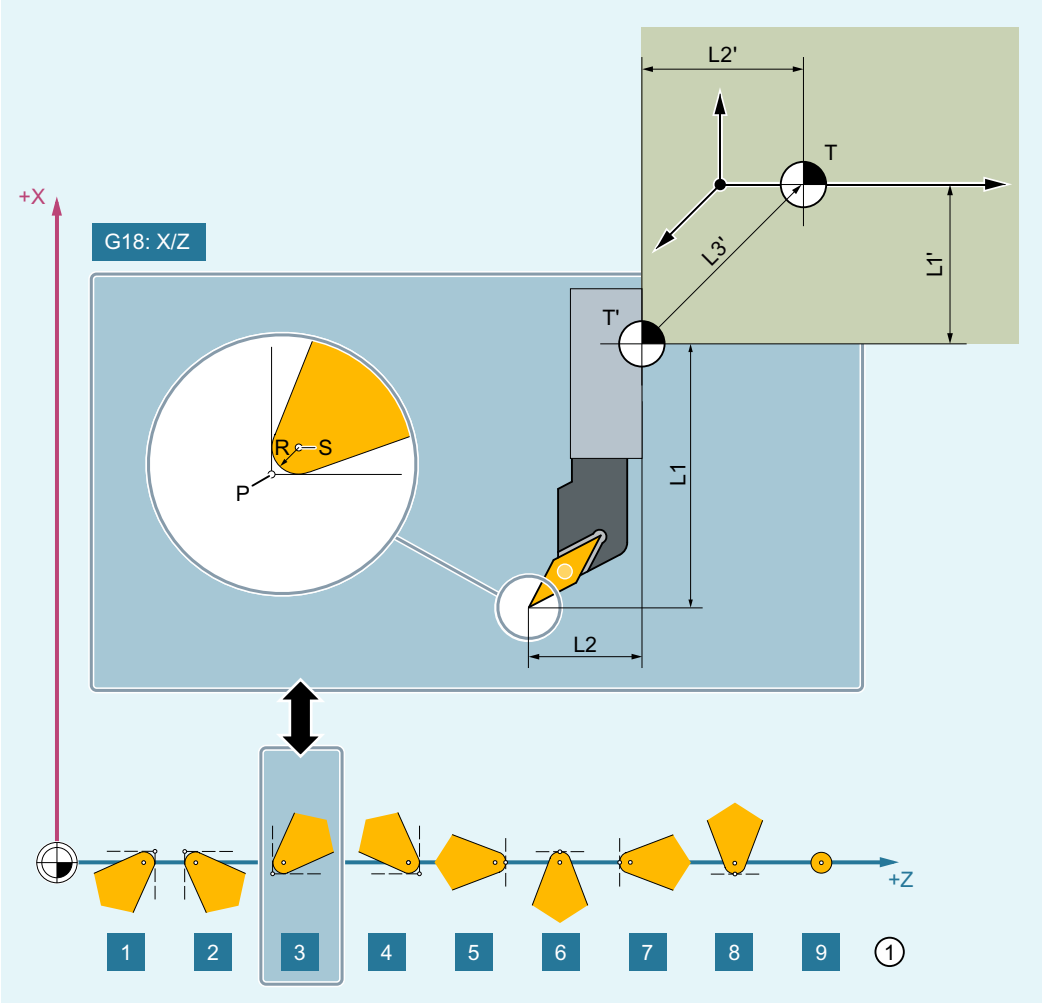
3.5.5.5 Soustružnické nástroje

V rámci skupiny nástrojů s názvem "Soustružnické nástroje" existují následující typy nástrojů:

500	Hrubovací nůž
510	Nůž pro obrábění načisto
520	Nůž na zápichy
530	Upichovací nůž
540	Nůž na soustružení závitů
550	Tvarový nůž
560	Vrták (ECOCUT)
580	Měřicí sonda s parametrem polohy břitu

Parametry nástroje

Následující obrázek poskytuje přehled o tom, které parametry nástroje se u soustružnických nástrojů ukládají do paměti korekčních parametrů:



①	Poloha břitu (1 - 9) při obrábění za středem otáčení
P	Špička nástroje
S	Střed břitu
R	Rádius břitu
T	Vztažný bod držáku nástroje
T'	Vztažný bod držáku nástroje
L1	Geometrie - délka 1
L2	Geometrie - délka 2
L1'	Základní rozměr - délka 1
L2'	Základní rozměr - délka 2
L3'	Základní rozměr - délka 3

Parametry nástroje	Význam
$TC_DP1	Typ nástroje
$TC_DP2	Poloha břitu
$TC_DP3	Geometrie - délka 1
$TC_DP4	Geometrie - délka 2
$TC_DP6	Geometrie - rádius (rádius břitu)
$TC_DP21	Základní rozměr - délka 1
$TC_DP22	Základní rozměr - délka 2
$TC_DP23	Základní rozměr - délka 3
• Hodnoty opotřebení v souladu s požadavky. • Zbývající hodnoty je potřeba nastavit na 0.	

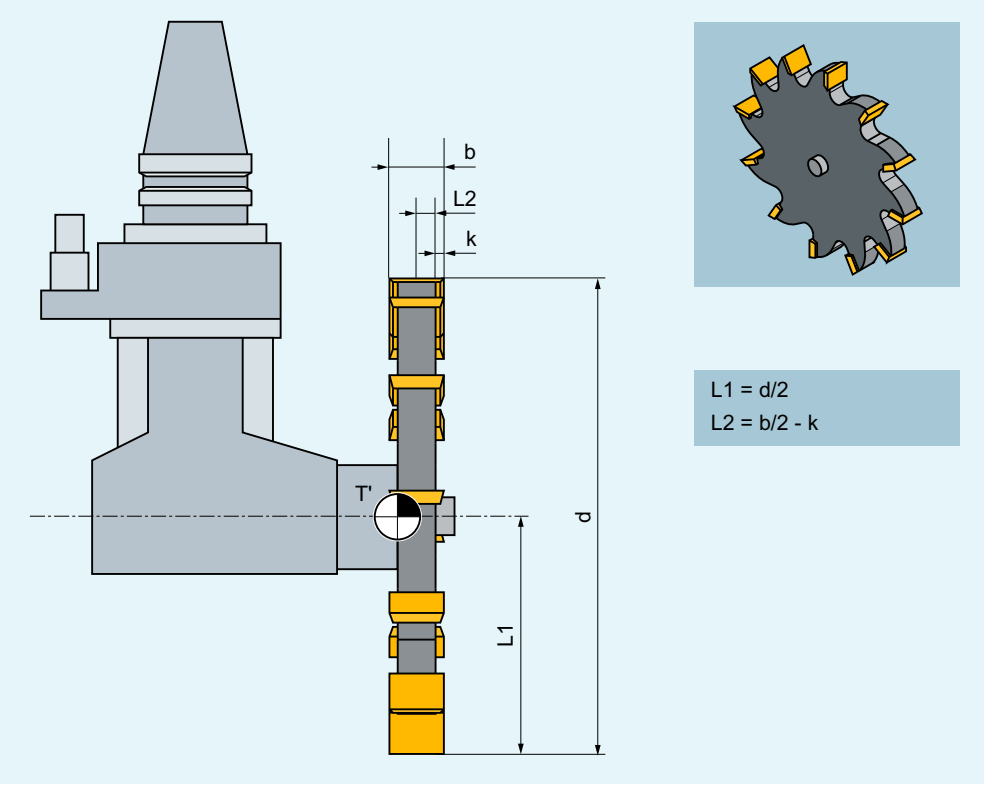
3.5.5.6 Speciální nástroje

V rámci skupiny nástrojů s názvem "Speciální nástroje" existují následující typy nástrojů:

700	Drážková pila
710	3D měřicí sonda
711	Hranová sonda
712	Jednosměrná sonda
713	L-sonda
714	Sonda ve tvaru hvězdy
725	Kalibrační nástroj
730	Doraz
731	Pinola
732	Luneta

Parametry nástroje

Následující obrázek poskytuje přehled o tom, které parametry nástroje se u typu nástroje "Drážková pila" ukládají do paměti korekčních parametrů:



T'	Vztažný bod držáku nástroje
L1	Geometrie - délka 1
L2	Geometrie - délka 2
d	Průměr
b	Šířka drážky
k	Přesah

Parametry nástroje	Význam
$TC_DP1	Typ nástroje
$TC_DP3	Geometrie - délka 1
$TC_DP4	Geometrie - délka 2
$TC_DP6	Průměr
$TC_DP7	Šířka drážky
$TC_DP8	Přesah
$TC_DP21	Základní rozměr - délka 1
$TC_DP22	Základní rozměr - délka 2
$TC_DP23	Základní rozměr - délka 3
• Hodnoty opotřebení v souladu s požadavky. • Zbývající hodnoty je potřeba nastavit na 0.	

3.5.6 Aktivování/deaktivování korekčních parametrů nástroje (D, D0)

Břitům 1 až 8 nástroje (v případě aktivní správy nástrojů 12) mohou být přiřazeny různé bloky korekčních parametrů nástroje (např. odlišné hodnoty korekčních parametrů pro levý a pravý břit nože pro soustružení zápichů).

Aktivování korekčních parametrů (tzn. údajů pro korekci délky nástroje) určitého speciálního břitu se uskutečňuje voláním D-čísla. Je-li naprogramováno D0, jsou korekční parametry daného nástroje deaktivovány.

Kromě toho musí být pomocí příkazů G41/G42 aktivována korekce rádiusu nástroje.

Poznámka

Korekce délky nástroje je v platnosti, když je naprogramováno odpovídající D-číslo. Pokud není žádné D-číslo naprogramováno, při výměně nástroje se aktivuje standardní nastavení definované pomocí strojního parametru (→ viz informace od výrobce stroje).

Syntaxe

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

Význam

D:	Adresa pro aktivování bloku korekčních parametrů nástroje pro momentálně vybraný nástroj Korekce délky nástroje se bude uplatňovat od prvního naprogramovaného posuvu příslušné osy délkové korekce. **Upozornění:** Korekce délky nástroje se uplatňuje i tehdy, pokud nebylo naprogramováno žádné D-číslo, a v případě, že je v konfiguraci pro výměnu nástroje nastaveno automatické aktivování břitu nástroje (→ viz informace od výrobce stroje).	
<č.>:	Prostřednictvím D-čísla se zadává blok korekčních parametrů nástroje, který má být aktivován. Způsob programování D-čísel je závislý na konfiguraci obráběcího stroje (→ viz údaje od výrobce stroje). Existují následující možnosti: • D-číslo = číslo břitu Ke každému nástroji T<č.> (bez správy nástrojů), příp. T="název" (se správou nástrojů) existují D-čísla od 1 až do max. 12. Tato D-čísla jsou přiřazena přímo břitům nástroje. Každému D-číslu (= číslu břitu) patří jeden blok korekčních parametrů ($TC_DPx[<t>,<d>]). • Libovolná volba D-čísel D-čísla mohou být přiřazována číslům břitů nástroje libovolně. Nejvyšší možná hodnota D-čísel, která se mohou používat, je definována strojním parametrem.	
	Rozsah hodnot:	0 - 32000
D0:	Deaktivování bloku korekčních parametrů nástroje pro momentálně vybraný nástroj	
G41:	Příkaz pro aktivování korekce rádiusu nástroje se směrem obrábění **vlevo** od kontury.	
G42:	Příkaz pro aktivování korekce rádiusu nástroje se směrem obrábění **vpravo** od kontury.	
G40:	Příkaz pro deaktivování korekce rádiusu nástroje	

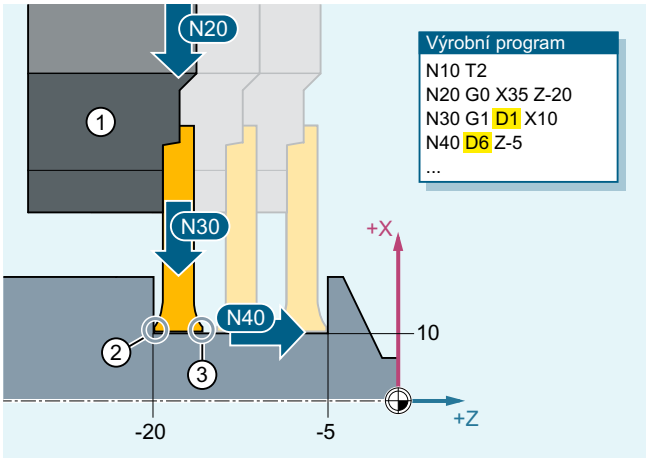
Pokud budete potřebovat další informace týkající se příkazů G40/G41/G42, viz kapitola „Korekce rádiusu nástroje (Strana 253)".

Příklady

Příklad 1: Výměna nástroje pomocí příkazu T (soustružení)

```
Programový kód          Komentář
N10 T1 D1               ; výměna a upnutí nástroje T1 a aktivování bloku korekčních
                          parametrů D1 nástroje T1
N11 G0 X... Z...        ; najíždění na délkovou korekci
N50 T4 D2               ; výměna a upnutí nástroje T4 a aktivování bloku korekčních
                          parametrů D2 nástroje T4
...
N70 G0 Z... D1          ; aktivování jiného břitu D1 nástroje T4
```

Příklad 2: Odlišné hodnoty korekčních parametrů pro levý a pravý břit zápichového nože



① Nůž pro soustružení zápichů (T2)
② Břit D1
③ Břit D6

Viz také

Korekce rádiusu nástroje (Strana 69)

Další informace

Změna hodnot korekčních parametrů nástroje

Při standardním nastavení vstoupí změny hodnot korekčních parametrů nástroje v platnost až po opětovném naprogramování T- nebo D-čísla.

Prostřednictvím následujícího strojního parametru je možno nastavit, aby zadané korekční parametry nástroje vstoupily v platnost okamžitě:

MD9440 $MM_ACTIVATE_SEL_USER

⚠ VÝSTRAHA
Nebezpečí kolize Jestliže je parametr MD9440 nastaven, potom budou korekční parametry nástroje, které vyplývají ze změn hodnot korekčních parametrů nástrojů provedených během zastavení výrobního programu, aplikovány v okamžiku opětovného spuštění zpracování výrobního programu.

3.5.7 Potlačení korekcí nástroje (SUPD)

Pomocí příkazu SUPD můžete v daném bloku potlačit korekce nástroje.

Výhoda oproti deaktivování korekčních parametrů nástroje pomocí příkazu D0 spočívá v tom, že v případě příkazu SUPD zůstávají informace o korekcích nástroje zachovány, takže opětovné aktivování korekčních parametrů nástroje naprogramováním D-čísla odpadá.

Příkaz SUPD nachází uplatnění výlučně pro potlačení korekce délky nástroje, např. v blocích výrobního programu, ve kterých se má započítávat nulový bod, nikoli ale délka nástroje. Když jsou aktivní příkazy G41/G42, používat tento příkaz pro dodatečné potlačení korekce rádiusu nástroje se nedoporučuje.

Syntaxe

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

Význam

SUPD:	Příkaz G-funkce pro deaktivování bloku korekčních parametrů nástroje pro momentálně vybraný nástroj v aktivním bloku	
	Platnost:	bloková

Okrajové podmínky

- Příkaz SUPD by se měl používat pouze v lineárních blocích.
- Příkaz SUPD se nemůže používat v synchronních akcích.
- Příkaz SUPD se nemůže používat ve spojení s následujícími funkcemi:
  - 3D-korekce rádiusu nástroje pro 3D frézování na čelní ploše (CUT3DFxx)
  - Tabulky křivek (CTAB)
- Použití příkazu SUPD, když je aktivní korekce rádiusu nástroje (G41/G42), je možné, ale nedoporučuje se.
  Pokud má být funkce přesto použita, musí být následujícímu nastavovanému parametru dosazena hodnota 0:
  SD42480 $SC_STOP_CUTCOM_STOPRE = 0
  Tím se zabrání, aby při aktivním příkazu G41/G42 docházelo k přerušení zpracování programu.

Příklad

V podprogramu SUB_SUP má být při posuvech potlačena délka nástroje.

Výrobní program

Programový kód	**Komentář**
...	

Programový kód	Komentář
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**	; Volání podprogramu
N370 G0 X1200	
N380 M30	

Podprogram s D0

Programový kód	Komentář
N10 PROC **SUB_SUP**	
N20 DEF INT NUMBER	
N30 NUMBER=$P_TOOL	
N40 G0 Z49 **D0**	; Deaktivování korekcí nástroje
N50 D=NUMBER	; Opětovné aktivování korekcí nástroje
N60 RET	

Podprogram se SUPD

Programový kód	Komentář
N10 PROC **SUB_SUP**	
N40 G0 Z49 **SUPD**	; Potlačení korekcí nástroje v aktivním bloku.
N60 RET	

3.5.8 Programovatelný offset korekce nástroje (TOFFL, TOFF, TOFFR, TOFFLR)

Prostřednictvím adres TOFFx má uživatel možnost změnit efektivní délku nástroje a efektivní rádius nástroje v NC programu, aniž by došlo ke změně hodnot korekčních parametrů nástroje uložených v paměti těchto parametrů.

Na konci programu budou tyto programovatelné offsety opět vymazány.

Syntaxe

Offset délky nástroje

```
TOFFL=<hodnota>
TOFFL[1]=<hodnota> TOFFL[2]=<hodnota> TOFFL[3]=<hodnota>
TOFF[<geometrická osa>]=<hodnota>
```

Délku nástroje je možné změnit u všech tří komponent současně. V jednom bloku se ale nesmí současně použít na jedné straně příkazy ze skupiny TOFFL/TOFFL[1..3] a na druhé straně příkazy

ze skupiny TOFF[<geometrická osa>]. Rovněž nesmí být napsány v jednom bloku současně příkazy TOFFL a TOFFL[1].

Jestliže nejsou v jednom bloku naprogramovány všechny tři délkové komponenty nástroje, zůstanou nenaprogramované komponenty nezměněny. Díky tomu je možné blok po bloku upravovat korekce pro větší počet komponent. To však platí jen do té doby, než jsou komponenty nástroje modifikovány buď pouze příkazem TOFFL nebo jen příkazem TOFF. Změna varianty programování z TOFFL na TOFF nebo obráceně napřed vymaže všechny eventuálně předtím naprogramované offsety délky nástroje (viz příklad 3).

Offset rádiusu nástroje

```
TOFFR=<hodnota>
```

Simultánní zadání offsetu délky a rádiusu nástroje

```
TOFFLR=<hodnota>
```

Význam

TOFFL:	Korekce efektivní délky nástroje Příkaz TOFFL může být naprogramován buď s indexem nebo bez něj:	
	TOFFL=...	Naprogramovaná hodnota offsetu se započítává ve směru, ve kterém se uplatňuje také délkový komponent **L1** nástroje uložený v paměti korekčních parametrů. Funkce příkazů TOFFL a TOFFL[1] je identická.
	TOFFL[1]=... TOFFL[2]=... TOFFL[3]=...	Naprogramovaná hodnota offsetu se započítává ve směru, ve kterém se uplatňuje také délkový komponent **L1**, **L2**, příp. **L3** nástroje uložený v paměti korekčních parametrů.
	Poznámka: Způsob, jakým se hodnoty korekcí délek nástroje přepočítávají do jednotlivých os, je dán typem nástroje a právě zvolenou pracovní rovinou (G17/G18/G19).	
TOFF:	Korekce délky nástroje ve složce, která je rovnoběžná se zadanou geometrickou osou. Příkaz TOFF se uplatňuje ve směru délkového komponentu nástroje, který je v případě neotočeného nástroje (orientovatelný držák nástroje, příp. transformace orientace) rovnoběžně s geometrickou osou udanou v indexu. **Poznámka:** Přiřazení naprogramovaných hodnot délkovým komponentům nástroje není framem nijak ovlivňováno, tzn. pro přiřazení délkového komponentu nástroje geometrickým osám se jako základ používá souřadný systém nástroje v základní poloze nástroje a nikoli souřadný systém obrobku (WCS).	
<GeoAx>:	Identifikátor geometrické osy	
TOFFR:	Korekce efektivního rádiusu nástroje Příkaz TOFFR mění efektivní rádius nástroje o naprogramovanou hodnotu offsetu tehdy, pokud je **aktivní korekce rádiusu nástroje**.	
TOFFLR:	Korekce efektivní délky nástroje ve složce L1 **a** efektivní rádius nástroje **Poznámka:** U nástrojů se zaoblením v rohu (typy 111, 121, 131 a 156) provádí příkaz TOFFLR korekci také rádiusu v rohu.	

<hodnota>:	Hodnota offsetu	
	Typ:	REAL

Příklady

Příklad 1: Kladný offset délky nástroje

Aktivním nástrojem je vrták s délkou L1 = 100 mm.

Aktivní rovinou je rovina G17. Tzn. vrták je špičkou obrácen ve směru osy Z.

Efektivní délka vrtáku má být prodloužena o 1 mm. Pro naprogramování tohoto offsetu délky nástroje máte k dispozici následující varianty:

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

Příklad 2: Záporný offset délky nástroje

Aktivním nástrojem je vrták s délkou L1 = 100 mm.

Aktivní rovinou je rovina G18. Tzn. vrták je špičkou obrácen ve směru osy Y.

Efektivní délka vrtáku má být zkrácena o 1 mm. Pro naprogramování tohoto offsetu délky nástroje máte k dispozici následující varianty:

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

Příklad 3: Změna varianty programování z TOFFL na TOFF

Aktivním nástrojem je frézovací nástroj. Aktivní rovinou je rovina G17.

Programový kód	Komentář
N10 TOFFL[1]=3 TOFFL[3]=5	; Platné offsety: L1=3, L2=0, L3=5
N20 TOFFL[2]=4	; Platné offsety: L1=3, L2=4, L3=5
N30 TOFF[Z]=1.3	; Platné offsety: L1=0, L2=0, L3=1.3

Příklad 4: Přiřazení hodnot offsetu po změně roviny

Programový kód	Komentář
N10 $TC_DP1[1,1]=120	
N20 $TC_DP3[1,1]=100	; délka nástroje L1=100 mm
N30 T1 D1 G17	
N40 **TOFF[Z]=1.0**	; offset ve směru osy Z (v případě G17 odpovídá L1)
N50 G0 X0 Y0 Z0	; poloha os stroje X0 Y0 Z101
N60 G18 G0 X0 Y0 Z0	; poloha os stroje X0 Y100 Z1
N70 G17	
N80 **TOFFL=1.0**	; offset ve směru L1 (v případě G17 odpovídá ose Z)
N90 G0 X0 Y0 Z0	; poloha os stroje X0 Y0 Z101

Programový kód	Komentář
N100 G18 G0 X0 Y0 Z0	; poloha os stroje X0 Y101 Z0

V tomto příkladu zůstává při přepnutí na rovinu G18 v bloku N60 offset o velikosti 1 mm v ose Z zachován, efektivní délka nástroje ve směru osy Z odpovídá nezměněné délce nástroje 100 mm.

V bloku N100 se oproti tomu při přepnutí do roviny G18 uplatňuje offset v ose Y, protože byl při programování přiřazen délce nástroje L1 a protože v rovině G18 se tento délkový komponent uplatňuje v ose Y.

Příklad 5: Simultánní zadání offsetu délky a rádiusu nástroje

a, stopková fréza **bez** zaoblení rohů (typ nástroje 120):

Programový kód	Komentář
...	
TOFFLR=0.1	; Platné offsety:
	; Offset délky nástroje (L1) = 5
	; Offset rádiusu nástroje = 5
...	

b, stopková fréza **se** zaoblením rohů (typ nástroje 121):

Programový kód	Komentář
...	
TOFFLR=0.1	; Platné offsety:
	; Offset délky nástroje (L1) = 0.1
	; Offset rádiusu nástroje = 0.1
	; Offset rádiusu v rohu = 0.1
...	

Další informace

Offsety délek nástroje

Programovatelné offsety délky nástroje jsou v závislosti na způsobu naprogramování přiřazovány buď délkovým komponentům nástroje L1, L2 a L3 uloženým v paměti korekčních parametrů (TOFFL) nebo geometrickým osám (TOFF). V případě změny roviny (G17/G18/G19 <---> G17/G18/G19) se s naprogramovanými offsety zachází odpovídajícím způsobem:

- Pokud jsou hodnoty offsetu přiřazeny délkovým komponentům nástroje, jsou směry, ve kterých se naprogramované offsety uplatňují, odpovídajícím způsobem vyměněny.
- Pokud jsou hodnoty offsetu přiřazeny geometrickým osám, pak změna roviny nijak přiřazení vztažené k osám souřadného systému nemění.

Při přiřazování naprogramované hodnoty offsetu délkovému komponentu nástroje jsou vyhodnocovány následující nastavované parametry:

SD42940 $SC_TOOL_LENGTH_CONST (změna délkového komponentu nástroje při změně roviny)

SD42950 $SC_TOOL_LENGTH_TYPE (přiřazení kompenzace délky nástroje nezávisle na typu nástroje)

Jestliže mají tyto nastavované parametry platnou hodnotu nerovnající se 0, potom mají přednost před obsahem příkazů ze skupiny G-funkcí 6 (volba roviny G17/G18/G19), příp. před typem nástroje obsaženým v parametrech nástroje ($TC_DP1[<T-číslo>, <D-číslo>]), což znamená, že tyto nastavované parametry ovlivňují vyhodnocování offsetu stejným způsobem jako délkové komponenty nástroje L1 až L3.

Offset rádiusu nástroje

Adresa TOFFR má téměř stejný efekt jako adresa OFFN (Strana 253). Rozdíl se vyskytne pouze tehdy, když je aktivní transformace křivky na válcovém plášti (TRACYL) a když je aktivní korekce stěny drážky. V tomto případě se příkaz OFFN uplatňuje na rádius nástroje se záporným znaménkem, zatímco příkaz TOFFR oproti tomu s kladným znaménkem.

Příkazy OFFN a TOFFR mohou být v platnosti i současně. Potom se zpravidla chovají aditivně (s výjimkou korekce stěny drážky).

Výměna nástroje

Všechny hodnoty offsetu zůstávají při výměně nástroje (změně břitu) zachovány, tzn. jsou v platnosti rovněž v případě nového nástroje (nového břitu).

Systémové proměnné pro načítání aktuální hodnoty offsetu

Momentálně platné hodnoty offsetů je možno načítat pomocí následujících systémových proměnných:

Systémové proměnné	Význam
$P_TOFFL [<n>] přičemž 0 ≤ n ≤ 3	Načtení aktuální hodnoty offsetu z funkce TOFFL (když je n = 0), příp. TOFFL[1...3] (když je n = 1, 2, 3) v kontextu předběžného zpracování.
$P_TOFF [<geometrická osa>]	Načtení aktuální hodnoty offsetu z funkce TOFF[<geometrická osa>] v kontextu předběžného zpracování.
$P_TOFFR	Načtení aktuální hodnoty offsetu z funkce TOFFR v kontextu předběžného zpracování.
$P_TOFFCR	Načtení aktuální hodnoty offsetu rádiusu v rohu v kontextu předběžného zpracování.
$AC_TOFFL[<n>] přičemž 0 ≤ n ≤ 3	Načtení aktuální hodnoty offsetu z funkce TOFFL (když je n = 0), příp. TOFFL[1...3] (když je n = 1, 2, 3) v kontextu hlavního zpracování (synchronní akce).
$AC_TOFF[<geometrická osa>]	Načtení aktuální hodnoty offsetu z funkce $AC_TOFF[<geometrická osa>] v kontextu hlavního zpracování (synchronní akce).
$AC_TOFFR	Načtení aktuální hodnoty offsetu z funkce TOFFR v kontextu hlavního zpracování (synchronní akce).
$AC_TOFFCR	Načtení aktuální hodnoty offsetu rádiusu v rohu v kontextu hlavního zpracování (synchronní akce).

Poznámka

Systémové proměnné $AC_TOFFL, $AC_TOFF, AC_TOFFR a AC_TOFFCR spouštějí při načítání z kontextu předběžného zpracování (NC program) automatické zastavení předběžného zpracování.

Použití

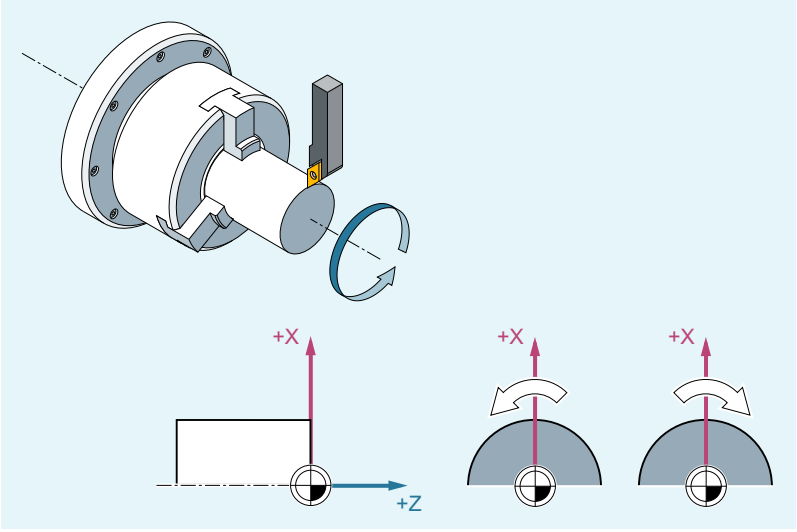
Funkce "Programovatelný offset korekce nástroje" je zajímavá speciálně pro kulové frézy a frézy s rohovým rádiusem, protože v systému CAM jsou výpočty často uskutečňovány na střed koule a nikoli na její špičku. Při měření nástroje se ale zpravidla měří jeho špička a zjištěný údaj se pak ukládá jako délka nástroje do paměti korekčních parametrů.

V případě 3D korekce rádiusu nástroje, jestliže se používá kulová fréza, je rozumné provádět korekci délky nástroje a rádiusu současně o stejnou hodnotu. Pro tento účel je k dispozici adresa TOFFLR.

3.6 pohyby vřetena

3.6.1 Otáčky vřetena (S), směr otáčení vřetena (M3, M4, M5)

Zadáním otáček a směru otáčení vřetena se vřeteno uvede do otáčivého pohybu, čímž je splněn předpoklad pro následné obrábění oddělováním třísky.



Vedle hlavního vřetena mohou být k dispozici ještě i další vřetena (např. v případě soustruhů protivřeteno nebo poháněný nástroj). Hlavní vřeteno je zpravidla pomocí strojního parametru deklarováno také jako řídící vřeteno. Toto přiřazení může být NC příkazem změněno.

Syntaxe

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

`M3 / M<n>=3`

`M4 / M<n>=4`

M5 / M<n>=5

SETMS(<n>)	
...	
SETMS	

Význam

S...:	Otáčky vřetena v otáčkách/min pro řídící vřeteno
S<n>=...:	Otáčky vřetena v otáčkách/min pro vřeteno <n>
	Poznámka Otáčky definované příkazem S0=... platí pro řídící vřeteno.
M3:	Směr otáčení vpravo pro řídící vřeteno
M<n>=3:	Směr otáčení vpravo pro vřeteno <n>
M4:	Směr otáčení vlevo pro řídící vřeteno
M<n>=4:	Směr otáčení vlevo pro vřeteno <n>
M5:	Zastavení vřetena pro řídící vřeteno
M<n>=5:	Zastavení vřetena pro vřeteno <n>
SETMS(<n>):	Vřeteno <n> má platit jako řídící vřeteno
SETMS:	Příkaz SETMS bez zadání vřetena způsobí přepnutí zpět na řídící vřeteno nastavené v konfiguraci

Poznámka

V jednom NC bloku smí být naprogramovány maximálně 3 S-hodnoty, např.:

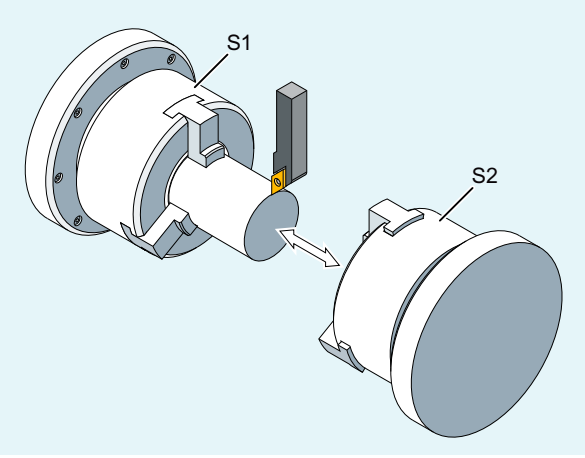
```
S... S2=... S3=...
```

Poznámka

Příkaz SETMS se musí nacházet v samostatném bloku.

Příklad

S1 je řídící vřeteno, S2 je druhé pracovní vřeteno. Soustružená část se má obrábět na dvou stranách. Kvůli tomu je nezbytné rozdělení pracovního postupu. Po upichování převezme synchronní zařízení (S2) obrobek za účelem opracování na straně upichování. Za tím účelem je toto vřeteno S2 definováno jako řídící vřeteno, pro něž pak platí příkaz G95.



Programový kód	Komentář
N10 S300 M3	; otáčky a směr otáčení pro hnací vřeteno = předem nastavené řídící vřeteno
...	; obrobení pravé strany obrobku
N100 SETMS(2)	; S2 je nyní řídícím vřetenem
N110 S400 G95 F…	; otáčky pro nové řídící vřeteno
...	; obrobení levé strany obrobku
N160 SETMS	; přepnutí zpět na řídící vřeteno S1

Další informace

Interpretace hodnoty S u hlavního vřetena

Jestliže je ve skupině G-funkcí 1 (příkazy pohybů s modální platností) aktivní funkce G331 nebo G332, je naprogramovaná hodnota S vždy interpretována jako otáčky v otáčkách za minutu. Jinak je interpretace hodnoty S závislá na skupině G-funkcí 15 (typ posuvu): Když je aktivní některý z příkazů G96, G961 nebo G962, je hodnota S interpretována jako konstantní řezná rychlost v m/min, ve všech ostatních případech jako otáčky v otáčkách/min.

Při přepnutí z funkce G96/G961/G962 na G331/G332 se hodnota konstantní řezné rychlosti nastavuje na nulu, při přepnutí z funkce G331/G332 na kteroukoli funkci ze skupiny G-funkcí 1 kromě G331/G332 se na nulu nastavuje hodnota otáček. Příslušnou hodnotu S je nutno v případě potřeby znovu naprogramovat.

Předem definované M-příkazy M3, M4, M5

V bloku s příkazy pro osy se příkazy M3, M4, M5 provádějí ještě **předtím**, než se spustí pohyby os (základní nastavení řídícího systému).

Příklad:

Programový kód	Komentář
N10 G1 F500 X70 Y20 S270 M3	; vřeteno se roztočí na 270 ot/min, potom se uskuteční pohyby v ose X a Y
N100 G0 Z150 M5	; zastavení vřetena před zpětným pohybem v ose Z

Poznámka

Prostřednictvím strojního parametru lze nastavit, jestli se pohyby os mají uskutečnit až po náběhu vřetena na požadované otáčky, příp. po jeho úplném zastavení, nebo zda mají být zahájeny bezprostředně po naprogramovaném přechodu na další blok.

Práce s větším počtem vřeten

V jednom kanálu může současně existovat 5 vřeten (řídící vřeteno plus 4 přídavná vřetena).

Jedno vřeteno je strojním parametrem definováno jako **řídící vřeteno**. Pro toto vřeteno platí speciální funkce, jako např. pro řezání a vrtání závitů, otáčkový posuv, doba prodlevy. Pro zbývající vřetena (např. pro druhé pracovní vřeteno a poháněný nástroj) se musí v příkazech pro otáčky, směr otáčení a zastavení vřetena uvádět odpovídající čísla.

Příklad:

Programový kód	Komentář
N10 S300 M3 S2=780 M2=4	; řídící vřeteno: 300 ot/min, vřeteno se otáčí vpravo 2. vřeteno: 780 ot/min, vřeteno se otáčí vlevo

Programovatelné přepnutí řídícího vřetena

Prostřednictvím příkazu `SETMS(<n>)` může být v NC programu definováno kterékoli z vřeten jako vřeteno řídící. Příkaz `SETMS` se musí nacházet v samostatném bloku.

Příklad:

Programový kód	Komentář
N10 SETMS(2)	; vřeteno 2 je nyní řídícím vřetenem

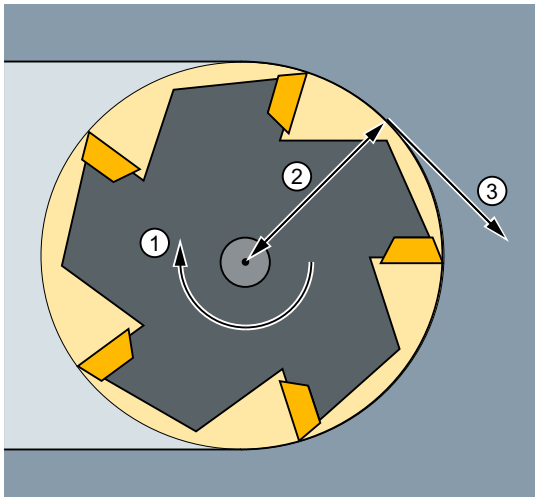
Poznámka

Pro nově deklarované řídící vřeteno nyní platí otáčky zadané příkazem `S...` a také funkce naprogramované příkazy `M3`, `M4` a `M5`.

Pomocí příkazu `SETMS` bez udání vřetena je možno přepnout zpět na řídící vřeteno definované ve strojním parametru.

3.6.2 Řezná rychlost nástroje (SVC)

Alternativou k otáčkám vřetena může být v případě frézovacích prací naprogramována také běžně využívaná řezná rychlost nástroje.



① Otáčky vřetena
② Rádius nástroje
③ Řezná rychlost nástroje

Na základě rádiusu aktivního nástroje vypočítá řídící systém pomocí následujícího vztahu z naprogramované řezné rychlosti nástroje potřebné otáčky vřetena:

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

kde:
- S: Otáčky vřetena v ot/min
- SVC: Řezná rychlost nástroje v m/min, příp. ve stopách/min
- R_{WKZ}: Rádius aktivního nástroje v mm

Na typ ($TC_DP1) aktivního nástroje se nebere ohled.

Naprogramovaná řezná rychlost nástroje je nezávislá na rychlosti posuvu po dráze F, stejně jako na skupině G-funkcí č. 15 (typ posuvu). Směr otáčení a spouštění vřetena se uskutečňuje pomocí příkazů M3, příp. M4, vřeteno se zastavuje příkazem M5.

Změna údajů o rádiusu nástroje v paměti korekčních parametrů vstupuje v platnost s následujícím vyvoláním korekčních parametrů nástroje, příp. s následující aktualizací aktivních korekčních parametrů.

Výměna nástroje a aktivování/deaktivování datového bloku korekčních parametrů nástroje má za následek nový výpočet právě používaných otáček vřetena.

Předpoklady

Programování řezné rychlosti nástroje vyžaduje znalost následujících informací:

- Geometrické charakteristiky otáčejícího se nástroje (frézovací nebo vrtací nástroj)
- Datový blok korekčních parametrů aktivního nástroje

Syntaxe

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

Význam

SVC:	Klíčové slovo pro programování řezné rychlosti nástroje	
[<n>]:	Číslo vřetena Pomocí tohoto rozšíření adresy se udává, pro které vřeteno má naprogramovaná řezná rychlost platit. Pokud toto rozšíření adresy není uvedeno, vztahuje se údaj vždy na aktuální řídící vřeteno. **Upozornění:** Pro každé vřeteno může být zadána jeho vlastní řezná rychlost. **Upozornění:** Jestliže je naprogramován příkaz SVC bez rozšíření adresy, předpokládá se, že se v řídícím vřetenu nachází aktivní nástroj. Při změně řídícího vřetena musí uživatel vybrat a aktivovat odpovídající nástroj.	
<hodnota>:	Hodnota řezné rychlosti nástroje	
	Jednotka:	m/min (v případě G71/G710), příp. stopy/min (v případě G70/G700)
T... D...:	V bloku s SVC musí být znám rádius nástroje. To znamená, že musí být aktivní nebo v daném bloku musí být vyvolán odpovídající nástroj včetně datového bloku svých korekčních parametrů. Pořadí příkazů SVC a T/D při programování v jednom bloku je libovolné.	
S... M3/M4:	Naprogramování otáček vřetena má za následek deaktivování řezné rychlosti nástroje. **Upozornění:** Přepínání mezi programováním pomocí příkazů SVC a S je možno provádět libovolně, a to i když se vřeteno otáčí. V daném okamžiku neaktivní hodnota se vymaže.	

Poznámka

Programování SVC není možné, jestliže jsou aktivní následující pracovní pohyby vřetena:

- Konstantní řezná rychlost: G96/G961/G962 S... (Strana 99)
- Konstantní obvodová rychlost brusného kotouče: SUG (Strana 104)
- Polohování vřetena: SPOS/SPOSA/M19 (Strana 118)
- Přepnutí hlavního vřetena do osového režimu: M70 (Strana 118)

A obráceně platí, že naprogramování kterékoli z těchto funkcí má za následek deaktivování SVC (řezná rychlost nástroje).

Poznámka

Maximální otáčky nástroje

Maximální otáčky nástroje (otáčky vřetena) je možné předem nastavit pomocí systémové proměnné $TC_TP_MAX_VELO[<T-číslo>].
Jestliže žádná mezní hodnota otáček není nastavena, žádné monitorování se neprovádí.

Poznámka

Dráhy "normovaných nástrojů" generované např. systémy CAD, ve kterých se již rádius nástroje započítává a které obsahují pouze rozdíl rádiusu břitu oproti normovanému nástroji, nejsou v souvislosti se SVC podporovány.

Příklady

Pro všechny příklady má platit: Držák nástroje = vřeteno (pro standardní frézování)

Příklad 1: Fréza s rádiusem 6 mm

Programový kód	Komentář
N10 G0 X10 T1 D1	; aktivování frézovacího nástroje s např. $TC_DP6[1,1] = 6 (rádius nástroje = 6 mm)
N20 SVC=100 M3	; řezná rychlost = 100 m/min ⇒ Z toho vyplývající otáčky vřetena: S = (100 m/min * 1000) / (6,0 mm * 2 * 3,14) = 2653,93 ot/min
N30 G1 X50 G95 FZ=0.03	; SVC a posuv na zub
...	

Příklad 2: Aktivování nástroje a SVC ve stejném bloku

Programový kód	Komentář
N10 G0 X20	
N20 T1 D1 SVC=100	; aktivování nástroje a datového bloku korekčních parametrů spolu s SVC v jednom bloku (na pořadí nezáleží)
N30 X30 M3	; spuštění vřetena, směr otáčení vpravo, řezná rychlost 100 m/min
N40 G1 X20 F0.3 G95	; SVC a otáčkový posuv

Příklad 3: Zadání řezných rychlostí pro dvě vřetena

Programový kód	Komentář
N10 SVC[3]=100 M6 T1 D1	
N20 SVC[5]=200	; rádius nástroje z aktivních korekčních parametrů nástroje je pro obě vřetena stejný, nastavené otáčky pro vřeteno 3 jsou jiné než pro vřeteno 5

Příklad 4:

Předpoklady:

Řídící vřeteno, příp. výměna nástroje jsou určeny pomocí proměnné Toolholder.

MD20124 $MC_TOOL_MANAGEMENT_TOOLHOLDER > 1

Při výměně nástroje zůstávají staré korekční parametry nástroje zachovány a teprve s naprogramováním příkazu `D` se korekční parametry nového nástroje aktivují:

MD20270 $MC_CUTTING_EDGE_DEFAULT = - 2

Programový kód	**Komentář**
N10 $TC_MPP1[9998,1]=2	; místo v zásobníku je držák nástroje
N11 $TC_MPP5[9998,1]=1	; místo v zásobníku je držák nástroje 1
N12 $TC_MPP_SP[9998,1]=3	; držák nástroje 1 je přiřazen vřetenu 3
N20 $TC_MPP1[9998,2]=2	; místo v zásobníku je držák nástroje
N21 $TC_MPP5[9998,2]=4	; místo v zásobníku je držák nástroje 4
N22 $TC_MPP_SP[9998,2]=6	; držák nástroje 4 je přiřazen vřetenu 6
N30 $TC_TP2[2]="WZ2"	
N31 $TC_DP6[2,1]=5.0	; rádius = 5,0 mm, nástroj T2, korekční parametry D1
N40 $TC_TP2[8]="WZ8"	
N41 $TC_DP6[8,1]=9.0	; rádius = 9,0 mm, nástroj T8, korekční parametry D1
N42 $TC_DP6[8,4]=7.0	; rádius = 7,0 mm, nástroj T8, korekční parametry D4
...	
N100 SETMTH(1)	; definice čísla hlavního držáku nástroje
N110 T="WZ2" M6 D1	; bude vyměněn a upnut nástroj T2 a aktivují se korekční parametry D1
N120 G1 G94 F1000 M3=3 SVC=100	; S3 = (100 m/min * 1000) / (5,0 mm * 2 * 3,14) = 3184,71 ot/min
N130 SETMTH(4)	; definice čísla hlavního držáku nástroje
N140 T="WZ8"	; odpovídá T8="WZ8"
N150 M6	; odpovídá M4=6
	Nástroj "WZ8" přichází na držáku Mastertoolholder, ale kvůli nastavení parametru MD20270=–2 zůstávají staré korekční parametry nástroje aktivní.
N160 SVC=50	; S3 = (50 m/min * 1000) / (5,0 mm * 2 * 3,14) = 1592,36 ot/min
	Korekční parametry držáku nástroje 1 jsou stále ještě aktivní a jsou přiřazeny vřetenu 3.
N170 D4	; korekční parametry D4 nového nástroje "WZ8" jsou aktivní (držák nástroje 4)
N180 SVC=300	; S6 = (300 m/min * 1000) / (7,0 mm * 2 * 3,14) = 6824,39 ot/min
	Vřetenu 6 je přiřazen držák nástroje 4.

Příklad 5:

Předpoklady:

Vřetena jsou současně držáky nástrojů:

MD20124 $MC_TOOL_MANAGEMENT_TOOLHOLDER = 0

Při výměně nástroje se automaticky aktivuje datový blok korekčních parametrů D4:

MD20270 $MC_CUTTING_EDGE_DEFAULT = 4

Programový kód	**Komentář**
N10 $TC_MPP1[9998,1]=2	; místo v zásobníku je držák nástroje
N11 $TC_MPP5[9998,1]=1	; místo v zásobníku je držák nástroje 1 = vřeteno 1
N20 $TC_MPP1[9998,2]=2	; místo v zásobníku je držák nástroje
N21 $TC_MPP5[9998,2]=3	; místo v zásobníku je držák nástroje 3 = vřeteno 3
N30 $TC_TP2[2]="WZ2"	
N31 $TC_DP6[2,1]=5.0	; rádius = 5,0 mm, nástroj T2, korekční parametry D1
N40 $TC_TP2[8]="WZ8"	
N41 $TC_DP6[8,1]=9.0	; rádius = 9,0 mm, nástroj T8, korekční parametry D1
N42 $TC_DP6[8,4]=7.0	; rádius = 7,0 mm, nástroj T8, korekční parametry D4
...	
N100 SETMS(1)	; vřeteno 1 = řídící vřeteno
N110 T="WZ2" M6 D1	; bude vyměněn a upnut nástroj T2 a aktivují se korekční parametry D1
N120 G1 G94 F1000 M3 SVC=100	; S1 = (100 m/min * 1000) / (5,0 mm * 2 * 3,14) = 3184,71 ot/min
N200 SETMS(3)	; vřeteno 3 = řídící vřeteno
N210 M4 SVC=150	; S3 = (150 m/min * 1000) / (5,0 mm * 2 * 3,14) = 4777,07 ot/min
	Vztahuje se na korekční parametry D1 nástroje T="WZ2", S1 se dál otáčí se starou hodnotou otáček.
N220 T="WZ8"	; odpovídá T8="WZ8"
N230 M4 SVC=200	; S3 = (200 m/min * 1000) / (5,0 mm * 2 * 3,14) = 6369,43 ot/min
	Vztahuje se na korekční parametry D1 nástroje T="WZ2".
N240 M6	; odpovídá M3=6
	Nástroje "WZ8" se dostává do řídícího vřetena, aktivují se korekční parametry D4 nového nástroje.
N250 SVC=50	; S3 = (50 m/min * 1000) / (7,0 mm * 2 * 3,14) = 1137,40 ot/min
	Korekční parametry D4 nástroje v řídícím vřetenu jsou aktivní.
N260 D1	; korekční parametry D1 nového nástroje "WZ8" jsou aktivní
N270 SVC[1]=300	; S1 = (300 m/min * 1000) / (9,0 mm * 2 * 3,14) = 5307,86 ot/min
	S3 = (50 m/min * 1000) / (9,0 mm * 2 * 3,14) = 884,64 ot/min
...	

Další informace

Rádius nástroje

K rádiusu nástroje se přispívají následující korekční parametry (aktivního nástroje):

- $TC_DP6 (rádius- geometrie)
- $TC_DP15 (rádius - opotřebení)
- $TC_SCPx6 (korekce k $TC_DP6)
- $TC_ECPx6 (korekce k $TC_DP6)

Nejsou brány v úvahu:

- On-line korekce rádiusu
- Přídavek rozměru pro naprogramovanou konturu (OFFN)

Korekce rádiusu nástroje (G41/G42)

Obě korekce rádiusu nástroje (G41/G42) a SVC se vztahují na rádius nástroje, ale pokud jde o jejich funkci, jsou uplatňovány odděleně a nezávisle na sobě.

Vrtání závitů bez vyrovnávací hlavičky (G331, G332)

Příkaz SVC je možné naprogramovat i ve spojení s příkazy G331, příp. G332.

Synchronní akce

Zadávání příkazu SVC ze synchronních akcí není možné.

Načtení řezné rychlosti a varianty programování otáček vřetena

Řeznou rychlost vřetena a variantu programování otáček (otáčky vřetena S nebo řezná rychlost nástroje SVC) je možné načíst pomocí systémových proměnných:

- Se zastavením předběžného zpracování ve výrobním programu pomocí systémových proměnných:

$AC_SVC[<n>]	Řezná rychlost, která byla v platnosti při přípravě aktuálního bloku v hlavní větvi programu pro vřeteno s číslem <n>.
$AC_S_TYPE[<n>]	Varianta programování otáček vřetena, která byla v platnosti při přípravě aktuálního bloku v hlavní větvi programu pro vřeteno s číslem <n>.

Hodnota: Význam:

1	Otáčky vřetena S v ot/min
2	Řezná rychlost nástroje SVC v m/min, příp. ve stopách/min

- Bez zastavení předběžného zpracování ve výrobním programu pomocí systémových proměnných:

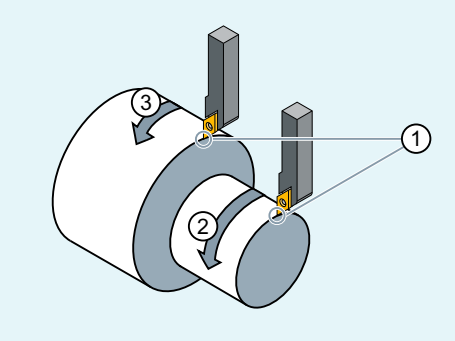
$P_SVC[<n>]	Naprogramovaná řezná rychlost pro vřeteno <n>
$P_S_TYPE[<n>]	Naprogramované otáčky pro danou variantu programování pro vřeteno <n>

Hodnota: Význam:

1	Otáčky vřetena S v ot/min
2	Řezná rychlost nástroje SVC v m/min, příp. ve stopách/min

3.6.3 Konstantní řezná rychlost (G96/G961/G962, G97/G971/G972, G973, LIMS, SCC)

Když je aktivována funkce "Konstantní řezná rychlost", v závislosti na průměru daného obrobku se automaticky mění otáčky vřetena tak, aby řezná rychlost S v m/min, příp. ve stopách/min na břitu nástroje zůstávala konstantní.



① Konstantní řezná rychlost
② Zvýšené otáčky vřetena
③ Snížené otáčky vřetena

Z toho vyplývají následující výhody:

- Stejnoměrný soustružený vzhled a v důsledku toho i lepší jakost povrchu
- Šetření nástroje při obrábění

Syntaxe

Zapnutí/vypnutí konstantní řezné rychlosti pro řídící vřeteno:

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

Omezení otáček pro řídící vřeteno:

```
LIMS=<hodnota>
LIMS[<vřeteno>]=<hodnota>
```

Jiná vztažná osa pro příkazy G96/G961/G962:
`SCC[<osa>]`

Poznámka

Příkaz SCC[<osa>] může být naprogramován odděleně nebo spolu s příkazy G96/G961/G962.

Význam

`G96`:	Otáčkový posuv (stejně jako u G95 (Strana 106)) a konstantní řezná rychlost Je-li v platnosti G96, automaticky se aktivuje G95. Pokud příkaz G95 ještě nebyl aktivován, je nutné při volání funkce G96 zadat pomocí příkazu F... novou hodnotu posuvu.	
`G961`:	Lineární posuv (stejně jako u G94 (Strana 106)) a konstantní řezná rychlost	
`G962`:	Lineární posuv nebo otáčkový posuv a konstantní řezná rychlost	
`S...`:	Spolu s příkazy G96, G961 příp. G962 není příkaz S... interpretován jako otáčky vřetena, nýbrž jako řezná rychlost. Řezná rychlost je vždy vztažena na řídící vřeteno.	
	Jednotka:	m/min (v případě G71/G710), příp. stopy/min (v případě G70/G700)
	Rozsah hodnot:	0,1 m/min ... 9999 9999,9 m/min
`G97`:	Otáčkový posuv a konstantní otáčky vřetena (konstantní řezná rychlost je vypnuta)	
`G971`:	Lineární posuv a konstantní otáčky vřetena (konstantní řezná rychlost je vypnuta)	
`G972`:	Lineární posuv nebo otáčkový posuv a konstantní otáčky vřetena (konstantní řezná rychlost je vypnuta)	
`G973`:	Otáčkový posuv bez omezení otáček vřetena a konstantní otáčky vřetena (G97 bez LIMS pro režim ISO)	
	Poznámka: Po příkazu G97 (příp. G971 ... G973) je příkaz S... znovu interpretován jako otáčky vřetena v otáčkách/min. Jestliže nejsou žádné nové otáčky vřetena udány, zůstanou zachovány otáčky, které byly naposled nastaveny příkazy G96 (příp. G961 nebo G962).	
`LIMS`:	Omezení otáček pro řídící vřeteno (je v platnosti, jen když je aktivní některý z příkazů G96/G961/G97). U strojů s možností přepínání řídícího vřetena mohou být v jednom bloku naprogramována omezení otáček pro až 4 vřetena s odlišnými hodnotami.	
	`<vřeteno>`:	Číslo vřetena
	`<hodnota>`:	Maximální mezní hodnota otáček vřetena v otáčkách/min
`SCC`:	Když je aktivní některá z funkcí G96/G961/G962, lze pomocí příkazu SCC[<osa>] přiřadit kteroukoli z geometrických os jako osu vztažnou.	

Poznámka

Při prvním vyvolání funkcí G96/G961/G962 musí být zadána také konstantní řezná rychlost S..., při opětovném vyvolávání funkcí G96/G961/G962 je její zadání volitelné.

Poznámka

Omezení otáček naprogramované pomocí příkazu LIMS nesmí být vyšší, než jsou mezní otáčky naprogramované příkazem G26 nebo definované nastavovaným parametrem.

Poznámka

Vztažná osa pro příkazy G96/G961/G962 musí být v okamžiku naprogramování příkazu SCC[<osa>] geometrickou osou, která je v kanálu známá. Naprogramování příkazu SCC[<osa>] je možné, i když je příkaz G96/G961/G962 aktivní.

Příklady

Příklad 1: Aktivování konstantní řezné rychlosti s omezením otáček

Programový kód	Komentář
N10 SETMS(3)	
N20 G96 S100 LIMS=2500	; Konstantní řezná rychlost = 100 m/min, max. otáčky = 2500 ot/min
...	
N60 G96 G90 X0 Z10 F8 S100 LIMS=444	; Max. otáčky = 444 ot/min

Příklad 2: Zadání omezení otáček pro 4 vřetena

Omezení otáček jsou definována pro vřeteno 1 (řídící vřeteno) a vřetena 2, 3 a 4.

Programový kód
N10 LIMS=300 LIMS[2]=450 LIMS[3]=800 LIMS[4]=1500
...

Příklad 3: Přiřazení osy Y při příčném obrábění pomocí osy X

Programový kód	Komentář
N10 G18 LIMS=3000 T1 D1	; omezení otáček na 3000 ot/min
N20 G0 X100 Z200	
N30 Z100	
N40 G96 S20 M3	; konstantní řezná rychlost = 20 m/min, je závislá na ose X
N50 G0 X80	
N60 G1 F1.2 X34	; příčné obrábění v ose X s rychlostí 1,2 mm/otáčku
N70 G0 G94 X100	
N80 Z80	
N100 T2 D1	
N110 G96 S40 SCC[Y]	; osa Y je přiřazena příkazu G96 a příkaz G96 je aktivován (to je možné v jednom bloku) Konstantní řezná rychlost = 40 m/min, je závislá na ose Y.
...	
N140 Y30	
N150 G01 F1.2 Y=27	; zápich v ose Y, posuv F = 1,2 mm/otáčku
N160 G97	; deaktivování konstantní řezné rychlosti
N170 G0 Y100	

Další informace

Výpočet otáček vřetena

Základem pro výpočet otáček vřetena z naprogramované řezné rychlosti je poloha ENS příčné osy (rádius).

Poznámka

Framy mezi WCS a ENS (např. programovatelné framy, jako jsou SCALE, TRANS nebo ROT) jsou při výpočtu otáček vřetena zohledňovány a mohou ovlivňovat změnu otáček (např. jestliže se v důsledku transformace SCALE změní platná hodnota průměru).

Omezení otáček pomocí příkazu LIMS

Jestliže má být opracováván obrobek s velkými rozdíly průměru, doporučuje se pomocí příkazu LIMS zadat omezení otáček vřetena (maximální otáčky vřetena), díky čemuž je možno vyloučit nepřípustně vysoké otáčky na malých průměrech. Omezení LIMS je v platnosti, jen když je aktivní některý z příkazů G96, G961 a G97. Když je v platnosti příkaz G971, funkce LIMS se neuplatňuje. Při přechodu na další blok v hlavní větvi programu se všechny naprogramované hodnoty přenášejí do nastavovaných parametrů.

Poznámka

Mezní hodnoty otáček změněné ve výrobním programu pomocí příkazu LIMS jsou předávány do nastavovaných parametrů a zůstávají v nich proto uloženy i po skončení programu.

Pokud mezní hodnoty otáček změněné pomocí příkazu LIMS nemají být oproti tomu po skončení programu zachovány, musí být v modulu GUD výrobce stroje vložena následující definice:

```
REDEF $SA_SPIND_MAX_VELO_LIMS PRLOC
```

Aktivování konstantní řezné rychlosti (G97/G971/G972/G973)

Po příkazu G97 (příp. G971 ... G973) je příkaz S... znovu interpretován jako otáčky vřetena v otáčkách/min. Jestliže nejsou žádné nové otáčky vřetena udány, zůstanou zachovány otáčky, které byly naposled nastaveny příkazy G96 (příp. G961 nebo G962).

Funkce G96/G961 může být deaktivována také příkazy G94 nebo G95. V tomto případě platí pro další obráběcí proces naposled naprogramované otáčky S....

Příkaz G97 může být naprogramován i bez předešlého G96. Funkce se potom chová stejně jako G95, navíc je však možné naprogramovat LIMS.

Pomocí příkazu G973 může být konstantní řezná rychlost vypnuta, aniž by se aktivovalo omezení otáček vřetena.

Poznámka

Příčná osa musí být definována pomocí strojního parametru.

Najíždění rychlým posuvem G0

Při najíždění rychlým posuvem G0 se žádné změny otáček neuskutečňují.

Výjimka:

Pokud se má rychlým posuvem najíždět na konturu a v dalším NC-bloku je dráhový příkaz G1/G2/G3/..., potom se už v bloku najíždění s G0 nastavují otáčky pro následující příkaz pohybu po dráze.

Jiná vztažná osa pro příkazy G96/G961/G962

Když je aktivní některá z funkcí G96/G961/G962, lze pomocí příkazu SCC[<osa>] přiřadit kteroukoli z geometrických os jako osu vztažnou. Pokud se změní vztažná osa a v důsledku toho i vztažná pozice špičky nástroje (TCP - Tool Center Point) pro konstantní řeznou rychlost, budou výsledné otáčky vřetena upraveny v souladu s nastavenou charakteristikou změny rychlosti (brždění - zrychlování).

Výměna přiřazené kanálové osy

Charakteristika vztažné osy pro příkazy G96/G961/G962 je vždy přiřazena geometrické ose. V případě výměny osy zahrnující přiřazenou kanálovou osu zůstává charakteristika vztažné osy pro příkazy G96/G961/G962 zachována ve starém kanálu.

Výměna geometrické osy neovlivňuje přiřazení této geometrické osy ke konstantní řezné rychlosti. Pokud je výměnou geometrické osy ovlivněna vztažná pozice TCP pro příkazy G96/G961/G962, vřeteno nastaveným způsobem změní své otáčky na nové.

Jestliže v důsledku výměny geometrické osy žádná nová kanálová osa přiřazena není (např. GEOAX(0,X)), potom zůstanou otáčky vřetena zachovány v souladu s příkazem G97.

Příklady pro výměnu geometrické osy s přiřazeními vztažné osy:

Programový kód	**Komentář**
N05 G95 F0.1	
N10 GEOAX(1,X1)	; kanálová osa X1 se stane první geometrickou osou
N20 SCC[X]	; první geometrická osa (X) se stane vztažnou osou
	; pro příkazy G96/G961/G962
N30 GEOAX(1,X2)	; kanálová osa X2 se stane první geometrickou osou
N40 G96 M3 S20	; kanálová osa X2 je vztažnou osou pro příkaz G96

Programový kód	**Komentář**
N05 G95 F0.1	
N10 GEOAX(1,X1)	; kanálová osa X1 se stane první geometrickou osou
N20 SCC[X1]	; X1 a implicitně se první geometrická osa (X)
	;stane vztažnou osou pro příkazy G96/G961/G962
N30 GEOAX(1,X2)	; kanálová osa X2 se stane první geometrickou osou
N40 G96 M3 S20	; vztažnou osou pro G96 je X2, resp. X, žádný alarm

Programový kód	**Komentář**
N05 G95 F0.1	
N10 GEOAX(1,X2)	; kanálová osa X2 se stane první geometrickou osou
N20 SCC[X1]	; X1 není geometrickou osou, alarm

Programový kód	**Komentář**
N05 G0 Z50	

```
Programový kód          Komentář
N10 X35 Y30
N15 SCC[X]              ; vztažná osa pro příkazy G96/G961/G962 je osa X
N20 G96 M3 S20          ; aktivování konstantní řezné rychlosti 10 mm/min
N25 G1 F1.5 X20         ; příčné obrábění v ose X s rychlostí 1,5 mm/otáčku
N30 G0 Z51
N35 SCC[Y]              ; vztažná osa pro příkaz G96 je osa Y,
                        ; snížení otáček vřetena (Y30)
N40 G1 F1.2 Y25         ; příčné obrábění v ose Y s rychlostí 1,2 mm/otáčku
```

3.6.4 Aktivování/deaktivování konstantní obvodové rychlosti brusného kotouče (GWPSON, GWPSOF)

Konstantní obvodová rychlost brusného kotouče (GWPS) pro brusné nástroje (typ nástroje: 400 až 499) je aktivována, příp. deaktivována pomocí předem definovaných procedur GWPSON(...) a GWPSOF(...).

Syntaxe

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

Význam

`GWPSON(...)`:	Aktivování konstantní obvodové rychlosti brusného kotouče
`GWPSOF(...)`:	Deaktivování konstantní obvodové rychlosti brusného kotouče
`<TNr>`:	T-číslo **Poznámka:** Je zapotřebí jen tehdy, pokud má být konstantní obvodová rychlost brusného kotouče aktivována, příp. deaktivována na neaktivní brusný kotouč a nikoli na aktivní nástroj, který je momentálně používán.
`S<n>=...`:	Obvodová rychlost brusného kotouče v m/s nebo ve stopách/s pro vřeteno <n>
`S0=...`, příp. `S...`:	Obvodová rychlost brusného kotouče pro řídící vřeteno

Zjištění stavu

Pomocí následujících systémových proměnných je možné ve výrobním programu zjistit, zda je konstantní obvodová rychlost brusného kotouče pro určité vřeteno aktivní:

$P_GWPS[<n>] ; kde <n> = číslo vřetena

Hodnota	Význam
0 (= FALSE)	GWPS je **vypnuta**.
1 (= TRUE)	GWPS je **zapnuta**.

3.6.5 Programovatelné omezení otáček vřetena (G25, G26)

Minimální a maximální otáčky vřetena definované ve strojních a v nastavovaných parametrech mohou být ve výrobním programu změněny.

Naprogramované mezní hodnoty otáček vřetena mohou být stanoveny pro všechna vřetena daného kanálu.

Syntaxe

```
G25 S... S1=... S2=...
G26 S... S1=... S2=...
```

Význam

`G25`:	**Spodní** mezní hodnota otáček vřetena
`G26`:	**Horní** mezní hodnota otáček vřetena
`S... S1=... S2=...` :	Minimální, příp. maximální otáčky vřetena nebo vřeten
	Upozornění: V jednom bloku smí být naprogramovány maximálně tři omezení otáček vřetena.
	Rozsah hodnot: 0.1 ... 9999 9999.9 ot/min

Poznámka

Omezení otáček vřetena naprogramované pomocí příkazů `G25` a `G26` přepisuje mezní hodnoty otáček definované v nastavovaných parametrech a proto zůstává uloženo i po skončení programu.

Pokud mezní hodnoty otáček změněné pomocí příkazu `G25`/`G26` nemají být oproti tomu po skončení programu zachovány, musí být v modulu GUD výrobce stroje vložena následující definice:

REDEF $SA_SPIND_MIN_VELO_G25 PRLOC

REDEF $SA_SPIND_MAX_VELO_G26 PRLOC

Příklad:

Programový kód	Komentář
`N10 G26 S1400 S2=350 S3=600`	`; horní mezní otáčky pro řídící vřeteno, vřeteno 2 a vřeteno 3`

3.7 Regulace posuvu

3.7.1 Posuv (G93, G94, G95, F, FGROUP, FL, FGREF)

Prostřednictvím těchto příkazů definujete rychlosti posuvu v NC programu pro všechny osy podílející se na posloupnosti obrábění.

Syntaxe

```
G93
G94
G95
F<hodnota>
FGROUP(<osa_1>,<osa_2>,...)
FGREF[<kruhová osa>]=<vztažný rádius>
FL[<osa>]=<hodnota>
```

Význam

G93:	Typ pohybu po dráze: Časově reciproční posuv [1/min]
G94:	Typ pohybu po dráze: Lineární posuv [mm/min], [palce/min] nebo [stupně/min]
G95:	Typ pohybu po dráze: Otáčkový posuv [mm/otáčku], příp. [palce/otáčku] Můžete si zvolit, zda má být otáčkový posuv vztažen k pohybům řídícího vřetena nebo jiného libovolného vřetena nebo kruhové osy.
F<hodnota>	Posuv po dráze pro všechny dráhové osy nebo jen osy vybrané pomocí příkazu FGROUP.
FGROUP:	Stanovení dráhových os, na které se vztahuje hodnota posuvu po dráze naprogramovaná pomocí příkazu F.
FGREF:	Pomocí příkazu FGREF se pro všechny kruhové osy uvedené v příkazu FGROUP naprogramuje efektivní rádius (<vztažný rádius>).
FL:	Mezní hodnota rychlosti pro synchronní/dráhové osy Platí jednotka nastavená příkazem G94. Na jednu osu (kanálová osa, geometrická osa nebo orientační osa) smí být naprogramována jen jedna hodnota FL.
<osa>:	Název kanálové osy, typ: AXIS

Příklady

Příklad 1: Způsob fungování příkazu FGROUP

Následující příklad má osvětlit způsob fungování příkazu FGROUP na dráhu a posuv po dráze. Proměnná $AC_TIME obsahuje čas od začátku bloku v sekundách. Může se používat jenom při synchronních akcích.

```
Programový kód                          Komentář
N100 G0 X0 A0
N110 FGROUP(X,A)
N120 G91 G1 G710 F100                   ; posuv = 100 mm/min, příp. 100 stupňů/min
```

Programový kód	Komentář
N130 DO $R1=$AC_TIME	
N140 X10	; posuv = 100 mm/min, úsek dráhy = 10 mm, R1 = asi 6 s
N150 DO $R2=$AC_TIME	
N160 X10 A10	; posuv = 100 mm/min, úsek dráhy = 14.14 mm, R2 = asi 8 s
N170 DO $R3=$AC_TIME	
N180 A10	; posuv = 100 stupňů/min, úsek dráhy = 10 stupňů, R3 = asi 6 s
N190 DO $R4=$AC_TIME	
N200 X0.001 A10	; posuv = 100 mm/min, úsek dráhy = 10 mm, R4 = asi 6 s
N210 G700 F100	; posuv = 2540 mm/min, příp. 100 stupňů/min
N220 DO $R5=$AC_TIME	
N230 X10	; posuv = 2540 mm/min, úsek dráhy = 254 mm, R5 = asi 6 s
N240 DO $R6=$AC_TIME	
N250 X10 A10	; posuv = 2540 mm/min, úsek dráhy = 254,2 mm, R6 = asi 6 s
N260 DO $R7=$AC_TIME	
N270 A10	; posuv = 100 stupňů/min, úsek dráhy = 10 stupňů, R7 = asi 6 s
N280 DO $R8=$AC_TIME	
N290 X0.001 A10	; posuv = 2540 mm/min, úsek dráhy = 10 mm, R8 = asi 0,288 s
N300 FGREF[A]=360/(2*$PI)	; nastavení 1 stupeň=1 palec pomocí efektivního rádiusu
N310 DO $R9=$AC_TIME	
N320 X0.001 A10	; posuv = 2540 mm/min, úsek dráhy = 254 mm, R9 = asi 6 s
N330 M30	

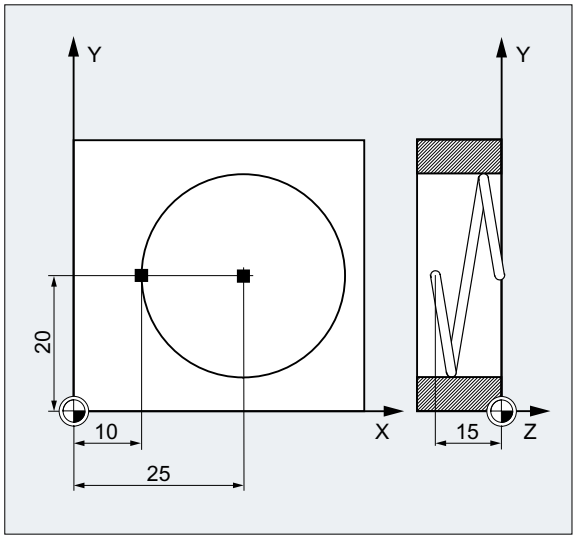
Příklad 2: Pohyb synchronních os s mezní hodnotou rychlosti FL

Rychlost pohybu po dráze dráhových os se sníží, jestliže synchronizovaná osa Z dosáhne své mezní rychlosti.

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

Příklad 3: Spirální interpolace

Dráhové osy X a Y se pohybují s naprogramovaným posuvem, osa Z je synchronní osou.

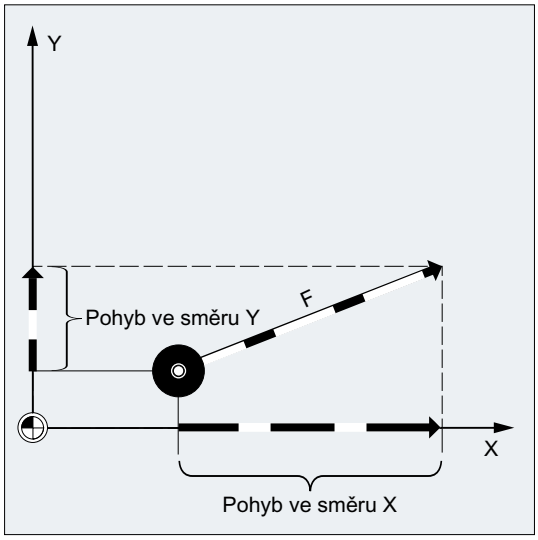


Programový kód	**Komentář**
`N10 G17 G94 G1 Z0 F500`	`; přísuv nástroje`
`N20 X10 Y20`	`; najíždění na počáteční pozici`
`N25 FGROUP(X,Y)`	`; osy X/Y jsou dráhové osy, Z je synchronní osa`
`N30 G2 X10 Y20 Z-15 I15 J0 F1000 FL[Z]=200`	`; na kruhové dráze platí posuv 1000 mm/min, ve směru Z je posuv synchronizovaný`
`...`	
`N100 FL[Z]=$MA_AX_VELO_LIMIT[0,Z]`	`; čtením hodnoty rychlosti z MD je mezní rychlost deaktivována, načtení hodnoty z MD`
`N110 M30`	`; konec programu`

Další informace

Rychlost posuvu pro dráhové osy (F)

V obvyklém případě se posuv po dráze skládá z jednotlivých složek rychlosti všech geometrických os podílejících se na pohybu a je vztažen na střed frézy, příp. na špičku soustružnického nože.



Rychlost posuvu se zadává pomocí adresy `F`. V závislosti na předdefinovaném nastavení strojních parametrů platí pomocí G-funkcí zadané rozměrové jednotky, a to buď mm nebo palce.

V jednom NC bloku smí být naprogramována jen jedna hodnota `F`. Jednotky rychlosti posuvu jsou definovány pomocí G-funkcí `G93`/`G94`/`G95`. Posuv `F` ovlivňuje pouze dráhové osy a platí tak dlouho, dokud není naprogramována nová hodnota posuvu. Po adrese `F` je přípustné použití oddělovacích znaků.

Příklady:

`F100` nebo `F 100`

`F.5`

```
F=2*FEED
```

Druh posuvu (G93/G94/G95)

Příkazy G-funkcí `G93`, `G94` a `G95` mají modální platnost. Pokud je příkaz `G93`, `G94` nebo `G95` změněn, je zapotřebí hodnotu posuvu po dráze znovu naprogramovat. Při obrábění pomocí kruhových os je možné posuv udávat také ve stupních/minutu.

Časově reciproční posuv (G93)

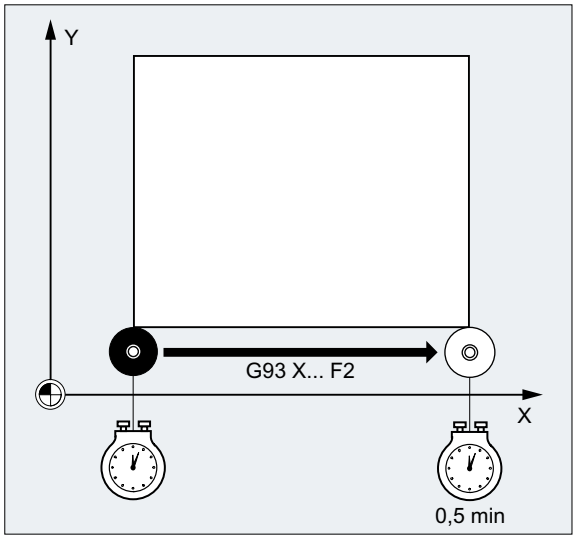
Časově inverzní posuv udává čas požadovaný na zpracování pohybového příkazu v bloku.

Jednotka: 1/min

Příklad:

```
N10 G93 G01 X100 F2
```

Znamená: Naprogramovaná dráha bude ujeta za 0,5 minuty.



Poznámka

Jestliže jsou délky drah blok od bloku velmi odlišné, v případě použití příkazu `G93` by měla být pro každý blok stanovena nová hodnota `F`-slova. Při obrábění pomocí kruhových os je možné posuv udávat také ve stupních/minutu.

Posuv pro synchronní osy

Posuv naprogramovaný pomocí adresy `F` platí pro všechny dráhové osy naprogramované v daném bloku, ne však pro synchronizované osy. Synchronizované osy jsou řízeny tak, aby pro svou dráhu potřebovaly stejný čas jako dráhové osy a všechny osy dosáhly svého koncového bodu ve stejný okamžik.

Mezní hodnota rychlosti pro synchronní osy (FL)

Pomocí příkazu `FL` je možné pro synchronní osy naprogramovat mezní hodnotu rychlosti. Jestliže příkaz `FL` není naprogramován, platí rychlost rychlého posuvu. Hodnota `FL` je deaktivována přiřazením do strojního parametru (MD36200 $MA_AX_VELO_LIMIT).

Ovládání dráhové osy v režimu synchronní osy (FGROUP)

Pomocí příkazu `FGROUP` lze definovat, zda se má dráhová osa pohybovat s rychlostí pohybu po dráze nebo jako synchronizovaná osa. Při spirální interpolaci (šroubovice) může být např. definováno, že se jen dvě geometrické osy X a Y mají pohybovat s naprogramovaným posuvem. Přísuvná osa Z by potom byla synchronní osou.

Příklad: `FGROUP(X,Y)`

Změna příkazu FGROUP

Nastavení vytvořené příkazem FGROUP je možné změnit:

1. novým naprogramováním příkazu FGROUP: např. FGROUP(X,Y,Z)
2. naprogramováním příkazu FGROUP bez udání osy: FGROUP()
   Po zpracování příkazu FGROUP() platí základní stav nastavený ve strojním parametru. Geometrické osy se nyní znovu pohybují ve skupině dráhových os.

Poznámka

Identifikátor osy v příkazu FGROUP musí být název kanálové osy.

Měřicí jednotky pro posuv F

Pomocí příkazů G-funkcí G700 a G710 se určuje, že systém měřicích jednotek bude platit nejen pro geometrické údaje, ale i pro posuvy F, tzn.:

- v případě G700: [palce/min]
- v případě G710: [mm/min]

Poznámka

Příkazy G70/G71 nejsou hodnoty posuvu **nijak** ovlivňovány.

Měřicí jednotky pro synchronní osy s mezní hodnotou rychlosti FL

Měřicí jednotky nastavené pro hodnotu F pomocí příkazů G-funkcí G700/G710 platí také pro příkaz FL.

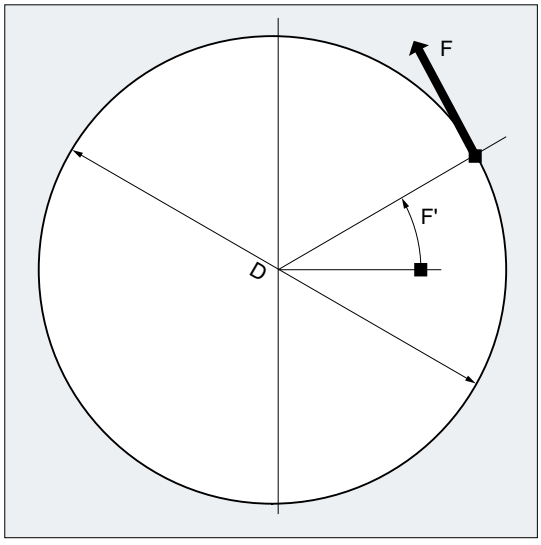
Měřicí jednotky pro kruhové a lineární osy

Pro lineární a kruhové osy, které jsou spolu spojeny příkazem FGROUP a mají společně urazit nějakou dráhu, platí posuv a měřicí jednotky lineárních os. V závislosti na předešlém nastavení G94/G95 v mm/min nebo v palcích/min, příp. v mm/otáčku nebo v palcích/otáčku.

Obvodová rychlost kruhové osy v mm/min nebo v palcích/min se vypočítá podle následujícího vzorce:

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

kde: F: obvodová rychlost
F': úhlová rychlost
π: konstanta kruhu
D: Průměr



Pohyb kruhových os s rychlostí pohybu po dráze F (FGREF)

Pro obráběcí operace, u kterých se nástroj nebo obrobek nebo oba mají pohybovat pomocí kruhové osy, může být platný pracovní posuv interpretován obvyklým způsobem jako rychlost pohybu po dráze pomocí F-slova. Za tím účelem musí být pro každou z podílejících se kruhových os udán efektivní rádius (vztažný rádius).

Jednotky, v nichž je vztažný rádius udán, závisí na nastavení pomocí příkazů `G70`/`G71`/`G700`/`G710`.

Všechny osy podílející se na pohybu musí být zahrnuty v příkazu `FGROUP`, jinak nebudou při výpočtu posuvu po dráze vyhodnocovány.

Aby zůstala zachována kompatibilita s chováním bez naprogramování příkazu `FGREF`, po zapnutí nebo po resetu systému je aktivováno nastavení 1 stupeň = 1 mm. To odpovídá referenčnímu rádiusu FGREF = 360 mm / (2π) = 57,296 mm.

Poznámka

Toto předdefinované nastavení je nezávislé jak na aktivním základním systému (MD10240 $MN_SCALING_SYSTEM_IS_METRIC), tak i na momentálně platném nastavení funkcí `G70`/`G71`/`G700`/`G710`.

Zvláštnosti:

Programový kód

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

V případě tohoto programu bude naprogramovaná hodnota F v bloku `N110` vyhodnocena jako posuv kruhové osy ve stupních/min, zatímco vyhodnocování posuvu v bloku `N120` bude záviset

na právě platném nastavení měřicích jednotek pomocí funkce G70/G71/G700/G710 buď jako 100 palců/min nebo jako 100 mm/min.

UPOZORNĚNÍ
Rozdílná hodnota posuvu Vyhodnocování FGREF se provádí i tehdy, když jsou v bloku naprogramovány jen kruhové osy. Obvyklá interpretace hodnoty F jako stupně/min platí v tomto případě jen tehdy, pokud referenční rádius odpovídá předdefinovanému nastavení funkce FGREF: • v případě G71/G710: FGREF[A]=57.296 • v případě G70/G700: FGREF[A]=57.296/25.4

Načtení vztažného rádiusu

Hodnotu vztažného rádiusu kruhové osy je možné načíst pomocí systémové proměnné:

- V synchronních akcích nebo se zastavením předběžného zpracování ve výrobním programu pomocí systémové proměnné:

  $AA_FGREF[<osa>] Aktuální hodnota v hlavní větvi programu

- Bez zastavení předběžného zpracování ve výrobním programu pomocí systémové proměnné:

  $PA_FGREF[<osa>] Naprogramovaná hodnota

Jestliže nejsou naprogramovány žádné hodnoty, v obou proměnných pro kruhové osy se načte předdefinované nastavení 360 mm / (2π) = 57,296 mm (což odpovídá 1 mm na stupeň).

Pro lineární osy se v obou proměnných vždy načte hodnota 1 mm.

Načítání dráhových os, které určují rychlost

Osy, které se podílejí na dráhové interpolaci, mohou být načítány pomocí systémových proměnných:

- V synchronních akcích nebo se zastavením předběžného zpracování ve výrobním programu pomocí systémových proměnných:

  $AA_FGROUP[<osa>] Jestliže má uvedená osa v základním nastavení nebo v důsledku naprogramování příkazu FGROUP vliv na rychlost pohybu po dráze v aktuálním bloku hlavní větve programu, je zjištěna hodnota "1". Pokud nemá, poskytuje proměnná hodnotu "0".

  $AC_FGROUP_MASK Poskytuje bitový klíč pro kanálové osy naprogramované pomocí příkazu FGROUP, které mají přispívat k rychlosti pohybu po dráze.

- Bez zastavení předběžného zpracování ve výrobním programu pomocí systémových proměnných:

$PA_FGROUP[<osa>]	Jestliže má uvedená osa v základním nastavení nebo v důsledku naprogramování příkazu `FGROUP` vliv na rychlost pohybu po dráze, je zjištěna hodnota "1". Pokud nemá, poskytuje proměnná hodnotu "0".
$P_FGROUP_MASK	Poskytuje bitový klíč pro kanálové osy naprogramované pomocí příkazu `FGROUP`, které mají přispívat k rychlosti pohybu po dráze.

Dráhové referenční faktory pro orientační osy s příkazem FGREF

U orientačních os je chování faktorů příkazu `FGREF[]` závislé na tom, zda se změna orientace nástroje uskutečňuje interpolací kruhové osy nebo vektorovou interpolací.

V případě **interpolace kruhové osy** se příslušné faktory `FGREF` orientačních os vypočítávají jednotlivě na základě vztažného rádiusu pro dráhu osy stejně jako u kruhových os.

V případě **vektorové interpolace** se použije efektivní faktor `FGREF`, který se vypočítá jako geometrický průměr jednotlivých faktorů `FGREF`.

FGREF[efektivní] = n-tá odmocnina z [(FGREF[A] * FGREF[B]...)]

kde: A: identifikátor 1. orientační osy
B: identifikátor 2. orientační osy
C: identifikátor 3. orientační osy
n: Počet orientačních os

Příklad:

Při standardní 5-osé transformaci existují dvě orientační osy a vypočítá se tedy efektivní faktor, který je odmocninou ze součinu faktorů obou os:

FGREF[efektivní] = druhá odmocnina z [(FGREF[A] * FGREF[B])]

Poznámka

Prostřednictvím efektivního faktoru orientačních os `FGREF` je možné na nástroji definovat vztažný bod, ke kterému se bude vztahovat naprogramovaný posuv po dráze.

3.7.2 Najíždění polohovacími osami (POS, POSA, POSP, FA, WAITP, WAITMC)

Polohovací osy se pohybují svým vlastním posuvem specifickým pro jednotlivé osy, nezávisle na dráhových osách. Neplatí žádné interpolační příkazy. Prostřednictvím příkazů POS/POSA/POSP se ovládá pohyb polohovacích os a současně se koordinují pohybové operace.

Typickými příklady polohovacích os jsou:

- Zařízení pro podávání palet
- Měřicí stanice

Pomocí příkazu WAITP může být v NC programu označeno místo, na němž se bude čekat tak dlouho, dokud osa naprogramovaná v předcházejícím bloku pomocí příkazu POSA neosáhne svého koncového bodu.

U příkazu WAITMC dojde k přechodu na další blok ihned, jakmile je přijata uvedená značka čekání.

Syntaxe

```
POS[<osa>]=<poloha>
POSA[<osa>]=<poloha>
POSP[<osa>]=(<koncová poloha>,<délka úseku>,<režim>)
FA[<osa>]=<hodnota>
WAITP(<osa>); programování vyžaduje samostatný NC blok!
WAITMC(<značka čekání>)
```

Význam

POS / POSA:	Najetí polohovací osou na uvedenou pozici Příkazy POS a POSA mají stejnou funkci, liší se ale chováním při přechodu na další blok: • V případě příkazu POS se na další NC blok se přejde až tehdy, když je dosaženo uvedené pozice. • V případě příkazu POSA se na další NC blok se přejde, i když uvedené pozice není ještě dosaženo.		
	<osa>:	Název osy, která se má pohybovat (identifikátor kanálové nebo geometrické osy)	
	<pozice>:	Posice osy, na krou se má najet.	
		Typ:	REAL
POSP:	Najetí polohovací osou na uvedenou koncovou pozici po zadaných menších úsecích		
	<koncová pozice>:	Koncová posice osy, na krou se má najet.	
	<dílčí úsek>:	Délka dílčího úseku dráhy	
	<režim>:	Způsob najíždění	
		= 0:	Posledním dvěma dílčím úsekům je přiřazena taková délka, aby zbývající dráha do koncové pozice byla rozdělena na dva stejně velké úseky (předdefinované nastavení).
		= 1:	Velikost dílčího úseku je přizpůsobena tak, aby součet všech vypočítaných dílčích úseků přesně odpovídal dráze do koncové pozice.
	Poznámka: Příkaz POSP se používá speciálně pro programování kyvných (Strana 964) (oscilačních) pohybů.		

FA:	Posuv pro uvedenou polohovací osu		
	<osa>:	Název osy, která se má pohybovat (identifikátor kanálové nebo geometrické osy)	
	<hodnota>:	Rychlost posuvu	
		Jednotka:	mm/min, příp. palce/min nebo stupně/min
	Upozornění: V jednom NC bloku může být naprogramováno maximálně 5 hodnot FA.		
WAITP:	Čekání na konec posuvu polohovací osy Se zpracováváním následujících bloků se bude čekat tak dlouho, dokud uvedená polohovací osa naprogramovaná v předcházejícím NC bloku s příkazem POSA nedosáhne své koncové pozice (s jemným přesným najetím).		
	<osa>:	Název osy (identifikátor kanálové nebo geometrické osy), pro kterou má platit příkaz WAITP.	
	Upozornění Pomocí příkazu WAITP může být osa uvolněna pro pohyb jako oscilační osa nebo jako konkurenční polohovací osa (prostřednictvím PLC).		
WAITMC:	Čekání, až se vyskytne uvedená čekací značka.. Jakmile je čekací značka zachycena, okamžitě se přejde na následující NC blok.		
	<čekací značka>:	Číslo značky, na kterou se čeká.	

⚠ POZOR

Posuv pomocí příkazu POSA

Jestliže je v následujícím bloku načten příkaz, který zastavení generuje implicitně, bude následující blok uskutečněn až tehdy, když jsou všechny předtím připravené a uložené bloky zpracovány. Předcházející blok bude zastaven v přesném najetí (jako při G9).

Příklady

Příklad 1: Posuv pomocí příkazu POSA a přístup ke stavovým údajům stroje

Při přístupu ke stavovým údajům stroje ($A...) generuje řídící systém interní zastavení předběžného zpracování. Zpracovávání bude pozastaveno, dokud nebudou zcela zpracovány všechny bloky, které byly připraveny a uloženy do paměti.

```
Programový kód                                  Komentář
N40 POSA[X]=100
N50 IF $AA_IM[X]==R100 GOTOF MARKE1             ; přístup ke stavovým údajům stroje
N60 G0 Y100
N70 WAITP(X)
N80 MARKE1:
N...
```

Příklad 2: Čekání na konec posuvu pomocí příkazu WAITP

Zařízení pro podávání palet

Osa U: Paletový zásobník
Doprava palety s obrobky do pracovního prostoru.

Osa V: Dopravníkový systém k měřicí stanici, ve které se uskutečňují kontroly náhodným výběrem doprovázející proces.

Programový kód	Komentář
`N10 FA[U]=100 FA[V]=100`	`; osové údaje posuvu pro jednotlivé polohovací osy U a V`
`N20 POSA[V]=90 POSA[U]=100 G0 X50 Y70`	`; pohyb polohovacích a dráhových os`
`N50 WAITP(U)`	`; zpracování programu bude pokračovat až tehdy, když osa U dosáhne pozice naprogramované v bloku N20`
…	

Další informace

Posuv pomocí příkazu POSA

Přechod na další blok, příp. zpracování programu, nejsou příkazem `POSA` nijak ovlivněny. Najíždění do koncového bodu může probíhat souběžně se zpracováváním následujících NC-bloků.

Posuv pomocí příkazu POS

Přechod na následující blok se uskuteční teprve tehdy, když všechny osy naprogramované v příkazu `POS` dosáhly své koncové pozice.

Čekání na konec posuvu pomocí příkazu WAITP

Po příkazu `WAITP` platí osa za neobsazenou NC programem tak dlouho, dokud není znovu naprogramována. Tyto osy pak mohou být ovládány prostřednictvím PLC jako polohovací osy nebo prostřednictvím NC programu/PLC nebo HMI jako oscilační osy.

Přechod na další blok během brždění pomocí příkazů IPOBRKA a WAITMC

Brždění osy bude probíhat jen tehdy, pokud nebyla dosud dosažena čekací značka nebo pokud jiné kritérium konce bloku zabraňuje přechodu na další blok. Po příkazu `WAITMC` se osy spouští okamžitě, jestliže žádné kritérium konce bloku zabraňuje přechodu na další blok.

3.7.3 Vřeteno v režimu regulace polohy (SPCON, SPCOF)

Pomocí příkazů SPCON, příp. SPCOF, se explicitně aktivuje nebo deaktivuje režim regulace polohy vřetena.

Poznámka

Aktivování režimu regulace polohy pomocí příkazu SPCON potřebuje maximálně tři takty regulátoru polohy.

Syntaxe

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

Význam

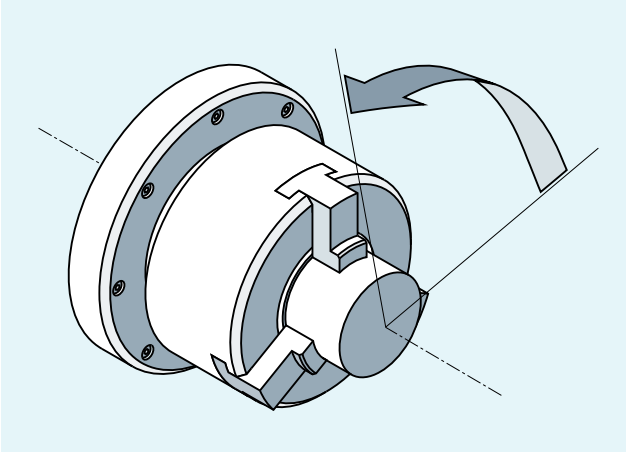
SPCON:	Aktivování režimu regulace polohy Uvedené vřeteno se přepne z režimu regulace otáček do režimu regulace polohy. Příkaz SPCON má modální působnost a zůstává v platnosti, dokud není přijat příkaz SPCOF.
SPCOF:	Deaktivování režimu regulace polohy Uvedené vřeteno se přepne z režimu regulace polohy do režimu regulace otáček.
<n>,<m>,... :	Čísla vřeten Bez udání čísla vřetena: Hlavní vřeteno v kanálu

Poznámka

V případě synchronního vřetena s vazbou pomocí požadované hodnoty nesmí být řídící vřeteno pomocí příkazu SPCOF přepnuto do režimu regulace otáček.

3.7.4 Nastavování polohy vřetena (SPOS, SPOSA, M19, M70, WAITS)

Pomocí příkazů SPOS, SPOSA nebo M19 je možné nastavovat vřetena do určité úhlové polohy, např. při výměně nástroje.



Příkazy SPOS, SPOSA a M19 způsobují dočasné přepnutí do režimu polohové regulace, a to až do následujícího zpracování některého z příkazů M3/M4/M5/M41 ... M45.

Nastavování polohy v osovém režimu

Vřeteno může být ovládáno také pomocí své adresy definované strojním parametrem jako dráhová, synchronizovaná nebo polohovací osa. Udáním identifikátoru osy se bude vřeteno nacházet v osovém režimu. Příkazem M70 se vřeteno přepne přímo do osového režimu.

Konec polohování

Kritérium konce pohybu může být při polohování vřetena naprogramováno prostřednictvím příkazů FINEA, CORSEA, IPOENDA nebo IPOBRKA.

Jestliže jsou splněna kritéria konce pohybu pro všechna vřetena a osy uvedené v daném bloku, kromě kritéria přechodu na další blok pro dráhovou interpolaci, přechod na další blok se provede.

Synchronizace

Aby bylo možno pohyby vřetena synchronizovat, lze pomocí příkazu WAITS nastavit čekání až do dosažení polohy vřetena.

Předpoklady

Vřeteno, jehož poloha má být nastavována, musí být schopno pracovat v režimu polohové regulace.

Syntaxe

Polohování vřetena:

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

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

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

Přepnutí vřetena do osového režimu:

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

Definice kritéria konce pohybu:

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

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

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

`IPOBRKA` / `IPOBRKA(<osa>[,<časový okamžik>])`; programování vyžaduje samostatný NC blok!

Synchronizace pohybů vřetena:

`WAITS` / `WAITS(<n>,<m>)` ; programování vyžaduje samostatný NC blok!

Význam

SPOS / SPOSA:	Nastavení vřetena do zadané úhlové polohy. Příkazy SPOS a SPOSA mají stejnou funkci, liší se ale chováním při přechodu na další blok: • V případě příkazu SPOS se na další NC blok se přejde až tehdy, když je pozice dosaženo. • V případě příkazu SPOSA se na další NC blok se přejde, i když pozice ještě dosaženo není.			
	<n>:	Číslo vřetena, jehož poloha má být nastavena. Pokud číslo vřetena není uvedeno nebo pokud je jako číslo vřetena zadána "0", vztahují se příkazy SPOS, příp. SPOSA na řídící vřeteno.		
	<hodnota>:	Úhlová poloha, do níž má být vřeteno nastaveno.		
		Jednotka:	stupně	
		Typ:	REAL	
		Pro programování režimu najíždění na zadanou polohu existují následující možnosti:		
		=AC(<hodnota>):	Absolutní údaj rozměru	
			Rozsah hodnot:	0 ... 359,9999
		=IC(<hodnota>):	Inkrementální údaj rozměru	
			Rozsah hodnot:	0 ... ±99 999,999
		=DC(<hodnota>):	Najíždění po přímé dráze na absolutní hodnotu	
		=ACN(<hodnota>):	Udávání absolutní hodnoty, najíždění v záporném směru	
		=ACP(<hodnota>):	Udávání absolutní hodnoty, najíždění v kladném směru	
		=<hodnota>:	stejné jako DC(<hodnota>)	
M<n>=19:	Řídící vřeteno (M19 nebo M0=19) nebo vřeteno s číslem <n> (M<n>=19) nastavit do úhlové polohy předem definované v parametru SD43240 $SA_M19_SPOS pomocí režimu polohování předdefinovaného pomocí parametru SD43250 $SA_M19_SPOSMODE. Na další NC-blok se přejde až tehdy, když je pozice dosaženo.			
M<n>=70:	Řídící vřeteno (M70 oder M0=70) nebo vřeteno s číslem <n> (M<n>=70) se přepne do režimu osy. Nebude najíždět na žádnou definovanou pozici. Na další NC-blok se přejde, až když je přepnutí dokončeno.			
FINEA:	Konec pohybu při dosažení jemného okna přesného najetí			
COARSEA:	Konec pohybu při dosažení hrubého okna přesného najetí			
IPOENDA:	Konec pohybu při dosažení "zastavení interpolátoru"			
S<n>:	Vřeteno, pro které má naprogramované kritérium konce pohybu platit.			
	<n>:	Číslo vřetena		
	Pokud není vřeteno [S<n>] uvedeno nebo pokud je jako číslo vřetena zadána "0", vztahuje se naprogramované kritérium konce pohybu na řídící vřeteno.			

IPOBRKA:	Přechod na další blok vztažený na hranu brzdné charakteristiky je možný.		
	<osa>:	Identifikátor kanálové osy	
	<časový okamžik>:	Časový okamžik přechodu na další blok vztáhnout ke hraně brzdné charakteristiky	
		Jednotka:	Procenta
		Rozsah hodnot:	100 (počáteční bod hrany brzdné charakteristiky) ... 0 (konec hrany brzdné charakteristiky)
		Pokud parametr <časový okamžik> není zadán, použije se aktuální hodnota z nastavovaného parametru: SD43600 $SA_IPOBRAKE_BLOCK_EXCHANGE **Poznámka:** Příkaz IBOBRKA s nastavením časového okamžiku na "0" je identický s příkazem IPOENDA.	
WAITS:	Synchronizační příkaz pro uvedená vřetena Se zpracováváním následujících bloků se bude čekat tak dlouho, dokud uvedené vřeteno nebo vřetena naprogramované v předcházejícím NC bloku s příkazem SPOSA nedosáhnou své koncové pozice (s jemným přesným najetím).		
	WAITS po M5:	Čekání, dokud se uvedená vřetena nezastaví.	
	WAITS po M3/M4:	Čekání, dokud uvedená vřetena nedosáhnou svých požadovaných otáček.	
	<n>,<m>:	Čísla vřeten, pro která má příkaz synchronní akce platit Pokud číslo vřetena není uvedeno nebo pokud je jako číslo vřetena zadána "0", vztahuje se příkaz WAITS na řídící vřeteno.	

Poznámka

V jednom bloku se mohou vyskytovat 3 příkazy polohování vřetena.

Poznámka

Když je použito inkrementálního zadávání rozměrů IC(<hodnota>), je možné polohování vřetena i přes více otáček.

Poznámka

Pokud byl před příkazem SPOS aktivován režim polohové regulace příkazem SPCON, zůstává tento režim zachován až do zadání příkazu SPCOF.

Poznámka

Na základě posloupnosti programových příkazů řídící systém samostatně rozpozná přechod do osového režimu. Explicitní naprogramování příkazu M70 ve výrobním programu není proto v zásadě nezbytné. Příkaz M70 však může být i přesto naprogramován, např. proto, aby se zlepšila čitelnost výrobního programu.

Další informace

Nastavování polohy pomocí příkazu SPOSA

Přechod na další blok, příp. zpracování programu, nejsou příkazem SPOSA nijak ovlivněny. Polohování vřetena může probíhat souběžně se zpracováváním následujících NC-bloků. Přechod na další blok se uskuteční, když dosáhnou svého kritéria konce bloku všechny v bloku naprogramované funkce (kromě vřetena). Polohování vřetena se přitom může protáhnout přes několik bloků (viz WAITS).

Poznámka

Jestliže je v následujícím bloku načten příkaz, který provádí zastavení interního předběžného zpracování, bude obrábění v tomto bloku pozastaveno tak dlouho, dokud se nezastaví všechna polohovaná vřetena.

Nastavování polohy pomocí příkazů SPOS / M19

Přechod na další blok se uskuteční, až když všechny v bloku naprogramované funkce dosáhly svého kritéria konce bloku (např. všechny pomocné funkce byly potvrzeny z PLC, všechny osy dosáhly svého koncového bodu) a pokud vřeteno dosáhlo naprogramované pozice.

Rychlost pohybů:

Rychlost a chování zpoždění pro nastavování polohy jsou uloženy ve strojních parametrech. Hodnoty nastavené v konfiguraci mohou být změněny jednak programovými příkazy, ale i synchronními akcemi, viz:

- Posuv pro polohovací osy/vřetena (FA, FPR, FPRAON, FPRAOF) (Strana 123)
- Programovatelná korekce zrychlení (ACC) (Strana 127)

Specifikace poloh vřetena:

Protože příkazy `G90`/`G91` se zde neuplatňují, explicitně platí odpovídající specifikace rozměrů, jako např. `AC`, `IC`, `DC`, `ACN` nebo `ACP`. Bez této specifikace se pohyb bude automaticky provádět, jako kdyby bylo zadáno `DC`.

Synchronizace pohybů vřetena pomocí příkazu WAITS

Pomocí příkazu `WAITS` může být v NC programu označeno místo, na němž se bude čekat tak dlouho, dokud jedno nebo více vřeten naprogramovaných v předešlém NC-bloku pomocí příkazu `SPOSA` nedosáhne své požadované pozice.

Příklad:

Programový kód	**Komentář**
`N10 SPOSA[2]=180 SPOSA[3]=0`	
`...`	
`N40 WAITS(2,3)`	`; v bloku se bude tak dlouho čekat, dokud vřetena 2 a 3 nedosáhnou svých pozic uvedených v bloku N10`

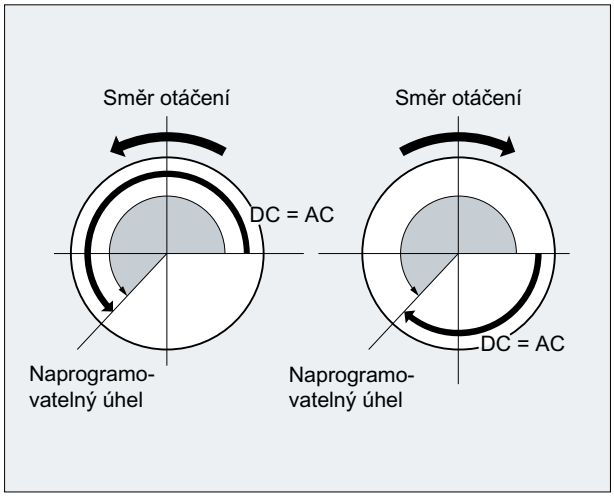
Po M5 je možné pomocí příkazu WAITS počkat, dokud se vřeteno nebo vřetena nezastaví. Po příkazu M3/M4 je možné pomocí WAITS počkat, dokud vřeteno nebo vřetena nedosáhnou požadovaných otáček/směru otáčení.

Poznámka

Pokud vřeteno dosud není synchronizováno pomocí synchronizační značky, potom se kladný směr otáčení přebírá ze strojního parametru (stav při dodávce).

Polohování vřetena z otáčení (M3/M4)

Když je aktivní funkce M3 nebo M4, vřeteno se zastaví na požadované hodnotě.



Mezi příkazy DC a AC není žádný rozdíl. V obou případech se vřeteno bude otáčet ve směru zvoleném příkazy M3/M4, dokud se nenastaví do požadované absolutní koncové polohy.
V případě příkazů ACN a ACP se vřeteno v případě potřeby zastaví a odpovídající směr najíždění zůstane zachován. Při zadání IC se vřeteno ze stávající pozice pootočí o uvedenou hodnotu.

Polohování vřetena z klidové polohy (M5)

Vřeteno se z klidu (M5) posune přesně po naprogramované dráze.

3.7.5 Posuv pro polohovací osy/vřetena (FA, FPR, FPRAON, FPRAOF)

Kromě toho existuje i možnost z pohybu nějaké jiné kruhové osy nebo vřetena odvodit otáčkový posuv pro dráhové a synchronní osy nebo pro jednotlivé polohovací osy/vřetena.

Polohovací osy, jako např. systémy pro podávání obrobků, revolverový zásobník nebo lunety, jsou ovládány nezávisle na dráhových a synchronních osách. Z tohoto důvodu se pro každou polohovací osu definuje vlastní hodnota posuvu.

Také pro vřetena může být naprogramována jejich vlastní hodnota axiálního posuvu.

Syntaxe

Posuv pro polohovací osu:

`FA[<osa>]=...`

Axiální posuv pro vřeteno:
`FA[SPI(<n>)]=...`
`FA[S<n>]=...`

Odvození otáčkového posuvu pro dráhovou/synchronní osu:

`FPR(<kruhová osa>)`

`FPR(SPI(<n>))`

`FPR(S<n>)`

Odvození otáčkového posuvu pro polohovací osy/vřetena:

`FPRAON(<osa>,<kruhová osa>)`

`FPRAON(<osa>,SPI(<n>))`

`FPRAON(<osa>,S<n>)`

`FPRAON(SPI(<n>),<kruhová osa>)`

`FPRAON(S<n>,<kruhová osa>)`

`FPRAON(SPI(<n>),SPI(<n>))`

`FPRAON(S<n>,S<n>)`

`FPRAOF(<osa>,SPI(<n>),...)`

`FPRAOF(<osa>,S<n>,...)`

Význam

FA[...]=...	Posuv pro uvedenou polohovací osu, příp rychlost polohování (axiální posuv) pro uvedené vřeteno	
	Jednotka:	mm/min, příp. palce/min nebo stupně/min
	Rozsah hodnot:	... 999 999,999 mm/min, stupňů/min ... 39 999,9999 palců/min
`FPR(...)`:	Pomocí příkazu `FPR` je identifikována kruhová osa (`<kruhová osa>`) nebo vřeteno (`SPI(<n>)` / `S<n>`)od které má být odvozen otáčkový posuv naprogramovaný pomocí příkazu `G95` pro otáčkový posuv dráhový os a synchronních os.	
`FPRAON(...)`:	Odvození otáčkového posuvu pro polohovací osy a vřetena První parametr (`<osa>` / `SPI(<n>)` / `S<n>`) identifikuje polohovací osu/vřeteno, které se mají pohybovat rychlostí otáčkového posuvu. Druhý parametr (`<kruhová osa>` / `SPI(<n>)` / `S<n>`) identifikuje kruhovou osu/vřeteno, od kterých má být otáčkový posuv odvozen. **Upozornění:** Druhý parametr může také odpadnout, pak bude posuv odvozen od hlavního vřetena.	
`FPRAOF(...)`:	Pomocí příkazu `FPRAOF` se odvozený otáčkový posuv pro uvedené osy nebo vřetena zruší.	
`<osa>`:	Identifikátor osy (polohovací nebo geometrická osa)	

SPI(<n>) / S<n>:	Identifikátor vřetena Funkce řetězců SPI(<n>) a S<n> je identická.	
	<n>:	Číslo vřetena
	Upozornění: SPI konvertuje číslo vřetena na identifikátor osy. Předávaný parametr (<n>) musí obsahovat platné číslo vřetena.	

Poznámka

Naprogramovaný posuv FA[...] má modální platnost.

V jednom NC bloku smí být naprogramováno max. 5 posuvů pro polohovací osy/vřetena.

Poznámka

Odvozený posuv se vypočítá podle následujícího vzorce:

Odvozený posuv = naprogramovaný posuv * absolutní hodnota řídícího posuvu

Příklady

Příklad 1: Vazba synchronizovaných vřeten

V případě synchronní vazby vřeten může být rychlost polohování vlečného vřetena naprogramována nezávisle na hlavním vřetenu – např. kvůli polohování.

Programový kód	Komentář
...	
FA[S2]=100	; rychlost polohování vlečného vřetena (vřeteno 2) = 100 stupňů/min
...	

Příklad 2: Odvozený otáčkový posuv pro dráhové osy

Dráhové osy X,Y se mají pohybovat s otáčkovým posuvem, který má být odvozen od pohybu kruhové osy A:

Programový kód
...
N40 FPR(A)
N50 G95 X50 Y50 F500
...

Příklad 3: Odvození otáčkového posuvu pro řídící vřeteno

Programový kód	Komentář
N30 FPRAON(S1,S2)	; otáčkový posuv pro řídící vřeteno (S1) má být odvozen od vřetena 2

Programový kód	Komentář
N40 SPOS=150	; nastavení polohy řídícího vřetena
N50 FPRAOF(S1)	; deaktivování odvozeného otáčkového posuvu pro řídící vřeteno

Příklad 4: Odvození otáčkového posuvu pro polohovací osu

Programový kód	Komentář
N30 FPRAON(X)	; otáčkový posuv pro polohovací osu X má být odvozen od řídícího vřetena
N40 POS[X]=50 FA[X]=500	; polohovací osa se má pohybovat rychlostí 500 mm/otáčku řídícího vřetena
N50 FPRAOF(X)	

Další informace

FA[...]

Platí vždy druh posuvu G94. Pokud je aktivní G70/G71, řídící se měřicí jednotky, zda jde o metrické jednotky nebo palce, podle předem definovaného nastavení ve strojním parametru. Pomocí příkazů G700/G710 je možné měřicí jednotky v programu měnit.

Poznámka

Jestliže FA není naprogramováno, platí hodnota nastavená předem ve strojním parametru.

FPR(...)

Jakožto rozšíření příkazu G95 (otáčkový posuv vztažený na řídící vřeteno) umožňuje příkaz FPR, aby byl otáčkový posuv odvozen od kterékoli kruhové osy nebo vřetena. Příkaz G95 FPR(...) platí pro dráhové a synchronizované osy.

Jestliže kruhová osa/vřeteno označené pomocí FPR pracují v režimu polohové regulace, namísto spojení pomocí skutečné hodnoty se používá spojení pomocí požadované hodnoty.

FPRAON(...)

Pomocí příkazu FPRAON je možné axiálně odvozovat otáčkový posuv určitých polohovacích os a vřeten od momentálního posuvu jiné kruhové osy nebo vřetena.

FPRAOF(...)

Příkazem FPRAOF se dá otáčkový posuv pro jedno nebo více společně se pohybujících vřeten nebo os deaktivovat.

3.7.6 Programovatelná korekce posuvu (OVR, OVRRAP, OVRA)

Rychlost dráhových a polohovacích os a vřeten je možno v NC programu modifikovat.

Syntaxe

```
OVR=<hodnota>
OVRRAP=<hodnota>
OVRA[<osa>]=<hodnota>
OVRA[SPI(<n>)]=<hodnota>
OVRA[S<n>]=<hodnota>
```

Význam

OVR:	Změna hodnoty posuvu pro posuv po dráze F	
OVRRAP:	Změna hodnoty rychlosti rychlého posuvu	
OVRA:	Změna posuvu pro polohovací posuv FA, příp. pro otáčky vřetena S	
<osa>:	Identifikátor osy (polohovací nebo geometrická osa)	
SPI(<n>) / S<n> :	Identifikátor vřetena Funkce řetězců SPI(<n>) a S<n> je identická.	
	<n>:	Číslo vřetena
	Poznámka: SPI konvertuje číslo vřetena na identifikátor osy. Předávaný parametr (<n>) musí obsahovat platné číslo vřetena.	
<hodnota>:	Změna hodnoty posuvu v procentech Tato hodnota se vztahuje na korigovaný posuv nastavený na ovládacím panelu stroje (override), příp. se s ním zkombinuje.	
	Rozsah hodnot:	0 ... 200%, jen celá čísla
	Poznámka: V případě korekce posuvu po dráze a rychlého posuvu nebude překročena maximální rychlost nastavená ve strojních parametrech.	

3.7.7 Programovatelná korekce zrychlení (ACC)

V kritických úsecích programu se může ukázat jako nezbytné omezit zrychlení na maximální možnou hodnotu, např. aby se zabránilo mechanickým kmitům.

Pomocí naprogramované korekce zrychlení může být pro každou dráhovou osu nebo vřeteno změněna pomocí příkazu v NC programu hodnota zrychlení. Omezení se vztahuje na všechny druhy interpolace. Jako 100 % zrychlení platí hodnota nastavená ve strojních parametrech.

Syntaxe

```
ACC[<osa>]=<hodnota>
ACC[SPI(<n>)]=<hodnota>
ACC(S<n>)=<hodnota>
```

Deaktivování:
`ACC[...]=100`

Syntaxe

ACC:	Změna zrychlení pro uvedenou dráhovou osu, příp. změna otáček pro uvedené vřeteno.	
<osa>:	Kanálový název dráhové osy	
SPI(<n>) / S<n>:	Identifikátor vřetena Funkce řetězců SPI(<n>) a S<n> je identická.	
	<n>:	Číslo vřetena
	Upozornění: SPI konvertuje číslo vřetena na identifikátor osy. Předávaný parametr (<n>) musí obsahovat platné číslo vřetena.	
<hodnota>:	Změna zrychlení v procentech Tato hodnota se vztahuje na korigovaný posuv nastavený na ovládacím panelu stroje (override), příp. se s ním zkombinuje.	
	Rozsah hodnot:	1...200%, jen celá čísla

Poznámka

Při vyšších hodnotách zrychlení může dojít k překročení maximálních přípustných hodnot stanovených výrobcem stroje.

Příklad:

Programový kód	**Komentář**
N50 ACC[X]=80	; saně osy ve směru X se nyní smí pohybovat pouze s 80% svého zrychlení
N60 ACC[SPI(1)]=50	; vřeteno 1 má zrychlovat nebo zpomalovat pouze s 50% svého maximálního zrychlení

Další informace

Příkazem ACC naprogramovaná korekce zrychlení

Korekce zrychlení naprogramovaná pomocí příkazu ACC[...] se vždy zohledňuje při výstupu systémové proměnné $AA_ACC, jak bylo výše popsáno. Při NC zpracování se čtení této proměnné uskutečňuje ve výrobním programu a při synchronních akcích v různých dobách.

Ve výrobním programu

Hodnota zapsaná ve výrobním programu je považována za hodnotu zapsanou výrobním programem v systémové proměnné $AA_ACC jen tehdy, pokud hodnota ACC nebyla mezitím změněna synchronní akcí.

V synchronních akcích

V souladu s výše uvedeným platí: Hodnota zapsaná v synchronní akci se bere v úvahu jako hodnota v systémové proměnné $AA_ACC zapsaná synchronizovanou akcí, pokud mezitím hodnota ACC nebyla změněna výrobním programem.

Stanovená hodnota zrychlení může být změněna také pomocí synchronních akcí.

Příklad:

Programový kód

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

Aktuální hodnota zrychlení může být zjištěna prostřednictvím systémové proměnné $AA_ACC[<osa>]. Prostřednictvím strojního parametru může být nastaveno, zda po resetu/ konci výrobního programu má být v platnosti naposled nastavená hodnota ACC nebo 100%.

3.7.8 Posuv s korekcí ručním kolečkem (FD, FDA)

Pomocí příkazů FD a FDA je možné osami v průběhu zpracovávání výrobního programu pohybovat ručními kolečky. Naprogramované pracovní posuvové pohyby os jsou přitom superponovány s impulzy ručního kolečka, které jsou vyhodnocovány jako zadané hodnoty dráhy nebo rychlosti.

Dráhové osy
V případě dráhových os může mýt korigován naprogramovaný posuv po dráze. Vyhodnocuje se přitom ruční kolečko 1. geometrické osy v kanálu. Impulzy ručního kolečka vyhodnocované v závislosti na směru otáčení na jeden takt IPO odpovídají rychlosti pohybu po dráze, která má být korigována. Mezní hodnoty rychlosti pohybu po dráze, které lze korekcí pomocí ručního kolečka dosáhnout, jsou následující:

- Minimum: 0
- Maximum: Mezní hodnota uložená ve strojních parametrech pro dráhové osy podílející se na daném pohybu

Poznámka

Posuv po dráze

Dráhový posuv F a posuv ručním kolečkem FD nesmí být naprogramovány společně v jednom NC bloku.

Polohovací osy
U polohovacích os může být axiálně korigována buď dráha posuvu nebo rychlost. Vyhodnocováno je přitom ruční kolečko přiřazené dané ose.

- Korekce dráhy
  Impulzy ručního kolečka vyhodnocované v závislosti na směru otáčení odpovídají dráze, kterou má osa urazit. Berou se přitom v úvahu pouze impulzy ručního kolečka ve směru naprogramované pozice.
- Korekce rychlosti
  Impulzy ručního kolečka vyhodnocované v závislosti na směru otáčení na jeden takt IPO odpovídají rychlosti osy, která má být korigována. Mezní hodnoty rychlosti pohybu po dráze, které lze korekcí pomocí ručního kolečka dosáhnout, jsou následující:
  - Minimum: 0
  - Maximum: Mezní hodnota polohovací osy uložená ve strojním parametru

Syntaxe

```
FD=<rychlost>
FDA[<osa>]=<rychlost>
```

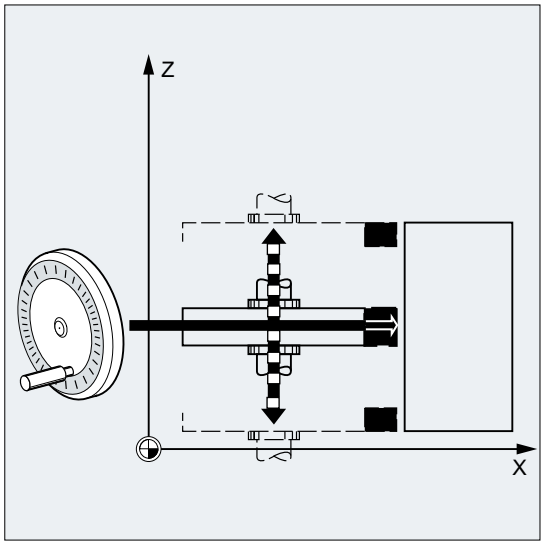
Význam

`FD=< rychlost >:`	Posuv po dráze a odblokování korekce rychlosti pomocí ručního kolečka. <rychlost>: • hodnota = 0: Není povoleno! • Hodnota ≠ 0: Rychlost pohybu po dráze
`FDA[<osa>]=<rychlost>:`	Posuv pro osu <rychlost>: • hodnota = 0: Zadaná hodnota dráhy pomocí ručního kolečka • Hodnota ≠ 0: rychlost osy
`<osa>`:	Identifikátor polohovací osy

Poznámka

`FD` a `FDA` mají blokovou platnost.

Příklad:



Zadání dráhy: S brusným kotoučem pohybujícím se tam a zpět ve směru osy Z se pomocí ručního kolečka ve směru osy X najíždí na obrobek.

Obsluhující pracovník přitom může kotouč manuálně přisunout, až dosáhne stejnoměrného odlétávání jisker. Aktivováním „vymazání zbytkové dráhy" se přejde na následující NC-blok a zpracování bude pokračovat v automatickém režimu.

Další informace

Pohyb dráhovými osami s korekcí rychlosti (FD=<rychlost>)
Pro blok výrobního programu, ve kterém je naprogramována korekce rychlosti pohybu po dráze, musí být splněny následující předpoklady:

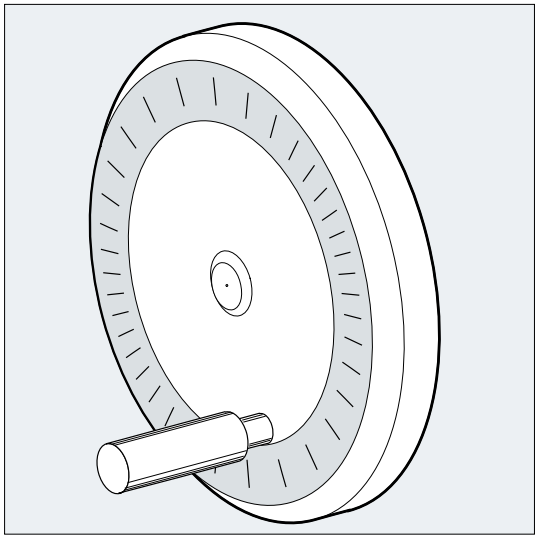
- Příkaz pohybu po dráze `G1`, `G2` nebo `G3` je aktivní.
- Přesné najetí `G60` je aktivní.
- Lineární posuv `G94` je aktivní.

Korekce posuvu
Korekce (override) posuvu má vliv pouze na naprogramovanou rychlost pohybu po dráze, nikoli na složku rychlosti vytvořenou ručním kolečkem (výjimka: override posuvu = 0).

Příklad:

Programový kód	Popis
`N10 X… Y… F500`	`; posuv po dráze = 500 mm/min`
`N20 X… Y… FD=700`	`; posuv po dráze = 700 mm/min a korekce (override) rychlosti pomocí ručního kolečka.`
	`; v bloku N20 je rychlost 500 mm/min zvýšena na 700 mm/min Pomocí ručního kolečka je možné měnit rychlost pohybu po dráze v závislosti na směru otáčení mezi 0 a maximální hodnotou (strojní parametry).`

Ovládání pohybu polohovacích os zadáním dráhy (FDA[<osa>]=0)
V NC bloku, v němž je naprogramován příkaz `FDA[<osa>]=0`, se posuv nastavuje na nulu, takže program neuskutečňuje žádné posuvové pohyby. Naprogramovaný pohyb na cílovou pozici je nyní řízen výlučně obsluhou otáčením ručního kolečka.



Příklad:

Programový kód	Popis
`...`	

Programový kód	Popis
N20 POS[V]=90 FDA[V]=0	; cílová pozice = 90 mm, posuv osy = 0 mm/min a Korekce dráhy pomocí ručního kolečka. ; rychlost pohybu osy V na začátku bloku = 0 mm/min ; zadání hodnot pro dráhu a rychlost zajišťují impulzy ručního kolečka

Směr pohybu, rychlost posuvu:
Osy se pohybují v souladu se svým znaménkem po dráze zadané impulzy ručního kolečka. V závislosti na směru otáčení je možné spouštět pohyb dopředu a dozadu. Čím rychleji ručním kolečkem otáčíte, tím vyšší je rychlost posuvu.

Rozsah posuvu:
Rozsah pohybu je omezen počáteční pozicí a naprogramovaným koncovým bodem.

Ovládání pohybu polohovacích os zadáním korekce rychlosti (FDA[<osa>]=<rychlost>)

V NC-bloku s naprogramovaným příkazem `FDA[...]=...` se hodnota posuvu naposled naprogramovaná pomocí `FA` zrychlí nebo zpomalí na hodnotu naprogramovanou pomocí příkazu `FDA`. Vychází se z právě platného posuvu a pomocí příkazu `FDA` je možno naprogramovaný pohyb na cílovou pozici otáčením ručního kolečka zrychlit nebo zpomalit, a to až na nulu. Jako maximální rychlost platí hodnoty stanovené ve strojních parametrech.

Příklad:

Programový kód	Popis
N10 POS[V]=… FA[V]=100	; posuv osy = 100 mm/min
N20 POS[V]=100 FAD[V]=200	; cílová pozice osy = 100, posuv osy = 200 mm/min ; a korekce rychlosti pomocí ručního kolečka ; v bloku N20 je rychlost 100 mm/min zvýšena na 200 mm/min Prostřed-nictvím ; ručního kolečka je možno v závislosti na směru otáčení měnit ; rychlost mezi 0 a maximální hodnotou (strojní parametry).

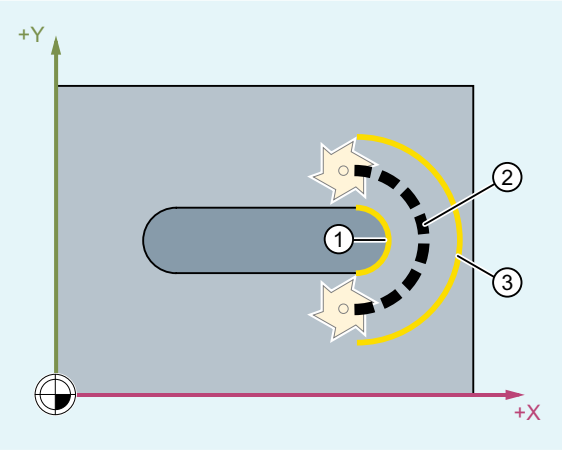
Rozsah posuvu:
Rozsah pohybu je omezen počáteční pozicí a naprogramovaným koncovým bodem.

3.7.9 Optimalizace posuvu na zakřivených úsecích dráhy (CFTCP, CFC, CFIN)

Naprogramovaný posuv se vztahuje při aktivovaném zohledňování korekcí G41/G42 pro rádius frézy zpočátku na dráhu středu frézy (srov. kapitola "Transformace souřadného systému (Frame) (Strana 307)").

Při frézování po kruhové dráze (totéž platí i pro polynomickou a splinovou interpolaci) se mění posuv na okraji frézy za určitých okolností tak silně, že to může mít vliv na výsledek obrábění.

Příklad: Frézování malého vnějšího rádiusu nástrojem o větším průměru. Dráha, kterou vnější hrana frézy musí urazit, je mnohem delší než dráha podél kontury.



V důsledku toho je kontura obráběna s velmi malým posuvem. Aby se takovým efektům zabránilo, měl by být posuv po zakřivených konturách odpovídajícím způsobem regulován.

Syntaxe

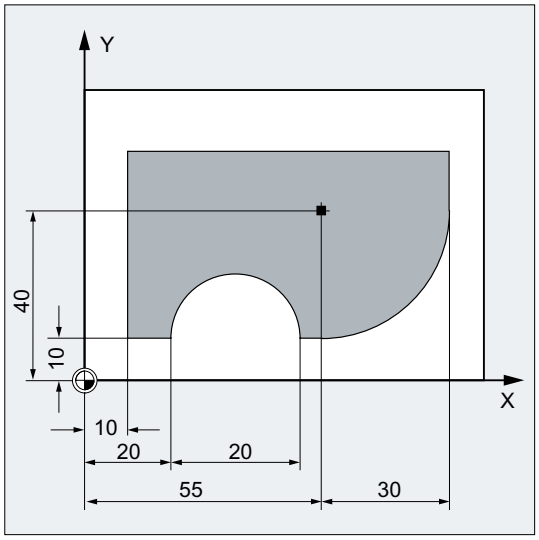
```
CFTCP
CFC
CFIN
```

Význam

CFTCP:	Konstantní posuv na dráze středu nástroje Řídící systém udržuje rychlost posuvu konstantní, korekce posuvu (override) je deaktivována.
CFC:	Konstantní posuv na kontuře (na břitu nástroje) Tato funkce je při standardním nastavení aktivní.
CFIN:	Konstantní posuv na břitu nástroje pouze na vnitřních zakřivených konturách, jinak na dráze středu frézy. Rychlost posuvu se přitom na vnitřních rádiusech snižuje.

Příklad

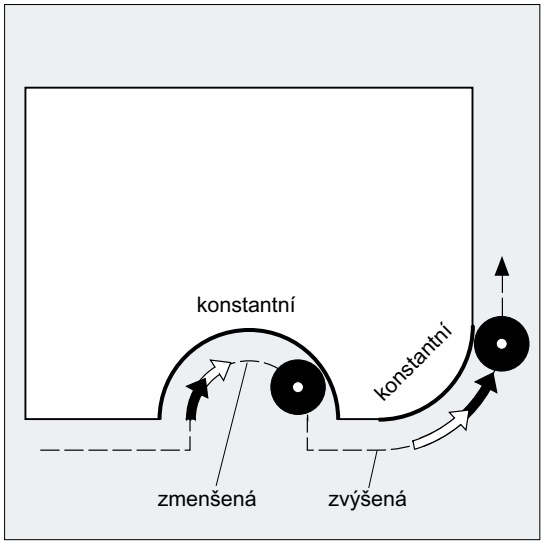
V tomto příkladu se daná kontura napřed obrábí s posuvem s korekcí CFC. Při obrábění načisto se frézovaná základna navíc opracovává s korekcí CFIN. To zabraňuje poškození obráběné základy na vnějších rozích v důsledku příliš vysoké rychlosti posuvu.



Programový kód	**Komentář**
N10 G17 G54 G64 T1 M6	
N20 S3000 M3 CFC F500 G41	
N30 G0 X-10	
N40 Y0 Z-10	; přísuv na hloubku prvního oddělování třísky
N50 KONTUR1	; volání podprogramu
N40 CFIN Z-25	; přísuv na hloubku druhého oddělování třísky
N50 KONTUR1	; volání podprogramu
N60 Y120	
N70 X200 M30	

Další informace

Konstantní posuv na kontuře s CFC



Rychlost posuvu se v případě vnitřních rádiusů snižuje, na vnějších rádiusech se zvyšuje.
V důsledku toho zůstává rychlost na břitech nástroje a tím pádem také na kontuře konstantní.

3.7.10 Větší počet hodnot posuvu v jednom bloku (F, ST, SR, FMA, STA, SRA)

Pomocí funkce "Větší počet posuvů v jednom bloku" mohou být synchronně s pohybem a v závislosti na externích digitálních a/nebo analogových vstupech aktivovány různé hodnoty posuvu v NC bloku, doba prodlevy a také zpětný pohyb.

Syntaxe

Pohyb po dráze:
```
F=... F7=... F6=... F5=... F4=... F3=... F2=... ST=... SR=...
```

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

Význam

`F=...`:	Do adresy `F` je naprogramován posuv po dráze, který platí, dokud se neobjeví nějaký vstupní signál.	
	Platnost:	modální

F2=... až F7=... :	Navíc k tomuto posuvu po dráze může být v bloku naprogramováno až 6 dalších hodnot posuvu. Numerické rozšíření udává číslo bitu vstupu, jehož změnou je posuv aktivován.	
	Platnost:	bloková
ST=...	Doba prodlevy v s (při technologii broušení: doba vyjiskřování)	
	Bit vstupu:	1
	Platnost:	bloková
SR=...	Návratová dráha Jednotky pro návratovou dráhu se vztahují na momentálně platné měřicí jednotky (mm nebo palce)	
	Bit vstupu:	0
	Platnost:	bloková
FA[<Ax>]=... :	Do adresy FA je naprogramován pohyb osy po dráze, který platí, dokud se neobjeví nějaký vstupní signál.	
	Platnost:	modální
FMA[2,<Ax>]=... až FMA[7,<Ax>]=... :	Navíc kromě hodnoty posuvu osy FA je možné pomocí příkazu FMA v daném bloku naprogramovat až 6 dalších posuvů na každou osu. První parametr udává číslo bitu vstupu, druhé osu, pro kterou má posuv platit.	
	Platnost:	bloková
STA[<Ax>]=...:	Doba prodlevy osy v s (při technologii broušení: doba vyjiskřování)	
	Bit vstupu:	1
	Platnost:	bloková
SRA[<Ax>]=...:	Dráha zpětného pohybu osy	
	Bit vstupu:	0
	Platnost:	bloková
<Ax>:	Osy, pro které má posuv platit.	

Poznámka

Priorita signálů

Posloupnost při zjišťování stavů signálů začíná vstupním bitem 0 (E0) a pokračuje vzestupně. V důsledku toho má zpětný pohyb nejvyšší a posuv F7 nejnižší prioritu. Doba prodlevy a zpětný pohyb ukončí posuvové pohyby, které byly aktivovány pomocí příkazů F2 až F7.

Aktuální posuv je určen signálem s nejvyšší prioritou.

Poznámka

Vymazání zbytkové dráhy

Jestliže je aktivován bit vstupu 1 pro dobu prodlevy, příp. pro zpětný pohyb (bit 0), bude zbytková dráha pro dráhové osy nebo příslušné jednotlivé osy vymazána a spustí se doba prodlevy, příp. zpětný pohyb.

Poznámka

Návratová dráha

Jednotky pro návratovou dráhu se vztahují na momentálně platné měřicí jednotky (mm nebo palce)

Zpětný zdvih se vždy provádí v opačném směru, než v jakém probíhá momentální pohyb. Pomocí příkazů SR/SRA je vždy programována vzdálenost zpětného zdvihu. Neprogramuje se žádné znaménko.

Poznámka

POS místo POSA

Pokud jsou pro osu naprogramovány posuvy, doba prodlevy nebo zpětný pohyb na základě externího vstupu, nesmí být tato osa naprogramována ve stejném bloku jako osa POSA (polohovací osy přes hranice bloku).

Poznámka

Dotaz na stav

Prostřednictvím synchronních příkazů pro odlišné osy je možné zjistit také stav určitého vstupu.

Poznámka

Funkce Look Ahead

Funkce Look-Ahead pracuje i při více posuvech v jednom bloku. Tak může být pomocí Look-Ahead aktuální posuv omezen.

Příklady

Příklad 1: Pohyb po dráze

Programový kód	Komentář

`G1 X48 F1000 F7=200 F6=50 F5=25 F4=5 ST=1.5 SR=0.5`	`; posuv po dráze = 1000`
	`; doplňkové hodnoty posuvu:`
	`; 200 (vstupní bit 7)`
	`; 50 (vstupní bit 6)`
	`; 25 (vstupní bit 5)`
	`; 5 (vstupní bit 4)`
	`; doba prodlevy 1,5 s`
	`; zpětný pohyb 0,5 mm`

Příklad 2: Pohyb osy

Programový kód	Komentář
POS[A]=300 FA[A]=800 FMA[7,A]=720 FMA[6,A]=640 FMA[5,A]=560 STA[A]=1.5 SRA[A]=0.5	; posuv pro osu A = 800 ; hodnoty doplňkových posuvů pro osu A: 720 (vstupní bit 7) ; 640 (vstupní bit 6) ; 560 (vstupní bit 5) ; doba prodlevy osy: 1,5 s ; zpětný pohyb osy: 0,5 mm

Příklad 3: Větší počet pracovních úkonů v jednom bloku

Programový kód	Komentář
N20 T1 D1 F500 G0 X100	Výchozí nastavení
N25 G1 X105 F=20 F7=5 F3=2.5 F2=0.5 ST=1.5 SR=0.5	; normální posuv F, ; obrábění nahrubo s F7, ; obrábění načisto s F3, ; jemné obrábění načisto s F2, ; doba prodlevy 1,5 s, ; dráha zpětného pohybu 0,5 mm
...	

3.7.11 Blokový posuv (FB)

Pomocí funkce "Blokový posuv" může být zadána samostatná hodnota posuvu pro jeden samostatný blok. Po tomto bloku je znovu aktivní předtím platný posuv modální.

Syntaxe

```
FB=<hodnota>
```

Význam

FB:	Posuv platný pouze pro aktuální blok
<hodnota>:	Naprogramovaná hodnota musí být větší než nula. Interpretace se uskutečňuje v závislosti na aktivním typu posuvu: • G94: posuv v mm/min nebo ve stupních/min • G95: posuv v mm/otáčku nebo v palcích/ot. • G96: konstantní řezná rychlost

Poznámka

Jestliže v bloku není naprogramován žádný příkaz pohybu (např. blok výpočtů), nebude mít příkaz FB žádný efekt.

Jestliže není naprogramován žádný explicitní posuv pro fasetu/rádius, platí hodnota FB i pro konturový prvek faseta/zaoblení nacházející se v tomto bloku.

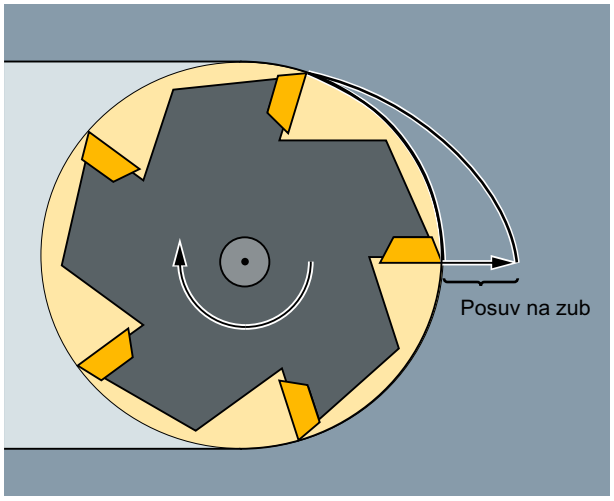
Použití posuvových interpolací FLIN, FCUB atd. není nijak omezeno.

Současné naprogramování příkazů FB a FD (manuální ovládání posuvu ručním kolečkem s korekcí posuvu) nebo F (modální posuv po dráze) **není** možné.

Příklad:

Programový kód	**Komentář**
N10 G0 X0 Y0 G17 F100 G94	; výchozí nastavení
N20 G1 X10	; posuv 100 mm/min
N30 X20 FB=80	; posuv 80 mm/min
N40 X30	; posuv je znovu 100 mm/min
...	

3.7.12 Posuv na zub (G95 FZ)



Prostřednictvím parametru nástroje $TC_DPNT (počet zubů) z datového bloku korekčních parametrů aktivního nástroje vypočítá řídící systém z naprogramovaného posuvu na zub pro každý pohybový blok platnou hodnotu otáčkového posuvu.

F = FZ * $TC_DPNT

kde:	F:	Otáčkový posuv v jednotkách mm/ot, příp. palců/ot
	FZ:	posuv na zub v mm/zub, příp. v palcích/zub
	$TC_DPNT:	Systémové proměnné pro parametry nástrojů: Počet zubů/otáčku

Na typ ($TC_DP1) aktivního nástroje se nebere ohled.

Naprogramovaná hodnota posuvu/zub je nezávislá na vyměňovaném nástroji a na aktivování/deaktivování datového bloku korekčních parametrů nástroje a má modální platnost.

Změna parametru nástroje $TC_DPNT pro aktivní břit vstupuje v platnost s následujícím vyvoláním korekčních parametrů nástroje, příp. s následující aktualizací aktivních korekčních parametrů.

Výměna nástroje a aktivování/deaktivování datového bloku korekčních parametrů nástroje má za následek nový výpočet právě platného otáčkového posuvu.

Poznámka

Posuv na zub se vztahuje jen na dráhu, programování specifických os není možné.

Syntaxe

```
G95 FZ...
```

Význam

G95:	Druh posuvu: Otáčkový posuv v jednotkách mm/ot, příp. palců/ot (v závislosti na příkazech G700/G710) pokud jde o příkaz G95, viz "Posuv (G93, G94, G95, F, FGROUP, FL, FGREF) (Strana 106)"	
FZ:	Rychlost posuvu na zub	
	Aktivování:	pomocí příkazu G95
	Platnost:	modální
	Měřicí jednotka:	mm/zub, příp. palců/zub (v závislosti na příkazech G700/G710)

UPOZORNĚNÍ
Výměna nástroje / změna řídícího vřetena
Náležitými příkazy v programu musí uživatel mít na paměti také následné změny nástroje nebo změny řídícího vřetena, např. opětovným naprogramováním příkazu FZ.

UPOZORNĚNÍ

Záběr nástroje není definován

Technologické požadavky, jako např. sousledné nebo nesousledné frézování, rovinné frézování na čelní ploše na na obvodovém plášti atd., stejně jako geometrie dráhy (přímka, kruh, ...) nejsou automaticky zohledňovány. Tyto faktory je proto nutno mít na paměti při programování posuvu na zub.

Poznámka

Přepínání mezi příkazy G95 F... a G95 FZ...

Přepnutím mezi příkazy `G95 F...` (otáčkový posuv) a `G95 FZ...` (posuv na zub) se vždy vymaže hodnota posuvu, která není aktivní.

Poznámka

Odvození hodnoty posuvu pomocí FPR

Pomocí příkazu `FPR` je možné analogicky k otáčkovému posuvu odvodit na základě pohybu libovolné kruhové osy nebo vřetena také posuv na zub (viz "Posuv pro polohovací osy/vřetena (FA, FPR, FPRAON, FPRAOF) (Strana 123)").

Příklady

Příklad 1: Fréza s 5 zuby ($TC_DPNT = 5)

Programový kód	Komentář
N10 G0 X100 Y50	
N20 G1 G95 FZ=0.02	; Posuv na zub 0,02 mm/zub
N30 T3 D1	; Výměna nástroje a aktivování datového bloku korekčních parametrů nástroje.
M40 M3 S200	; Otáčky vřetena 200 ot/min
N50 X20	; Frézování s těmito parametry:
	FZ = 0,02 mm/zub
	platný otáčkový posuv:
	F = 0,02 mm/zub* 5 zubů/ot = 0,1 mm/ot
	nebo
	F = 0,1 mm/ot * 200 ot/min = 20 mm/min
...	

Příklad 2: Přepínání mezi příkazy G95 F... a G95 FZ...

Programový kód	Komentář
N10 G0 X100 Y50	
N20 G1 G95 F0.1	; Otáčkový posuv 0,1 mm/ot
N30 T1 M6	
N35 M3 S100 D1	
N40 X20	

Programový kód	Komentář
N50 G0 X100 M5	
N60 M6 T3 D1	; Výměna a upnutí nástroje s např. 5 zuby ($TC_DPNT = 5).
N70 X22 M3 S300	
N80 G1 X3 G95 FZ=0.02	; Přepnutí z G95 F… na G95 FZ…, je aktivní posuv na zub 0,02 mm/zub.
...	

Příklad 3: Odvození hodnoty posuvu na zub od pohybu vřetena (FBR)

Programový kód	Komentář
…	
N41 FPR(S4)	; Nástroj ve vřetenu 4 (nikoli v řídícím vřetenu).
N51 G95 X51 FZ=0.5	; Posuv na zub 0,5 mm/zub v závislosti na vřetenu S4.
...	

Příklad 4: Následná výměna nástroje

Programový kód	Komentář
N10 G0 X50 Y5	
N20 G1 G95 FZ=0.03	; Posuv na zub 0,03 mm/zub
N30 M6 T11 D1	; Výměna a upnutí nástroje s např. 7 zuby ($TC_DPNT = 7).
N30 M3 S100	
N40 X30	; platný otáčkový posuv 0,21 mm/ot
N50 G0 X100 M5	
N60 M6 T33 D1	; Výměna a upnutí nástroje s např. 5 zuby ($TC_DPNT = 5).
N70 X22 M3 S300	
N80 G1 X3	; Posuv na zub 0,03 mm/zub s modální platností, platný otáčkový posuv 0,15 mm/ot
...	

Příklad 5: Změna řídícího vřetena

Programový kód	Komentář
N10 SETMS(1)	; Vřeteno 1 je řídící vřeteno.
N20 T3 D3 M6	; Výměna a upnutí nástroje 3 ve vřetenu 1.
N30 S400 M3	; Otáčky S400 vřetena 1 (a tedy i nástroje T3).
N40 G95 G1 FZ0.03	; Posuv na zub 0,03 mm/zub
N50 X50	; Pohyb po dráze, na kterém je platný posuv závislý: - Posuv na zub FZ - Otáčky vřetena 1 - Počet zubů aktivního nástroje T3
N60 G0 X60	
...	
N100 SETMS(2)	; Vřeteno 2 se stává řídícím vřetenem.
N110 T1 D1 M6	; Výměna a upnutí nástroje 1 ve vřetenu 2.
N120 S500 M3	; Otáčky S500 vřetena 2 (a tedy i nástroje T1).

Programový kód	Komentář
N130 G95 G1 FZ0.03 X20	; Pohyb po dráze, na kterém je platný posuv závislý: - Posuv na zub FZ - Otáčky vřetena 2 - Počet zubů aktivního nástroje T1

Poznámka

Po výměně řídícího vřetena (N100) musí být provedena výměna a upnutí nástroje (N110), který má být poháněn vřetenem 2.

Další informace

Přepínání mezi G93, G94 a G95

Příkaz FZ může být naprogramován, i když není aktivní funkce G95, nebude však mít žádný efekt a při aktivování příkazu G95 bude vymazán, tzn. při přepnutí mezi funkcemi G93, G94 a G95 bude analogicky s funkcí F příkaz FZ vymazán.

Opětovné aktivování funkce G95

Opětovné vyvolání funkce G95, když už je funkce G95 aktivní, nemá žádný efekt (jestliže však nebylo naprogramováno přepnutí mezi F a FZ).

Posuv s blokovou platností (FB)

Posuv s blokovou platností FB..., když jsou aktivní příkazy G95 FZ... (modální), je interpretován jako posuv na zub.

Mechanismus SAVE

U podprogramů s atributem SAVE se příkaz FZ zapisuje před spuštěním podprogramu, stejně jako je tomu v případě hodnoty F.

Více hodnot posuvu v jednom bloku

Funkci "Větší počet hodnot posuvu v jednom bloku" není možné v případě posuvu na zub používat.

Synchronní akce

Zadávání příkazu FZ ze synchronních akcí není možné.

Načítání rychlosti posuvu na zub a typu posuvu po dráze

Rychlost posuvu na zub a typ posuvu po dráze je možno načítat pomocí systémových proměnných:

- Se zastavením předběžného zpracování ve výrobním programu pomocí systémových proměnných:

	$AC_FZ	Rychlost posuvu na zub, která byla v platnosti při přípravě aktuálního bloku v hlavní větvi programu.	
	$AC_F_TYPE	Typ posuvu po dráze, který byl v platnosti při přípravě aktuálního bloku v hlavní větvi programu.	
		Hodnota:	**Význam:**
		0	mm/min
		1	mm/ot
		2	palce/min
		3	palce/ot
		11	mm/zub
		33	palce/zub

- Bez zastavení předběžného zpracování ve výrobním programu pomocí systémových proměnných:

	$P_FZ	Naprogramovaná rychlost posuvu na zub	
	$P_F_TYPE	Naprogramovaný typ posuvu po dráze	
		Hodnota:	**Význam:**
		0	mm/min
		1	mm/ot
		2	palce/min
		3	palce/ot
		11	mm/zub
		33	palce/zub

Poznámka

Pokud funkce `G95` není aktivní, obsahují proměnné $P_FZ a $AC_FZ vždy nulovou hodnotu.

3.8 Geometrická nastavení

3.8.1 Nastavitelná posunutí počátku (G54 ... G57, G505 ... G599, G53, G500, SUPA, G153)

Prostřednictvím příkazů G54 až G57 a G505 až G599 se aktivují nastavitelná posunutí počátku definovaná prostřednictvím uživatelského rozhraní, které představují posunutí souřadného systému obrobku vůči základnímu souřadnému systému.

Syntaxe

Aktivování:
```
G54
...
G57
G505
...
G599
```

Deaktivování, příp. potlačení:

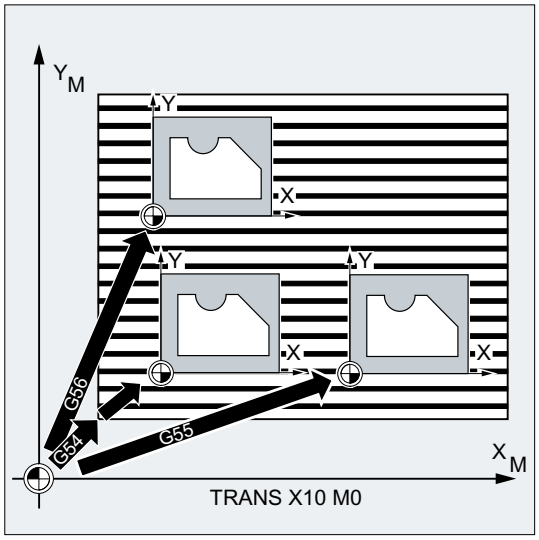
`G500` / `G53` / `G153` / `SUPA`

Význam

`G54` ... `G57`:	Vyvolání 1. až 4. nastavitelného posunutí počátku (NV)	
`G505` ... `G599`:	Vyvolání 5. až 99. nastavitelného posunutí počátku	
`G500`:	Deaktivování momentálního nastavitelného posunutí počátku	
	G500 = nulový frame: (základní nastavení, neobsahuje žádné posunutí, otočení, zrcadlové převrácení ani změnu měřítka)	Deaktivování nastavitelného posunutí počátku až do následujícího vyvolání, aktivování celkového základního framu ($P_ACTBFRAME).
	G500 se nerovná nule:	Aktivování prvního nastavitelného posunutí počátku ($P_UIFR[0]) a aktivování celkového základního framu ($P_ACTBFRAME), příp. se aktivuje eventuálně upravený základní frame.
`G53`:	Příkaz G53 blokově potlačuje naprogramovaná a nastavitelná posunutí počátku (nulového bodu).	
`G153`:	Příkaz G153 se chová jako příkaz G53 a potlačuje také celkový základní frame.	
`SUPA`:	Příkaz SUPA se chová stejně jako příkaz G153 a potlačuje kromě toho ještě i následující: • Posunutí ručním kolečkem (DRF) • Superponované pohyby • Externí posunutí počátku • Posunutí PRESET	

Příklad:

Postupně mají být opracovány tři obrobky, které jsou umístěny na paletě a které jsou uspořádány podle hodnot posunutí počátku G54 až G56. Posloupnost opracování je naprogramována v podprogramu L47.



Programový kód	**Komentář**
N10 G0 G90 X10 Y10 F500 T1	; najíždění
N20 G54 S1000 M3	; volání prvního posunutí počátku, vřeteno se otáčí vpravo
N30 L47	; zpracování programu jako podprogramu
N40 G55 G0 Z200	; volání druhého posunutí počátku, Z kvůli překonání překážky
N50 L47	; zpracování programu jako podprogramu
N60 G56	; volání třetího posunutí počátku
N70 L47	; zpracování programu jako podprogramu
N80 G53 X200 Y300 M30	; potlačení posunutí počátku, konec programu

3.8.2 Nastavitelná posunutí počátku (G54 ... G57, G505 ... G599, G53, G500, SUPA, G153): Další informace

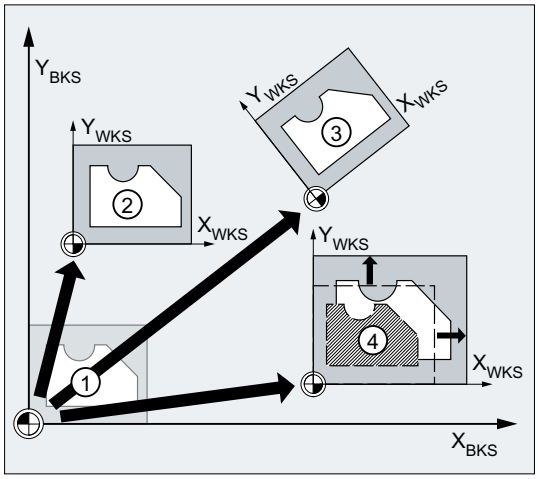
Další informace

Nastavitelné posunutí počátku

Nastavitelné posunutí počátku je v principu nastavitelný frame (Strana 307). Z tohoto důvodu jsou při zadávání nastavitelného posunutí počátku k dispozici také následující komponenty, příp. hodnoty framu:

- Posunutí
- Otočení

- Změna měřítka
- Měřítko



① Počáteční poloha v BCS
② Posunutí
③ Posunutí + otočení
④ Posunutí + změna měřítka

Hodnoty framu pro nastavitelná posunutí počátku se zadávají prostřednictvím uživatelského rozhraní:

SINUMERIK Operate: Systémová oblast „Parametry“ > "Posunutí počátku" > „Detaily"

Počet nastavitelných framů (G505 - G599) nastavitelný pomocí parametrů

Počet specifických uživatelských nastavitelných posunutí počátku (G505 - G599) je možno stanovit pro určitý kanál prostřednictvím parametru:

MD28080 $MC_MM_NUM_USER_FRAMES = <počet>

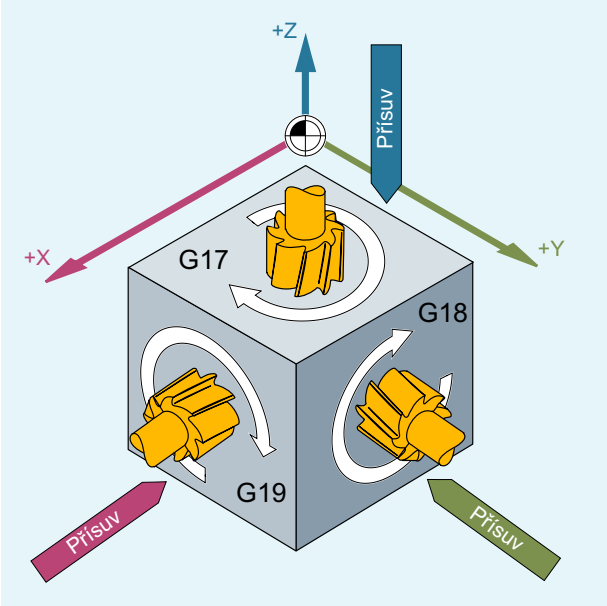
Viz také

Programovatelné posunutí počátku (G58, G59) (Strana 316)

3.8.3 Volba pracovní roviny (G17/G18/G19)

Zadáním pracovní roviny, ve které má být vyrobena požadovaná kontura, jsou současně definovány následující funkce:

- Rovina pro korekci rádiusu nástroje
- Směr přísuvu pro korekci délky nástroje v závislosti na jeho typu.
- Rovina pro kruhovou interpolaci



Syntaxe

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

Význam

G17:	Pracovní rovina X/Y Směr přísuvu Z, volba roviny, 1. – 2. geometrická osa
G18:	Pracovní rovina Z/X Směr přísuvu Y, volba roviny, 3. – 1. geometrická osa
G19:	Pracovní rovina Y/Z Směr přísuvu X, volba roviny, 2. – 3. geometrická osa

Poznámka

Při základním nastavení je pro frézování předem definována rovina G17 (rovina X/Y) a pro soustružení G18 (rovina Z/X).

Spolu s voláním korekce nástroje při pohybu po dráze G41/G42 (viz kapitola "Korekce rádiusu nástroje (Strana 253)") musí být pracovní rovina udána, aby řídící systém mohl vypočítat potřebné korekce pro délku a rádius nástroje.

Příklad:

"Klasický" postup při frézování je:

1. Definice pracovní roviny (pro frézování je základní nastavení G17).
2. Vyvolání typu nástroje (T) a hodnot korekčních parametrů nástroje (D).

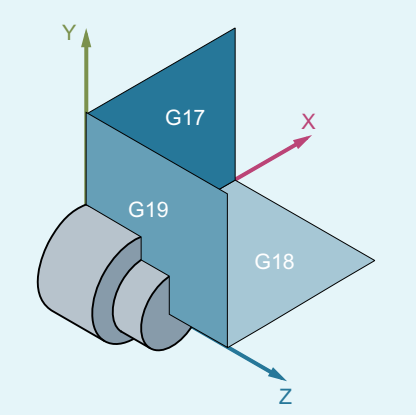
3. Aktivování dráhové korekce nástroje (G41)
4. Programování pohybů nástroje

Programový kód	Komentář
N10 G17 T5 D8	; vyvolání pracovní roviny X/Y, vyvolání nástroje Korekce délky se uskutečňuje ve směru osy Z.
N20 G1 G41 X10 Y30 Z-5 F500	; korekce rádiusu se uskutečňuje v rovině X/Y
N30 G2 X22.5 Y40 I50 J40	; Kruhová interpolace/korekce rádiusu nástroje v rovině X/Y.

Další informace

Všeobecně

Doporučuje se, aby pracovní rovina G17 až G19 byla definována hned na začátku programu. V základním nastavení je pro soustružení předdefinována příkazem G18 rovina Z/X:

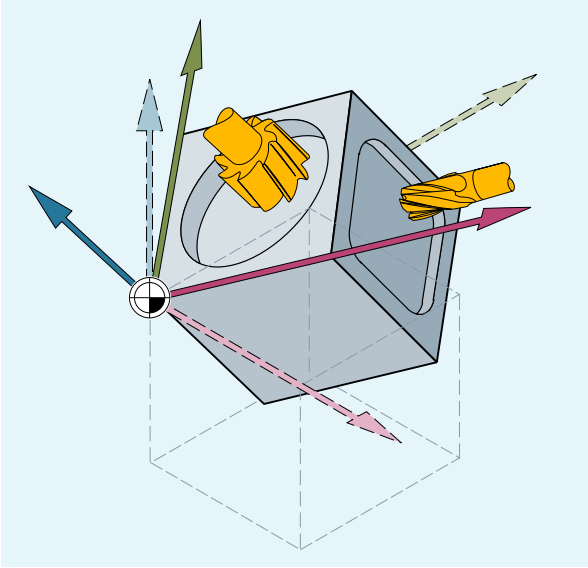


Zadání pracovní roviny potřebujete řídící systém pro výpočet směru opisování. Pokud budete potřebovat k tomuto tématu další informace, viz kapitola „Kruhová interpolace (Strana 190)“.

Obrábění na šikmo ležících rovinách

Prostřednictvím otáčení souřadného systému pomocí funkce ROT (Strana 318) nastavte souřadné osy tak, aby se kryly s šikmo položenou rovinou. Pracovní roviny se pootočí odpovídajícím způsobem.

Korekce délky nástroje na šikmých rovinách



Délková korekce nástroje se obecně vždy vztahuje na pevnou neotočenou pracovní rovinu.

Poznámka

Pomocí funkcí pro „Délkovou korekci nástrojů pro orientovatelné nástroje“ mohou být vypočítávány komponenty délky nástroje, které jsou přizpůsobeny pootočeným pracovním rovinám.

Volba roviny korekcí se uskutečňuje pomocí příkazů CUT2D, CUT2DF. Podrobnější informace k tomuto tématu a popis těchto možností výpočtu naleznete v kapitole „Korekce rádiusu nástroje (Strana 253)“.

Pro účely prostorové definice pracovní roviny nabízí řídící systém velmi pohodlné možnosti pro transformace souřadných systémů. Pokud budete potřebovat k tomuto tématu další informace, viz kapitola „Transformace souřadného systému (Frame) (Strana 307)“.

3.8.4 Údaje rozměrů

Základem většiny NC programů je výrobní výkres obrobku s konkrétními údaji rozměrů.

Tyto údaje rozměrů mohou být následující:

- v absolutních rozměrech nebo v inkrementálních rozměrech
- v milimetrech nebo v palcích
- jako rádius nebo jako průměr (v případě soustružení)

Aby bylo možné údaje přenášet z výkresu rozměrů přímo (bez přepočítávání) do NC programu, jsou uživateli k dispozici specifické programové příkazy, které nabízejí různé možnosti zadávání těchto rozměrů.

3.8.4.1 Zadávání absolutních rozměrů (G90, AC)

Při zadávání absolutních rozměrů jsou údaje polohy vždy vztaženy na počátek (nulu) momentálně platného souřadného systému, tzn. do programu se zadává absolutní pozice, na kterou má nástroj najet.

Zadávání absolutních rozměrů s modální platností

Zadávání absolutních rozměrů s modální platností se aktivuje pomocí příkazu `G90`. Tento příkaz platí pro všechny osy, které jsou naprogramovány v následujících NC blocích.

Zadávání absolutních rozměrů s blokovou platností

Jestliže bylo předtím aktivováno programování inkrementálních rozměrů (`G91`), je možné pomocí příkazu `AC` nastavit pro jednotlivé osy zadávání absolutních rozměrů s blokovou platností.

Poznámka

Zadávání absolutních rozměrů s blokovou platností (`AC`) je možné i pro nastavování polohy vřetena (`SPOS`, `SPOSA`) a pro interpolační parametry (`I`, `J`, `K`).

Syntaxe

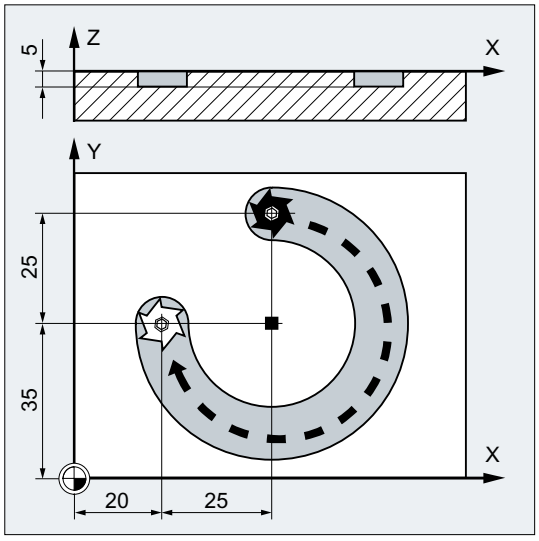
```
G90
<osa>=AC(<osa>)
```

Význam

`G90`:	Příkaz pro aktivování zadávání absolutních rozměrů s modální platností
`AC`:	Příkaz pro aktivování zadávání absolutních rozměrů s blokovou platností
`<osa>`:	Identifikátor osy, která se má pohybovat
`<hodnota>`:	Požadovaná poloha osy, která se má pohybovat, v absolutních rozměrech

Příklady

Příklad 1: Frézování

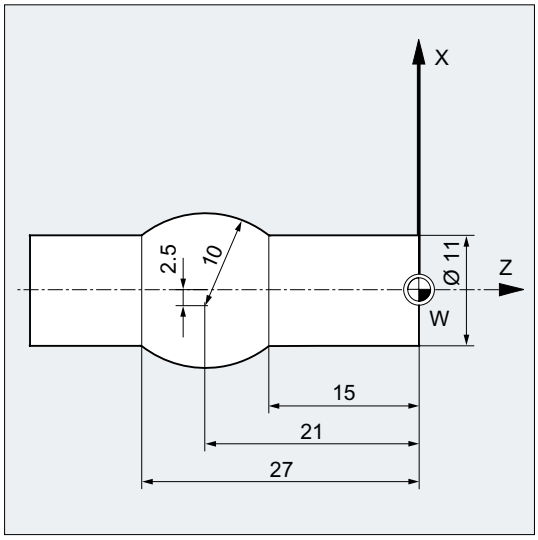


Programový kód	Komentář
`N10 G90 G0 X45 Y60 Z2 T1 S2000 M3`	`; zadávání absolutních rozměrů, rychlým posuvem na pozici XYZ, volba nástroje, zapnutí vřetena, vřeteno se otáčí vpravo`
`N20 G1 Z-5 F500`	`; přímková interpolace, přísuv nástroje`
`N30 G2 X20 Y35 I=AC(45) J=AC(35)`	`; kruhová interpolace ve směru hodinových ručiček, koncový bod kruhového oblouku a střed kruhu v absolutních rozměrech`
`N40 G0 Z2`	`; vyjíždění`
`N50 M30`	`; koncový blok`

Poznámka

Pokud budete potřebovat informace o zadávání souřadnic středu I a J, nahlédněte do kapitoly"Kruhová interpolace".

Příklad 2: Soustružení



Programový kód	Komentář
N5 T1 D1 S2000 M3	; výměna a upnutí nástroje T1, zapnutí vřetena a vřeteno se otáčí vpravo
N10 G0 G90 X11 Z1	; zadávání absolutních rozměrů, rychlý posuv na pozici XZ
N20 G1 Z-15 F0.2	; přímková interpolace, přísuv nástroje
N30 G3 X11 Z-27 I=AC(-5) K=AC(-21)	; kruhová interpolace proti směru hodinových ručiček, koncový bod kruhového oblouku a střed kruhu v absolutních rozměrech
N40 G1 Z-40	; vyjíždění
N50 M30	; koncový blok

Poznámka

Pokud budete potřebovat informace o zadávání souřadnic středu I a J, nahlédněte do kapitoly "Kruhová interpolace".

Viz také

Zadávání absolutních a inkrementálních rozměrů při soustružení a frézování (G90/G91) (Strana 157)

3.8.4.2 Zadávání inkrementálních rozměrů (G91, IC)

V případě inkrementálních rozměrů je údaj polohy vztažen na naposled naprogramovaný bod, tzn. programování v inkrementálních rozměrech udává, o kolik se má nástroj posunout.

Zadávání inkrementálních rozměrů s modální platností

Zadávání inkrementálních rozměrů s modální platností se aktivuje pomocí příkazu `G91`. Tento příkaz platí pro všechny osy, které jsou naprogramovány v následujících NC blocích.

Zadávání inkrementálních rozměrů s blokovou platností

Jestliže bylo předtím aktivováno programování absolutních rozměrů (`G90`), je možné pomocí příkazu `IC` nastavit pro jednotlivé osy zadávání inkrementálních rozměrů s blokovou platností.

Poznámka

Zadávání inkrementálních rozměrů s blokovou platností (`IC`) je možné i pro nastavování polohy vřetena (`SPOS`, `SPOSA`) a pro interpolační parametry (`I`, `J`, `K`).

Syntaxe

```
G91
<osa>=IC(<osa>)
```

Význam

`G91`:	Příkaz pro aktivování zadávání inkrementálních rozměrů s modální platností
`IC`:	Příkaz pro aktivování zadávání inkrementálních rozměrů s blokovou platností
`<osa>`:	Identifikátor osy, která se má pohybovat
`<hodnota>`:	Požadovaná poloha osy, která se má pohybovat, v inkrementálních rozměrech

Rozšíření G91

Pro účely určitých operací, jako je např. škrábnutí, je zapotřebí, aby při inkrementálních rozměrech byl posuv uskutečněn jen o naprogramovanou dráhu. Aktivní posunutí počátku nebo korekce délky nástroje nebudou posuvem provedeny.

Toto chování může být nastaveno odděleně pro aktivní posunutí počátku a pro korekci délky nástroje prostřednictvím následujících nastavovaných parametrů:

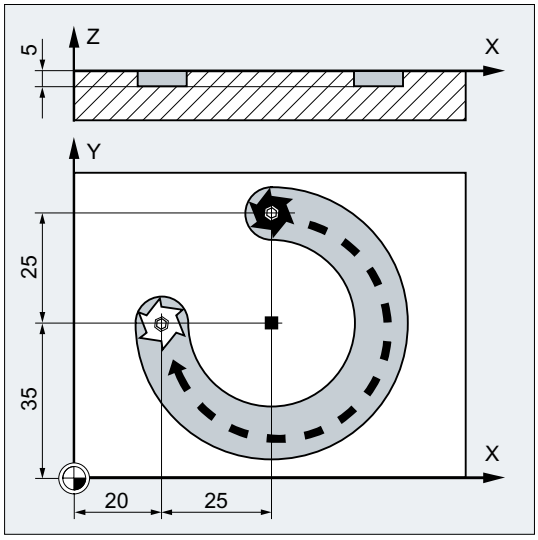
SD42440 $SC_FRAME_OFFSET_INCR_PROG (posunutí počátku ve framech)

SD42442 $SC_TOOL_OFFSET_INCR_PROG (korekce délky nástroje)

Hodnota	Význam
0	Při programování inkrementálních rozměrů (řetězové kóty) pro určitou osu se **nebude** posuvem provádět aktivní posunutí počátku, příp. korekce délky nástroje.
1	Při programování inkrementálních rozměrů (řetězové kóty) pro určitou osu se aktivní posunutí počátku, příp. korekce délky nástroje bude posuvem provádět.

Příklady

Příklad 1: Frézování

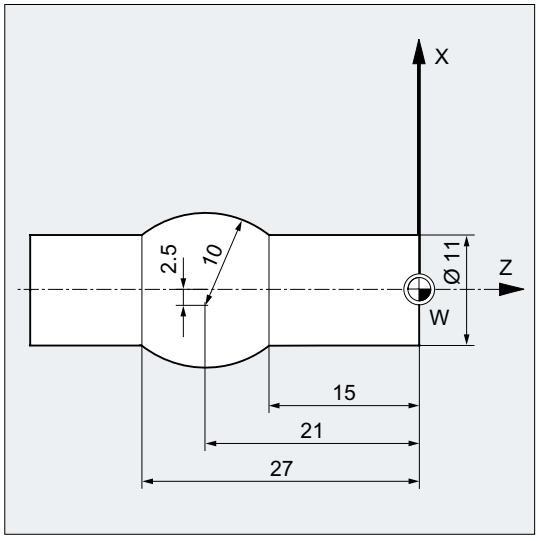


Programový kód	Komentář
`N10 G90 G0 X45 Y60 Z2 T1 S2000 M3`	`; zadávání absolutních rozměrů, rychlým posuvem na pozici XYZ, volba nástroje, zapnutí vřetena, vřeteno se otáčí vpravo`
`N20 G1 Z-5 F500`	`; přímková interpolace, přísuv nástroje`
`N30 G2 X20 Y35 I0 J-25`	`; kruhová interpolace ve směru hodinových ručiček, koncový bod kruhového oblouku v absolutních rozměrech, střed kruhu v inkrementálních rozměrech`
`N40 G0 Z2`	`; vyjíždění`
`N50 M30`	`; koncový blok`

Poznámka

Pokud budete potřebovat informace o zadávání souřadnic středu I a J, nahlédněte do kapitoly"Kruhová interpolace".

Příklad 2: Soustružení



Programový kód	Komentář
N5 T1 D1 S2000 M3	; výměna a upnutí nástroje T1, zapnutí vřetena a vřeteno se otáčí vpravo
N10 G0 G90 X11 Z1	; zadávání absolutních rozměrů, rychlý posuv na pozici XZ
N20 G1 Z-15 F0.2	; přímková interpolace, přísuv nástroje
N30 G3 X11 Z-27 I-8 K-6	; kruhová interpolace proti směru hodinových ručiček, koncový bod kruhového oblouku v absolutních rozměrech, střed kruhu v inkrementálních rozměrech
N40 G1 Z-40	; vyjíždění
N50 M30	; koncový blok

Poznámka

Pokud budete potřebovat informace o zadávání souřadnic středu I a J, nahlédněte do kapitoly"Kruhová interpolace".

Příklad 3: Zadávání inkrementálních rozměrů bez posuvu kvůli aktivnímu posunutí počátku

Nastavení:

- G54 obsahuje posunutí v ose X o 25
- SD42440 $SC_FRAME_OFFSET_INCR_PROG = 0

Programový kód	Komentář
N10 G90 G0 G54 X100	
N20 G1 G91 X10	; zadávání inkrementálních rozměrů je aktivní, pohyb ve směru X o 10 mm (posunutí počátku nebude posuvem osy provedeno)

Programový kód	Komentář
N30 G90 X50	; zadávání absolutních rozměrů je aktivní, najíždění na pozici X75 (posunutí počátku bude pohybem osy provedeno)

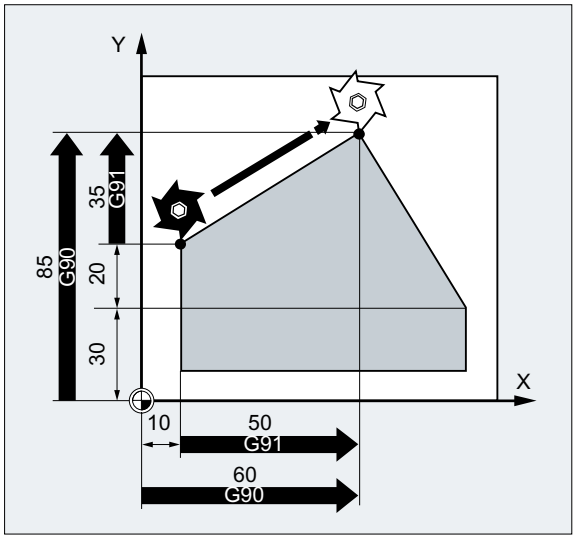
Viz také

Zadávání absolutních a inkrementálních rozměrů při soustružení a frézování (G90/G91) (Strana 157)

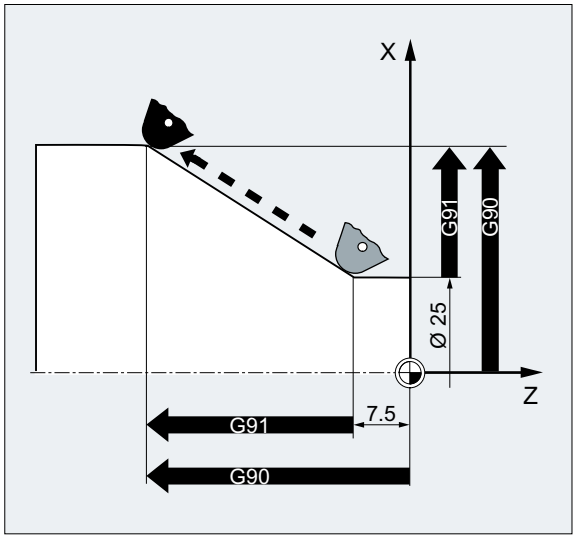
3.8.4.3 Zadávání absolutních a inkrementálních rozměrů při soustružení a frézování (G90/G91)

Následující dva obrázky názorně ukazují programování zadávání absolutních rozměrů (G90), příp. zadávání inkrementálních rozměrů (G91) na příkladu technologie soustružení a frézování.

Frézování:



Soustružení:



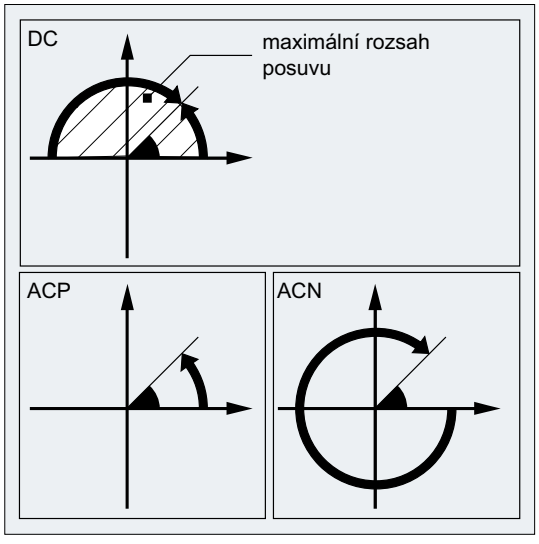
Poznámka

U konvenčních soustruhů je obvyklou praxí interpretovat inkrementální bloky posuvů ve směru příčné osy jako hodnoty rádiusu, zatímco rozměry průměrů se používají pro absolutní souřadnice. Tato konvence pro `G90` se uskutečňuje pomocí příkazů `DIAMON`, `DIAMOF` nebo `DIAM90`.

3.8.4.4 Zadávání absolutních rozměrů pro kruhové osy (DC, ACP, ACN)

Pro nastavování polohy kruhových os v absolutních rozměrech jsou k dispozici příkazy DC, ACP a ACN, které mají blokovou platnost a které jsou nezávislé na příkazech G90/G91.

Příkazy DC, ACP a ACN se v zásadě odlišují strategií najíždění:



Syntaxe

```
<kruhová osa>=DC(<hodnota>)
<kruhová osa>=ACP(<hodnota>)
<kruhová osa>=ACN(<hodnota>)
```

Význam

<kruhová osa>:	Identifikátor kruhové osy, která se má pohybovat (např. A, B nebo C).	
DC:	Příkaz pro **přímé** najíždění na pozici Kruhová osa najíždí na naprogramovanou pozici po přímé nejkratší dráze. Kruhová osa se pohybuje maximálně v rozsahu 180°.	
ACP:	Příkaz pro najíždění na danou pozici v **kladném** směru Kruhová osa najíždí na naprogramovanou pozici v kladném směru otáčení osy (proti směru hodinových ručiček).	
ACN:	Příkaz pro najíždění na danou pozici v **záporném** směru Kruhová osa najíždí na naprogramovanou pozici v záporném směru otáčení osy (ve směru hodinových ručiček).	
<hodnota>:	Absolutní údaj polohy kruhové osy, na kterou se má najet.	
	Rozsah hodnot:	0 - 360 stupňů

Poznámka

Kladný směr otáčení (ve směru nebo proti směru hodinových ručiček) se nastavuje pomocí strojního parametru.

Poznámka

Pro polohování s udáním směru pohybu (ACP, ACN) musí být ve strojním parametru nastaven rozsah pohybu mezi 0° a 360° (chování typu modulo). Jestliže chcete kruhovou osu modulo v jednom bloku pootočit o více než 360°, je zapotřebí naprogramovat G91, příp. IC.

Poznámka

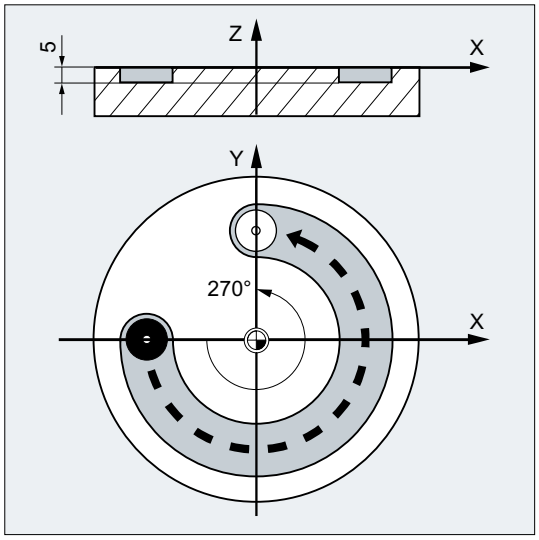
Příkazy DC, ACP a ACN se mohou používat také pro nastavování polohy zastaveného vřetena (SPOS, SPOSA).

Příklad: SPOS=DC(45)

Příklad:

Obrábění frézováním na otočném stole

Nástroj stojí, stůl se otočí o 270° **ve směru hodinových ručiček**. Vzniká přitom kruhová drážka.



Programový kód	**Komentář**
`N10 SPOS=0`	`; vřeteno v režimu regulace polohy`
`N20 G90 G0 X-20 Y0 Z2 T1`	`; zadávání absolutních rozměrů, přísuv nástroje T1 rychlým posuvem`
`N30 G1 Z-5 F500`	`; spuštění nástroje pracovním posuvem`
`N40 C=ACP(270)`	`; stůl se otočí o 270 stupňů ve směru hodinových ručiček (kladný směr), nástroj frézuje kruhovou drážku`
`N50 G0 Z2 M30`	`; pozvednutí nástroje, konec programu`

3.8.4.5 Systém měřicích jednotek palce/metrické jednotky (G70/G71, G700/G710)

Mezi systémem metrických měřicích jednotek a systémem imperiálních měřicích jednotek je možno v rámci výrobního programu přepínat pomocí příkazů ze skupiny G-funkcí 13 (systém měřicích jednotek palce/metrické jednotky).

Aktivování

Aby byly příkazy G700 a G710 k dispozici, musí být aktivovány rozšířené funkce měřicího systému (MD10260 $MN_CONVERT_SCALING_SYSTEM = 1).

Syntaxe

```
G70
G71
G700
G710
```

Význam

G70:	Přepnutí na imperiální měřicí systém (palce) **Geometrické údaje související** s délkami jsou načítány a vypisovány **v systému imperiálních jednotek (palce)**. **Technologické údaje související s délkami** (např. posuvy, korekční parametry nástroje, nastavitelná posunutí počátku, strojní parametry a systémové proměnné) jsou načítány a zapisovány v jednotkách **základního systému nastaveného v konfiguraci**.	
	G-skupina:	13
	Základní nastavení:	Může být nastaveno pomocí parametru MD20150 $MC_GCODE_RESET_VALUES
	Platnost:	Modální
G71:	Přepnutí na metrický měřicí systém **Geometrické údaje související s délkami** jsou načítány a vypisovány v **systému metrických jednotek**. **Technologické údaje související s délkami** (např. posuvy, korekční parametry nástroje, nastavitelná posunutí počátku, strojní parametry a systémové proměnné) jsou načítány a zapisovány v jednotkách **základního systému nastaveného v konfiguraci**.	
	G-skupina:	13
	Základní nastavení:	Může být nastaveno pomocí parametru MD20150 $MC_GCODE_RESET_VALUES
	Platnost:	Modální
G700:	Přepnutí na imperiální měřicí systém (palce) Všechny geometrické a technologické údaje související s délkami jsou načítány a vypisovány v systému imperiálních jednotek (palce).	
	G-skupina:	13
	Základní nastavení:	Může být nastaveno pomocí parametru MD20150 $MC_GCODE_RESET_VALUES
	Platnost:	Modální
G710:	Přepnutí na metrický měřicí systém Všechny geometrické a technologické údaje související s délkami jsou načítány a vypisovány v systému metrických jednotek.	
	G-skupina:	13
	Základní nastavení:	Může být nastaveno pomocí parametru MD20150 $MC_GCODE_RESET_VALUES
	Platnost:	Modální

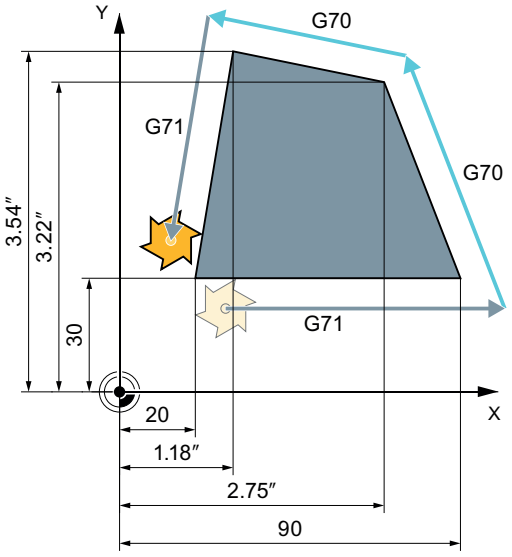
UPOZORNĚNÍ

Specifická data pro kruhové osy

Specifická data pro kruhové osy jsou jsou načítána a zapisována v jednotkách základního systému nastaveného v konfiguraci.

Příklad

Základní systém je metrický (MD10240 $MN_SCALING_SYSTEM_IS_METRIC = 1). Výkres obrobku ale obsahuje údaje rozměrů v palcích. Z tohoto důvodu se v rámci výrobního programu přepíná na imperiální systém měřicích jednotek. Po zadání rozměrů v palcích se přepne zpět do systému metrických měřicích jednotek.



Programový kód	**Komentář**
N10 G0 G90 X20 Y30 Z2 S2000 M3 T1	; X=20 mm, Y=30 mm, Z=2 mm, F=rychlý posuv v mm/min
N20 G1 Z-5 F500	; Z=-5 mm, F=500 mm/min
N30 X90	; X=90 mm
N40 **G70** X2.75 Y3.22	; naprogramovaný systém měřicích jednotek: palce
	; X=2.75 palců, Y=3.22 palců, F=500 mm/min
N50 X1.18 Y3.54	; X=1.18 palců, Y=3.54 palců, F=500 mm/min
N60 **G71** X20 Y30	; naprogramovaný systém měřicích jednotek: metrický
	; X=20 mm, Y=30 mm, F=500 mm/min
N70 G0 Z2	; Z=2 mm, F=rychlý posuv v mm/min
N80 M30	; konec programu

Další informace

Načítání a zápis dat v souvislosti s příkazy G70/G71 a G700/G710

Datová oblast	G70 / G71		G700 / G710	
	Čtení	Zapisování	Čtení	Zapisování
Zobrazování míst za desetinnou čárkou (WCS)	P	P	P	P
Zobrazování míst za desetinnou čárkou (MCS)	G	G	G	G
Posuvy	G	G	P	P
Údaje polohy X, Y, Z	P	P	P	P
Interpolační parametry I, J, K	P	P	P	P
Rádius kruhu (CR)	P	P	P	P
Polární rádius (RP)	P	P	P	P
Stoupání závitu	P	P	P	P
Programovatelný FRAME	P	P	P	P
Nastavitelný FRAME	G	G	P	P
Základní frame	G	G	P	P
Externí posunutí počátku	G	G	P	P
axiální předvolba posunutí	G	G	P	P
Ohraničení pracovního pole (G25/G26)	G	G	P	P
Chráněné oblasti	P	P	P	P
Korekční parametry nástroje	G	G	P	P
Strojní parametry související s délkami	G	G	P	P
Nastavované parametry související s délkami	G	G	P	P
Systémové proměnné související s délkami	G	G	P	P
Proměnné GUD	G	G	G	G
Proměnné LUD	G	G	G	G
Proměnné PUD	G	G	G	G
R-parametry	G	G	G	G
Cykly firmy Siemens	P	P	P	P
Vyhodnocování inkrementů v režimu JOG/Ruční kolečko	G	G	G	G
P: Zápis/čtení se uskutečňuje v naprogramovaném systému měřicích jednotek G: Zápis/čtení se uskutečňuje v základním systému nastaveném v parametrech				

Synchronní akce

Poznámka

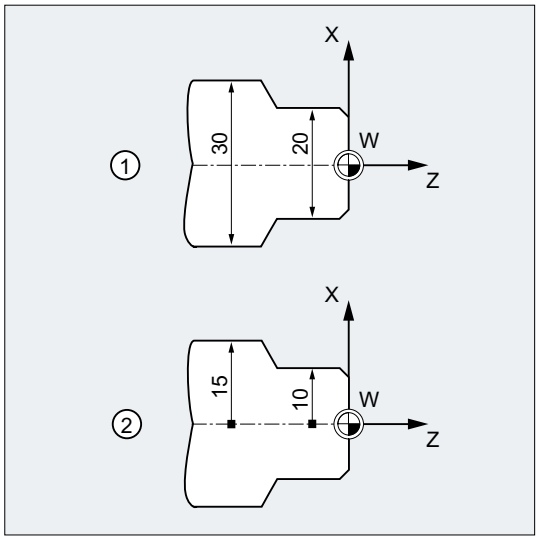
Načítání údajů polohy v synchronních akcích

Pokud v rámci synchronní akce (úsek podmínek a/nebo úsek vlastní akce, příp. technologické funkce) není explicitně naprogramován žádný měřicí systém, jsou **údaje polohy týkající se délek** v synchronní akci načítány vždy v **základním měřicím systému, který je nastaven v konfiguraci**.

Další informace: Příručka k funkcím, Synchronní akce

3.8.4.6 Programování rádiusů/průměrů ve specifickém kanálu (DIAMON, DIAM90, DIAMOF, DIAMCYCOF)

Při soustružení mohou být rozměry **pro příčnou osu** zadávány buď jako průměry (①) nebo jako rádiusy (②):



Aby bylo možné do NC programu přebírat údaje rozměrů přímo z technického výkresu bez přepočítávání. je možné prostřednictvím příkazů `DIAMON`, `DIAM90`, `DIAMOF` a `DIAMCYCOF` s modální platností pro daný kanál aktivovat programování průměrů nebo rádiusů.

Poznámka

Programování průměrů/rádiusů pro specifický kanál se vztahuje na geometrickou osu, která byla prostřednictvím parametru MD20100 $MC_DIAMETER_AX_DEF definována jako příčná osa (--> viz informace od výrobce stroje).

Pomocí strojního parametru MD20100 může být v každém kanálu definována jen jedna příčná osa.

Syntaxe

```
DIAMON
DIAM90
DIAMOF
```

Význam

DIAMON:	Příkaz pro aktivování **nezávislého** programování průměrů pro specifický kanál. Funkce příkazu DIAMON je nezávislá na naprogramovaném režimu zadávání rozměrů (zadávání absolutních rozměrů G90 nebo zadávání inkrementálních rozměrů G91):	
	• v případě G90:	Udávání rozměrů v průměrech
	• v případě G91:	Udávání rozměrů v průměrech
DIAM90:	Příkaz pro aktivování **závislého** programování průměrů pro specifický kanál. Funkce příkazu DIAM90 závisí na naprogramovaném režimu zadávání rozměrů:	
	• v případě G90:	Udávání rozměrů v průměrech
	• v případě G91:	Zadávání rozměrů v rádiusech
DIAMOF:	Příkaz pro deaktivování programování průměrů pro specifický kanál. Když je programování průměrů deaktivováno, je v platnosti programování rádiusů pro specifický kanál. Funkce příkazu DIAMOF nezávisí na naprogramovaném režimu zadávání rozměrů:	
	• v případě G90:	Zadávání rozměrů v rádiusech
	• v případě G91:	Zadávání rozměrů v rádiusech
DIAMCYCOF:	Příkaz pro deaktivování programování průměrů pro specifický kanál v průběhu zpracovávání cyklů. Takto je možné zajistit, aby se výpočty v cyklu uskutečňovaly pouze s rádiusy. Pro vypisování údajů o polohách a vypisování základního bloku zůstává aktivní naposled používaný příkaz G-funkce této skupiny.	

Poznámka

Když je aktivní příkaz DIAMON nebo DIAM90, vypisují se skutečné hodnoty pro příčnou osu vždy jako průměry. To platí také pro odečítání skutečné hodnoty v souřadném systému obrobku pomocí funkcí MEAS, MEAW, $P_EP[x] a $AA_IW[x].

Příklad:

Programový kód	**Komentář**
N10 G0 X0 Z0	; najetí na počáteční bod
N20 DIAMOF	; deaktivování programování průměrů

Programový kód	Komentář
N30 G1 X30 S2000 M03 F0.7	; osa X = příčná osa, programování rádiusů aktivní; najíždění na pozici rádiusu X30
N40 DIAMON	; pro příčnou osu je aktivováno programování průměrů
N50 G1 X70 Z-20	; najíždění na pozici průměru X70 a Z-20
N60 Z-30	
N70 DIAM90	; programování průměrů pro absolutní rozměr a programování rádiusů pro inkrementální rozměr
N80 G91 X10 Z-20	; zadávání inkrementálních rozměrů aktivováno
N90 G90 X10	; zadávání absolutních rozměrů aktivováno
N100 M30	; konec programu

Další informace

Hodnoty průměru (DIAMON/DIAM90)

Hodnoty průměru se vztahují na následující údaje:

- Výpis skutečné hodnoty příčné osy v souřadném systému obrobku
- Režim JOG: Inkrementy pro krokový posuv a manuální posuv ručním kolečkem
- Programování koncových pozic:
  Interpolační parametry `I`, `J`, `K` u příkazů `G2`/`G3`, jestliže byly tyto příkazy pomocí `AC` naprogramovány s absolutními souřadnicemi.
  V případě programování inkrementálních hodnot (`IC`) parametrů `I`, `J`, `K` jsou tyto údaje vždy započítávány jako rádius.
- Načítání skutečných hodnot v souřadném systému obrobku při použití příkazů:
  `MEAS`, `MEAW`, `$P_EP[X]`, `$AA_IW[X]`

3.8.4.7 Programování rádiusů/průměrů pro specifickou osu (DIAMONA, DIAM90A, DIAMOFA, DIACYCOFA, DIAMCHANA, DIAMCHAN, DAC, DIC, RAC, RIC)

Kromě programování průměrů pro specifický kanál je k dispozici také funkce pro programování průměrů pro specifickou osu, což vám umožňuje pomocí příkazů s modální nebo blokovou platností specifikovat a zobrazovat rozměry pro jednu nebo více os jako hodnoty průměru.

Poznámka

Programování průměrů pro specifickou osu je možné používat jen pro osy, které byly pomocí strojního parametru MD30460 $MA_BASE_FUNCTION_MASK prohlášeny za další příčné osy a bylo pro ně povoleno programování průměrů (--> viz informace od výrobce stroje!).

Syntaxe

Osové programování průměrů pro větší počet příčných os s modální platností v kanálu:

```
DIAMONA[<osa>]
DIAM90A[<osa>]
DIAMOFA[<osa>]
```

```
DIACYCOFA[<osa>]
```

Převzetí programování průměrů/rádiusů pro specifický kanál:
```
DIAMCHANA[<osa>]
DIAMCHAN
```

Programování průměrů/rádiusů s blokovou platností pro specifickou osu:
```
<osa>=DAC(<hodnota>)
<osa>=DIC(<hodnota>)
<osa>=RAC(<hodnota>)
<osa>=RIC(<hodnota>)
```

Význam

Programování průměrů s modální platností pro specifickou osu		
DIAMONA:	Příkaz pro aktivování **nezávislého** programování průměrů pro specifickou osu. Funkce příkazu DIAMONA je nezávislá na naprogramovaném režimu zadávání rozměrů (G90/G91 příp. AC/IC):	
	• v případě G90, AC:	Udávání rozměrů v průměrech
	• v případě G91, IC:	Udávání rozměrů v průměrech
DIAM90A:	Příkaz pro aktivování **závislého** programování průměrů pro specifickou osu. Funkce příkazu DIAM90A závisí na naprogramovaném režimu zadávání rozměrů:	
	• v případě G90, AC:	Udávání rozměrů v průměrech
	• v případě G91, IC:	Zadávání rozměrů v rádiusech
DIAMOFA:	Příkaz pro deaktivování programování průměrů pro specifickou osu. Když je programování průměrů deaktivováno, je v platnosti programování rádiusů pro specifickou osu. Funkce příkazu DIAMOFA nezávisí na naprogramovaném režimu zadávání rozměrů:	
	• v případě G90, AC:	Zadávání rozměrů v rádiusech
	• v případě G91, IC:	Zadávání rozměrů v rádiusech
DIACYCOFA:	Příkaz pro deaktivování programování průměrů pro specifickou osu v průběhu zpracovávání cyklů. Takto je možné zajistit, aby se výpočty v cyklu uskutečňovaly pouze s rádiusy. Pro vypisování údajů o polohách a vypisování základního bloku zůstává aktivní naposled používaný příkaz G-funkce této skupiny.	
<osa>:	Identifikátor osy, pro kterou má být aktivováno osové programování průměrů. Přípustné identifikátory os jsou: • Název geometrické/kanálové osy nebo • Název osy stroje	
	Rozsah hodnot:	Uvedená osa musí být jednou z os v kanálu známých. Ostatní podmínky: • Pro osu musí být prostřednictvím strojního parametru MD30460 $MA_BASE_FUNCTION_MASK povoleno osové programování průměrů. • Kruhové osy jsou jako příčné osy nepřípustné.
Převzetí programování průměrů/rádiusů pro specifickou osu:		

DIAMCHANA:	Pomocí příkazu DIAMCHANA[<osa>] přebírá **uvedená osa** stav programování průměrů/rádiusů platný v daném kanálu a bude následovat změny nastavení programování průměrů/rádiusů pro specifický kanál.
DIAMCHAN:	Pomocí příkazu DIAMCHAN přebírají **všechny osy**, které jsou schváleny pro osové programování průměrů, stav programování průměrů/rádiusů platný v daném kanálu a bude následovat změny nastavení programování průměrů/rádiusů pro specifický kanál.
Programování průměrů/rádiusů s blokovou platností pro specifickou osu Programování průměrů/rádiusů s blokovou platnosti pro specifickou osu definuje způsob zadávání rozměrů ve výrobním programu a v synchronních akcích jako hodnotu průměru nebo rádiusu. Modální stav programování průměrů/rádiusů se nemění.	
DAC:	Pomocí příkazu DAC je pro zadanou osu v platnosti následující blokové zadávání rozměrů: Průměr v absolutních rozměrech
DIC:	Pomocí příkazu DIC je pro zadanou osu v platnosti následující blokové zadávání rozměrů: Průměr v inkrementálních rozměrech
RAC:	Pomocí příkazu RAC je pro zadanou osu v platnosti následující blokové zadávání rozměrů: Rádius v absolutních rozměrech
RIC:	Pomocí příkazu RIC je pro zadanou osu v platnosti následující blokové zadávání rozměrů: Rádius v inkrementálních rozměrech

Poznámka

Když je aktivní příkaz DIAMONA[<osa>] nebo DIAM90A[<osa>], vypisují se skutečné hodnoty pro příčnou osu vždy jako průměry. To platí také pro odečítání skutečné hodnoty v souřadném systému obrobku pomocí funkcí MEAS, MEAW, $P_EP[x] a $AA_IW[x].

Poznámka

V případě výměny os bude další příčná osa na základě příkazů GET a RELEASE[<osa>] přebírat stav programování průměrů/rádiusů v jiném kanálu.

Příklady

Příklad 1: Programování průměrů/rádiusů s modální platností pro specifickou osu

X je příčná osa v kanálu, pro Y je přípustné osové programování průměrů/rádiusů.

Programový kód	Komentář
N10 G0 X0 Z0 DIAMON	; programování průměrů pro osu X ve specifickém kanálu je aktivováno
N15 DIAMOF	; deaktivování kanálového programování průměrů
N20 DIAMONA[Y]	; osové programování průměrů s modální platností pro osu Y je aktivní
N25 X200 Y100	; programování rádiusů pro osu X je aktivováno

Programový kód	Komentář
N30 DIAMCHANA[Y]	; osa Y přebírá stav programování průměrů/rádiusů specifického kanálu a stává se podřízenou tomuto stavu
N35 X50 Y100	; programování rádiusů pro osy X a Y je aktivováno
N40 DIAMON	; aktivování programování průměrů v daném kanálu
N45 X50 Y100	; programování průměrů pro osy X a Y je aktivováno

Příklad 2: Programování průměrů/rádiusů s blokovou platností pro specifickou osu

X je příčná osa v kanálu, pro Y je přípustné osové programování průměrů/rádiusů.

Programový kód	Komentář
N10 DIAMON	; aktivování programování průměrů v daném kanálu
N15 G0 G90 X20 Y40 DIAMONA[Y]	; osové programování průměrů s modální platností pro osu Y je aktivní
N20 G01 X=RIC(5)	; platné zadávání rozměrů pro osu X v tomto bloku: Rádius v inkrementálních rozměrech.
N25 X=RAC(80)	; platné zadávání rozměrů pro osu X v tomto bloku: Rádius v absolutních rozměrech.
N30 WHEN $SAA_IM[Y]> 50 DO POS[X]=RIC(1)	; X je řídící osou. Pro tento blok platí následující zadávání rozměrů pro osu X: Rádius v inkrementálních rozměrech.
N40 WHEN $SAA_IM[Y]> 60 DO POS[X]=DAC(10)	; X je řídící osou. Pro tento blok platí následující zadávání rozměrů pro osu X: Rádius v absolutních rozměrech.
N50 G4 F3	

Další informace

Hodnoty průměru (DIAMONA/DIAM90A)

Hodnoty průměru se vztahují na následující údaje:

- Výpis skutečné hodnoty příčné osy v souřadném systému obrobku
- Režim JOG: Inkrementy pro krokový posuv a manuální posuv ručním kolečkem
- Programování koncových pozic:
  Interpolační parametry `I`, `J`, `K` u příkazů `G2`/`G3`, jestliže byly tyto příkazy pomocí `AC` naprogramovány s absolutními souřadnicemi.
  V případě programování inkrementálních hodnot `IC` parametrů `I`, `J`, `K` jsou tyto údaje vždy započítávány jako rádius.
- Načítání skutečných hodnot v souřadném systému obrobku při použití příkazů:
  `MEAS`, `MEAW`, `$P_EP[X]`, `$AA_IW[X]`

Programování průměrů pro specifickou osu s blokovou platností (DAC, DIC, RAC, RIC)

Příkazy DAC, DIC, RAC a RIC jsou přípustné pro všechny příkazy, pro které je zohledňováno programování průměrů ve specifickém kanálu:

- Poloha osy: X..., POS, POSA
- Oscilace: OSP1, OSP2, OSS, OSE, POSP
- Interpolační parametry: I, J, K
- Definice kontury: Přímka se zadáním úhlu
- Rychlé pozvednutí: POLF[AX]
- Posuv ve směru nástroje: MOVT
- Měkké najíždění a odjíždění:
  G140 až G143, G147, G148, G247, G248, G347, G348, G340 a G341

3.8.5 Poloha obrobku při soustružení

Identifikátory os

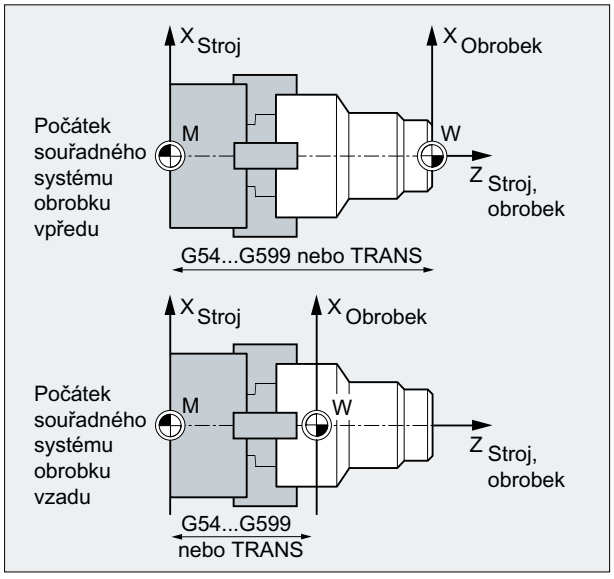
Obě geometrické osy, které jsou na sebe kolmé, jsou obvykle označovány následujícím způsobem:

Podélná osa	= osa Z (abscisa)
Příčná osa	= osa X (ordináta)

Počátek souřadného systému obrobku

Zatímco počátek souřadného systému stroje je pevný, polohu počátku souřadného systému obrobku na podélné ose si můžete libovolně zvolit. Obecně se počátek souřadného systému obrobku nachází na jeho přední nebo zadní straně.

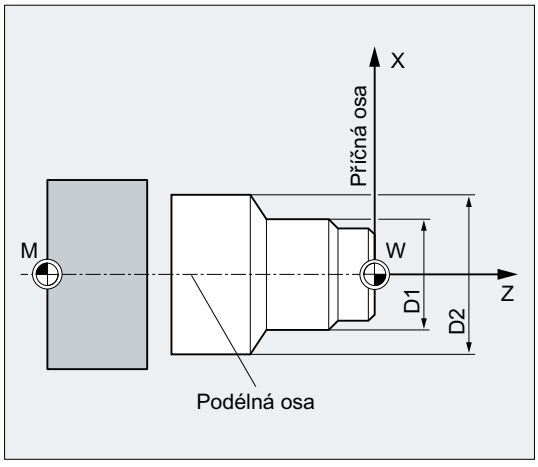
Jak počátek souřadné soustavy stroje, tak i obrobku, leží na ose otáčení. Nastavitelné posunutí ve směru osy X je proto nulové.



M	Počátek souřadného systému stroje
W	Počátek souřadného systému obrobku
`Z`	Podélná osa
`X`	Příčná osa
`G54` až `G599` nebo `TRANS`	Volání pro polohu počátku souřadné soustavy obrobku

Příčná osa

Rozměry pro příčnou osu jsou obecně specifikovány jsou údaje průměru (dvojnásobek délky dráhy ve srovnání s ostatními osami).



To, která geometrická osa slouží jako příčná osa, je nutno definovat ve strojních parametrech (--> výrobce stroje!).

3.9 Příkazy dráhy

3.9.1 Všeobecné informace týkající se příkazů dráhy

Konturové prvky

Naprogramovaná kontura obrobku se může skládat z následujících konturových prvků:

- přímky
- kruhové oblouky
- spirální dráhy (pomocí superpozice přímek a kruhových oblouků)

Příkazy posuvu

Pro výrobu těchto konturových prvků jsou k dispozici různé příkazy posuvu:

- Pohyb rychlým posuvem (`G0`)
- Přímková interpolace (`G1`)
- Kruhová interpolace ve směru hodinových ručiček (`G2`)
- Kruhová interpolace proti směru hodinových ručiček (`G3`)

Příkazy posuvu mají modální platnost.

Konečné pozice

Pohybový blok obsahuje cílové pozice pro osy, které se mají pohybovat (dráhové osy, synchronní osy, polohovací osy).

Programování cílových pozic se může uskutečňovat v kartézských souřadnicích nebo v polárních souřadnicích.

Poznámka

Jedna adresa osy smí být v bloku naprogramována jen jednou.

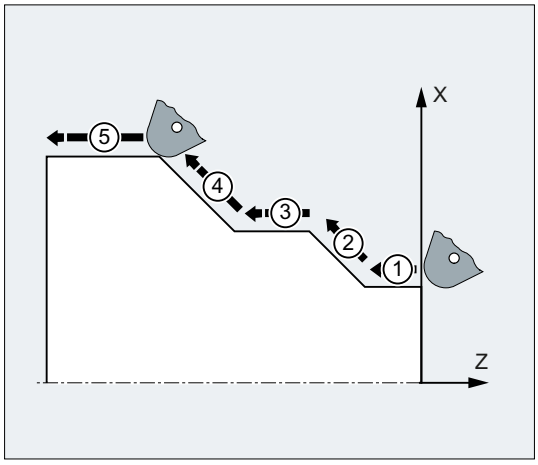
Počáteční bod – koncový bod

Pohyby po dráze začínají vždy z pozice, na kterou se naposled najelo, a končí v naprogramované cílové pozici. Tato cílová pozice je opět počáteční pozicí pro následující příkaz dráhy.

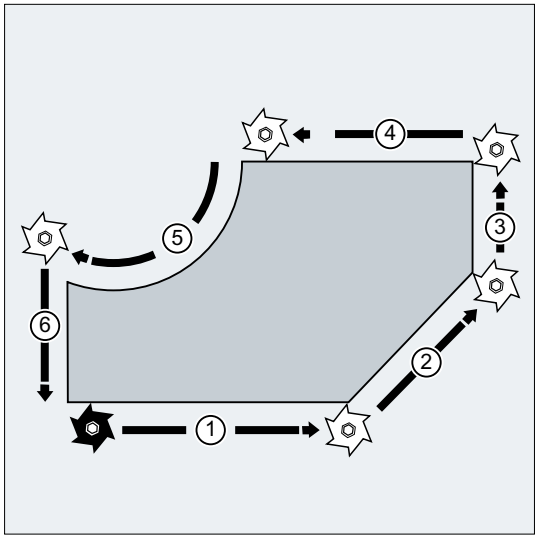
Kontura obrobku

UPOZORNĚNÍ
Záběr nástroje není definován Před zahájením procesu obrábění musí být nástroj nastaven do takové pozice, aby při spuštění opracování bylo poškození nástroje nebo obrobku vyloučeno.

Tyto pohybové bloky jsou prováděny jeden po druhém a tvoří konturu obrobku.



Obrázek 3-1 Pohybové bloky při soustružení



Obrázek 3-2 Pohybové bloky při frézování

3.9.2 Příkazy posuvu s kartézskými souřadnicemi (G0, G1, G2, G3, X..., Y..., Z...)

Na pozici zadanou v NC bloku pomocí kartézských souřadnic je možno najet rychlým posuvem `G0`, pomocí přímkové interpolace `G1` nebo pomocí kruhové interpolace `G2` /`G3`.

Syntaxe

```
G0 X... Y... Z...
G1 X... Y... Z...
G2 X... Y... Z... ...
G3 X... Y... Z... ...
```

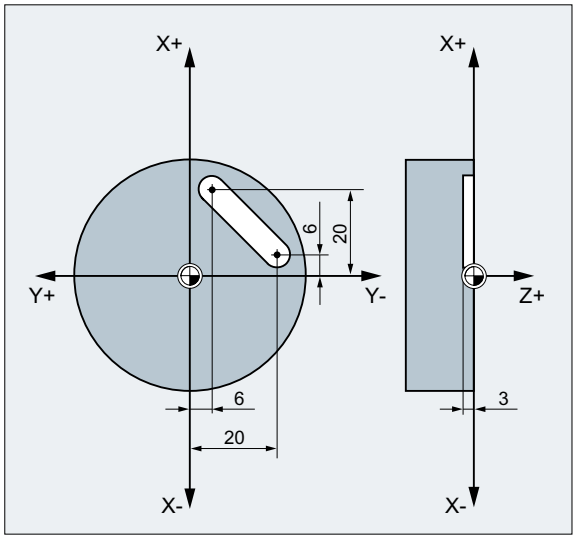
Význam

G0:	Příkaz pro aktivování pohybu rychlým posuvem
G1:	Příkaz pro aktivování přímkové interpolace
G2:	Příkaz pro aktivování kruhové interpolace ve směru hodinových ručiček
G3:	Příkaz pro aktivování kruhové interpolace proti směru hodinových ručiček
X...:	Kartézská souřadnice cílové pozice ve směru osy X
Y...:	Kartézská souřadnice cílové pozice ve směru osy Y
Z...:	Kartézská souřadnice cílové pozice ve směru osy Z

Poznámka

Kruhová interpolace G2 / G3 potřebuje kromě souřadnic cílové pozice X..., Y..., Z... ještě i další údaje (např. souřadnice středu kruhu, viz "Přehled (Strana 190)").

Příklad:



Programový kód	Komentář
N10 G17 S400 M3	; volba pracovní roviny, vřeteno se otáčí vpravo
N20 G0 X40 Y-6 Z2	; najíždění rychlým posuvem na počáteční pozici zadanou v kartézských souřadnicích
N30 G1 Z-3 F40	; aktivování přímkové interpolace, přísuv nástroje
N40 X12 Y-20	; najíždění po šikmo ležící přímce na koncovou pozici zadanou v kartézských souřadnicích

Programový kód	Komentář
N50 G0 Z100 M30	; volný pohyb rychlým posuvem za účelem výměny nástroje

3.9.3 Příkazy posuvu s polárními souřadnicemi

3.9.3.1 Vztažný bod polárních souřadnic (G110, G111, G112)

Bod, od něhož kótování vychází, se nazývá pól.

Pro zadání pólu je možné používat buď kartézské nebo polární souřadnice.

Prostřednictvím příkazů G110 až G112 je vztažný bod pro polární souřadnice jednoznačně definován. Zadávání absolutních nebo inkrementálních rozměrů nemá proto žádný vliv.

Syntaxe

```
G110/G111/G112 X... Y... Z...
G110/G111/G112 AP=... RP=...
```

Význam

G110 ...:	Pomocí příkazu G110 jsou následující souřadnice pólu vztaženy **na pozici, na kterou se naposled najelo**.		
G111 ...:	Pomocí příkazu G111 jsou následující souřadnice pólu vztaženy **na počátek aktuálního souřadného systému obrobku**.		
G112 ...:	Pomocí příkazu G112 jsou následující souřadnice pólu vztaženy **na poslední platný pól**.		
	Upozornění: Příkazy G110...G112 musí být naprogramovány v samostatném NC bloku.		
X... Y... Z...:	Zadání polohy pólu v kartézských souřadnicích		
AP=... RP=...:	Zadání polohy pólu v polárních souřadnicích		
	AP=...:	Polární úhel Úhel mezi polárním rádiusem a vodorovnou osou pracovní roviny (např. v případě roviny G17 je to osa X). Za kladný je považován směr proti směru hodinových ručiček.	
		Rozsah hodnot:	± 0...360°
	RP=...:	Polární rádius Údaj se vždy zadává jako **absolutní kladná hodnota** v [mm] nebo v [palcích].	

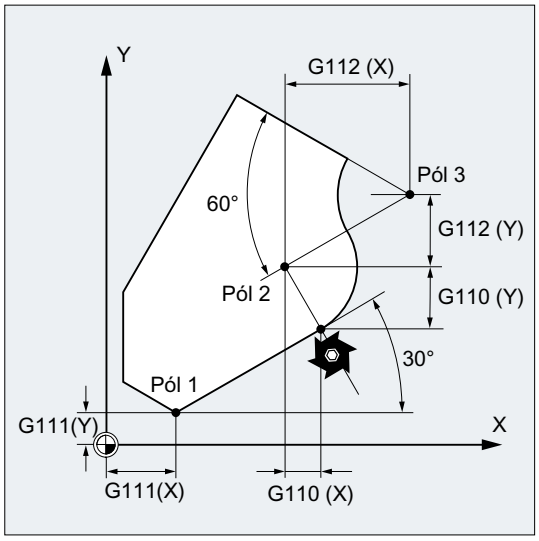
Poznámka

V NC programu je možné blokově přepínat mezi zadáváním polárních a kartézských rozměrů. Jestliže použijete identifikátory kartézských souřadných os (X..., Y..., Z...), znovu se vrátíte přímo do kartézského souřadného systému. Definovaný pól kromě toho zůstává zachován až do konce programu.

Poznámka

Jestliže nebyl udán žádný pól, použije se počátek aktuálního souřadného systému obrobku.

Příklad:

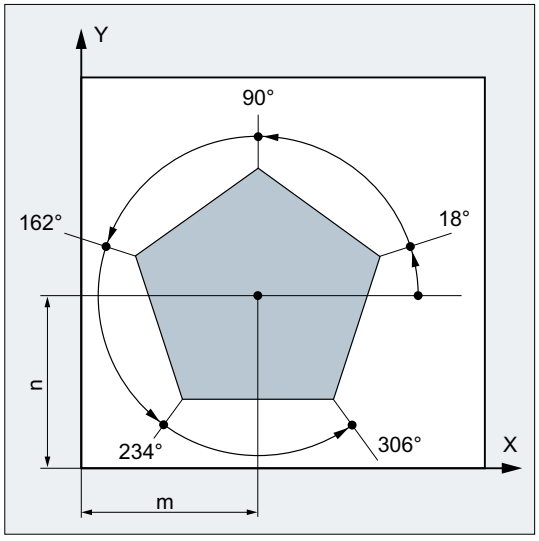


Póly 1 až 3 jsou definovány následujícím způsobem:

- Pól 1 pomocí `G111 X... Y...`
- Pól 2 pomocí `G110 X... Y...`
- Pól 3 pomocí `G112 X... Y...`

3.9.3.2 Příkazy posuvů pomocí polárních souřadnic (G0, G1, G2, G3, AP, RP)

Příkazy pohybu v polárních souřadnicích mají smysl tehdy, pokud jsou rozměrové údaje obrobku nebo jeho součásti vztaženy na jeden centrální bod a pokud jsou udávány rozměry v úhlech a v rádiusech (např. v případě vrtacích vzorů).



Syntaxe

```
G0/G1/G2/G3 AP=... RP=...
```

Význam

G0:	Příkaz pro aktivování pohybu rychlým posuvem	
G1:	Příkaz pro aktivování přímkové interpolace	
G2:	Příkaz pro aktivování kruhové interpolace ve směru hodinových ručiček	
G3:	Příkaz pro aktivování kruhové interpolace proti směru hodinových ručiček	
AP:	Polární úhel Úhel mezi polárním rádiusem a vodorovnou osou pracovní roviny (např. v případě roviny G17 je to osa X). Za kladný je považován směr proti směru hodinových ručiček.	
	Rozsah hodnot:	± 0...360°
	Údaj úhlu může být zadán jako absolutní i jako inkrementální hodnota:	
	AP=AC(...):	Zadávání absolutních rozměrů
	AP=IC(...):	Zadávání inkrementálních rozměrů V případě inkrementálních rozměrů (řetězové kóty) platí jako vztažný naposled naprogramovaný úhel.
	Polární úhel zůstává uložený tak dlouho, dokud není definován nový pól nebo dokud se nezmění pracovní rovina.	
RP:	Polární rádius Údaj se vždy zadává jako **absolutní kladná hodnota** v [mm] nebo v [palcích]. Polární rádius zůstává uložen až do zadání nové hodnoty.	

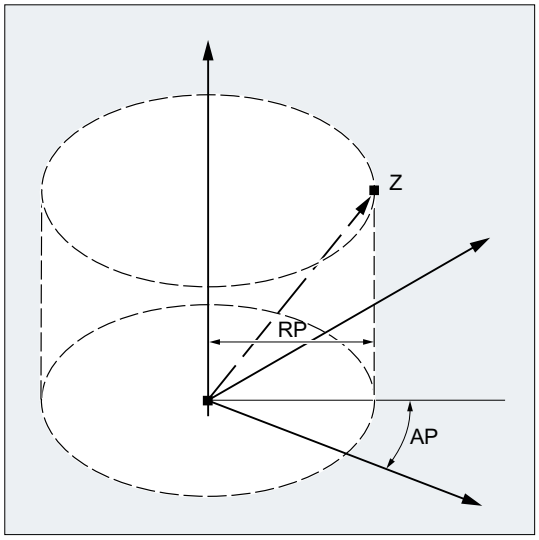
Poznámka

Polární souřadnice se vztahují na pól definovaný pomocí příkazů G110 ... G112 a platí v pracovní rovině stanovené příkazy G17 až G19.

Poznámka

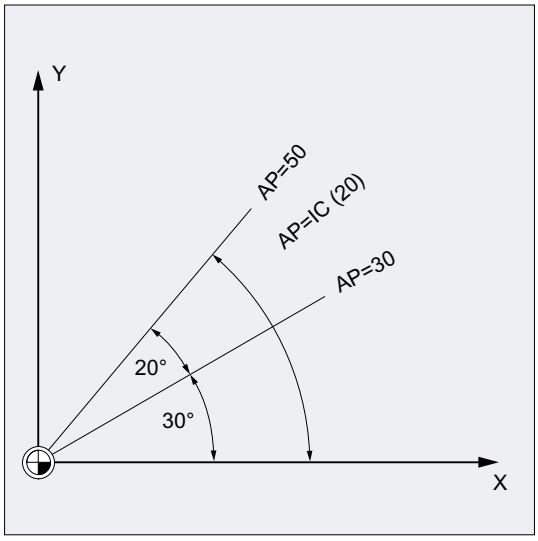
3. geometrická osa ležící kolmo na pracovní rovinu může být zadána navíc jako kartézská souřadnice (viz následující obrázek). Tímto způsobem můžete naprogramovat prostorové polohy ve válcových souřadnicích.

Příklad: `G17 G0 AP... RP... Z...`



Okrajové podmínky

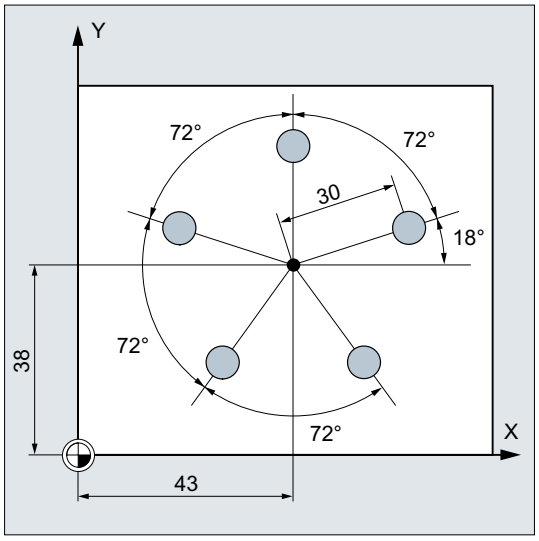
- V NC blocích s polárním zadáním koncového bodu nesmí být pro zvolenou pracovní rovinu naprogramovány žádné další kartézské souřadnice, jako jsou interpolační parametry, adresy os atd.
- Jestliže není pomocí příkazů `G110` ... `G112` definován žádný pól, bude za pól automaticky považován počátek právě platného souřadného systému obrobku.



- Polární rádius RP = 0
  Polární rádius se vypočítává ze vzdálenosti mezi vektorem počátečního bodu v rovině pólu a aktivním vektorem pólu. Potom se vypočítaný polární rádius modálně uloží.
  Tato zásada platí nezávisle na zvolené definici pólu (`G110` ... `G112`). Pokud jsou oba body naprogramovány jako identické, bude mít tento rádius nulovou hodnotu a aktivuje se alarm 14095.
- Je naprogramován pouze polární úhel AP
  Pokud se v aktuálním bloku nenalézá žádný polární rádius RP, je ale naprogramován polární úhel AP, potom pokud je nějaký rozdíl mezi aktuální pozicí a pólem v souřadném systému obrobku, použije se tento rozdíl jako polární rádius a modálně se uloží. Pokud je tento rozdíl roven nule, jsou souřadnice pólu specifikovány znovu a modální polární rádius zůstane nulový.

Příklad:

Výroba vrtacího vzoru



Polohy vrtaných děr jsou zadány v polárních souřadnicích.

Každá vrtaná díra je vyráběna stejným výrobním postupem:

Předvrtání, vrtání na daný rozměr, vystružování ...

Postup obrábění je naprogramován v podprogramu.

Programový kód	**Komentář**
`N10 G17 G54`	`; pracování rovina X/Y, počátek souřadného systému obrobku`
`N20 G111 X43 Y38`	`; definice pólu`
`N30 G0 RP=30 AP=18 Z5`	`; najíždění na počáteční bod, zadání ve válcových souřadnicích`
`N40 L10`	`; volání podprogramu`
`N50 G91 AP=72`	`; najíždění na následující pozici rychlým posuvem, polární úhel v inkrementálních souřadnicích, polární rádius zůstává uložen z bloku N30 a nemusí být zadáván znovu`
`N60 L10`	`; volání podprogramu`
`N70 AP=IC(72)`	`.`
`N80 L10`	…

Programový kód	**Komentář**
N90 AP=IC(72)	
N100 L10	…
N110 AP=IC(72)	
N120 L10	…
N130 G0 X300 Y200 Z100 M30	; vyjíždění nástroje, konec programu

Viz také

Přehled (Strana 190)

3.9.4 Pohyby rychlým posuvem

3.9.4.1 Aktivování rychlého posuvu (G0)

Pohyby dráhových os rychlým posuvem se spouštějí příkazem G-funkce G0.

Syntaxe

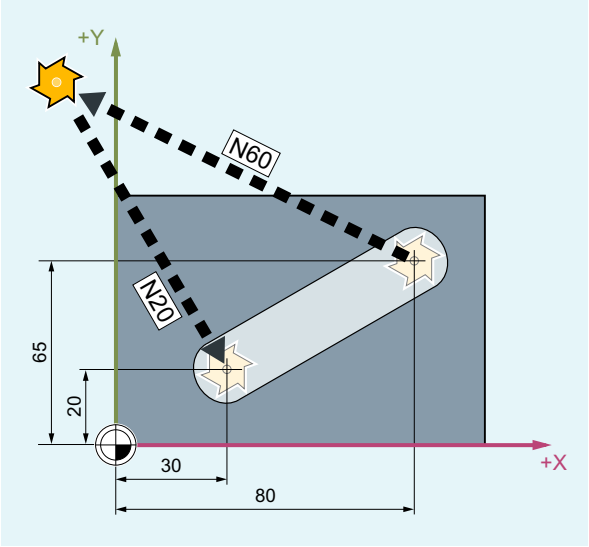
```
G0 X… Y… Z…
G0 RP=… AP=…
```

Význam

G0:	Pohyb osy rychlostí rychlého posuvu	
	Platnost:	modální
X... Y... Z...:	Zadání koncového bodu v kartézských souřadnicích	
RP=... AP=... :	Zadání koncového bodu v polárních souřadnicích	

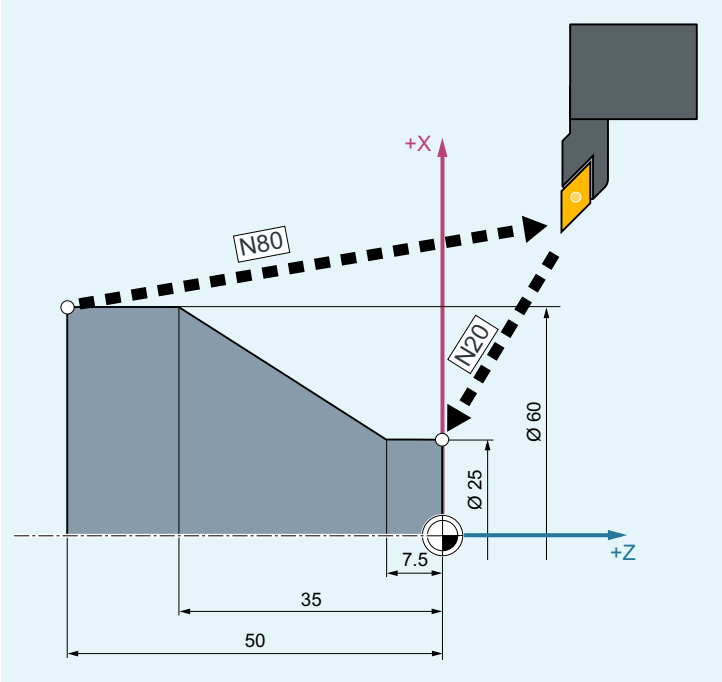
Příklady

Příklad 1: Frézování



Programový kód	Komentář
N10 G90 S400 M3	; zadávání absolutních rozměrů, vřeteno se otáčí vpravo
N20 G0 X30 Y20 Z2	; najíždění na počáteční pozici
N30 G1 Z-5 F1000	; přísuv nástroje
N40 X80 Y65	; pohyb po přímkách
N50 G0 Z2	
N60 G0 X-20 Y100 Z100 M30	; vyjíždění nástroje, konec programu

Příklad 2: Soustružení



Programový kód	Komentář
N10 G90 S400 M3	; zadávání absolutních rozměrů, vřeteno se otáčí vpravo
N20 G0 X25 Z5	; najíždění na počáteční pozici
N30 G1 G94 Z0 F1000	; přísuv nástroje
N40 G95 Z-7.5 F0.2	
N50 X60 Z-35	; pohyb po přímkách
N60 Z-50	
N70 G0 X62	
N80 G0 X80 Z20 M30	; vyjíždění nástroje, konec programu

3.9.4.2 Aktivování/deaktivování lineární interpolace pro pohyby rychlým posuvem (RTLION, RTLIOF)

Nezávisle na předem definovaném nastavení (MD20730 $MC_G0_LINEAR_MODE) může být chování interpolace při pohybech rychlým posuvem nastavováno i ve výrobním programu, a to pomocí příkazů ze skupiny G-funkcí č. 55.

Syntaxe

```
RTLIOF
...
RTLION
```

Význam

RTLIOF:	G-příkaz pro deaktivování lineární interpolace. ⇒ Při rychlém posuvu (G0) je aktivní **nelineární** interpolace. Všechny dráhové osy dosáhnou svého koncového bodu nezávisle na sobě.	
	Platnost:	modální
RTLION:	G-příkaz pro aktivování lineární interpolace. ⇒ Při rychlém posuvu (G0) je aktivní **lineární** interpolace. Všechny dráhové osy dosáhnou svého koncového bodu současně.	
	Platnost:	modální

Poznámka

Předpoklady pro zpracování funkce RTLIOF

Když je aktivní funkce RTLIOF a když se má uskutečňovat **nelineární** interpolace, musí být splněny následující předpoklady:

- Není aktivní žádná transformace (TRAORI, TRANSMIT, ...).
- Je aktivní příkaz G60 (zastavení na konci bloku).
- Kompresor není aktivní (COMPOF).
- Není aktivní žádná korekce rádiusu nástroje (G40).
- Nebyla aktivována žádná korekce ručním kolečkem.
- Není aktivní prostřihování.

Jestliže některý z těchto předpokladů není splněn, uskutečňuje se lineární interpolace jako s funkcí RTLION.

Příklad

```
Programový kód                  Komentář
                                ; Při standardním nastavení je aktivována lineár-
                                ní interpolace:
                                ; MD20730 $MC_GO_LINEAR_MODE == TRUE
...
N30 RTLIOF                      ; Lineární interpolace deaktivována.
N40 G0 X0 Y10                   ; Posuvy v blocích s příkazem G0 jsou realizovány
                                s nelineární interpolací.
N50 G41 X20 Y20                 ; Korekce rádiusu nástroje aktivní ⇒ posuvy v
                                blocích s příkazem G0 jsou realizovány s lineární
                                interpolací.
N60 G40 X30 Y30                 ; Korekce rádiusu nástroje neaktivní ⇒ posuvy v
                                blocích s příkazem G0 jsou realizovány s nelineár-
                                ní interpolací.
N70 RTLION                      ; Aktivování lineární interpolace
...
```

Další informace

Načtení aktuálního interpolačního chování

Aktuální interpolační chování je možno načíst pomocí systémové proměnné $AA_G0MODE.

3.9.4.3 Přizpůsobení tolerancí pro pohyby rychlým posuvem (STOLF, CTOLG0, OTOLG0)

Tolerance pro pohyby rychlým posuvem (tolerance G0), které jsou definovány ve strojních parametrech, mohou být ve výrobním programu dočasně změněny. Nastavení ve strojních parametrech se přitom nemění. Po resetu kanálu nebo po resetu na konci programu jsou znovu v platnosti tolerance nastavené v konfiguraci.

Předpoklady

Tolerance pohybů s příkazem G0 jsou v platnosti jen tehdy, pokud jsou splněny následující podmínky:

- Je aktivní některá z následujících funkcí:
  - Funkce kompresoru COMP...
  - Funkce pro přechodová zaoblení G642 nebo G645
  - Vyhlazování orientace pomocí funkce OST
  - Vyhlazování orientace pomocí příkazu ORISON
  - Vyhlazování orientace vztažené k dráze, ORIPATH
- Ve výrobním programu za sebou následuje větší počet (≥ 2) bloků s G0.
  V případě osamoceného bloku s G0 se tolerance posuvu s G0 neuplatňuje, protože při přechodu z pohybu, který není zadán pomocí G0, do pohybu s G0 (a obráceně) se v zásadě uplatňuje "**menší tolerance**" (tolerance pro opracovávání obrobku)!

Syntaxe

Přizpůsobení relativní hodnoty tolerance pro pohyby s G0

```
STOLF=<hodnota>
```

Přizpůsobení absolutní hodnoty tolerancí pro pohyby s příkazem G0

```
CTOLG0=<hodnota>
OTOLG0=<hodnota>
```

Význam

STOLF:	Adresa pro programování faktoru tolerance pro pohyby rychlým posuvem s dočasnou platností			
	<Value>:	Toleranční faktor G0		
		Typ:	REAL	
		Hodnota:	≥ 0:	Toleranční faktor G0 může být jak větší, tak také menší než 1,0. Jestliže je faktor roven 1,0 (standardní hodnota), platí pro pohyby rychlým posuvem stejné tolerance jako pro pohyby uskutečňované jinak než rychlým posuvem. Za normálních okolností se pro toleranční faktor nastavuje hodnota > 1,0. Naprogramovaný toleranční faktor G0 je v platnosti tak dlouho, dokud není přepsán naprogramováním nového příkazu STOLF, příp. dokud není deaktivován naprogramováním příkazu CTOLG0/OTOLG0 nebo resetem kanálu, příp. resetem na konci programu.
			< 0:	Vymazání naprogramovaného faktoru tolerance ⇒ Znovu platí toleranční faktor předem nastavený ve strojním parametru.
CTOLG0:	Adresa pro programování tolerance kontury pro pohyby rychlým posuvem s dočasnou platností			
	<hodnota>:	Absolutní hodnota tolerance kontury		
		Typ:	REAL	
		Hodnota:	≥ 0:	Naprogramovaná absolutní hodnota tolerance kontury je v platnosti tak dlouho, dokud není přepsána naprogramováním nového příkazu CTOLG0, příp. dokud není deaktivována naprogramováním příkazu STOLF nebo resetem kanálu, příp. resetem na konci programu.
			< 0:	Vymazání naprogramované hodnoty tolerance ⇒ Znovu platí hodnota tolerance nastavená předem ve strojním parametru.
OTOLG0:	Adresa pro programování tolerance orientace pro pohyby rychlým posuvem s dočasnou platností			
	<hodnota>:	Absolutní hodnota tolerance orientace		
		Typ:	REAL	
		Hodnota:	≥ 0:	Naprogramovaná absolutní hodnota tolerance orientace je v platnosti tak dlouho, dokud není přepsána naprogramováním nového příkazu OTOLG0, příp. dokud není deaktivována naprogramováním příkazu STOLF nebo resetem kanálu, příp. resetem na konci programu.
			< 0:	Vymazání naprogramované hodnoty tolerance ⇒ Znovu platí hodnota tolerance nastavená předem ve strojním parametru.

Poznámka

Přednost má vždy naposled naprogramovaná adresa, jak ukazují následující příklady:

- Když je aktivní příkaz STOLF a je naprogramován příkaz CTOLG0, bude se používat zaoblení kontury s hodnotou tolerance naprogramovanou v příkazu CTOLG0.
- Stejně tak když je aktivní příkaz STOLF a je naprogramován příkaz OTOLG0, bude se používat zaoblení orientace s hodnotou tolerance naprogramovanou v příkazu OTOLG0.
- Po novém naprogramování příkazu STOLF se bude opět používat faktor pro toleranci kontury a toleranci orientace.

Příklady

Příklad 1: Přizpůsobení relativní hodnoty tolerance pro pohyby s G0

```
Programový kód                  Komentář
COMPCAD G645 G1 F10000          ; funkce kompresoru COMPCAD
X... Y... Z...                  ; zde se uplatňují strojní a nastavované parametry
X... Y... Z...
X... Y... Z...
G0 X... Y... Z...
G0 X... Y... Z...               ; zde se uplatňuje strojní parametr $MC_G0_TOLERAN-
                                CE_FACTOR (např. =3) a také tolerance pro přechodová
                                zaoblení
                                $MC_G0_TOLERANCE_FACTOR * $MA_COMPRESS_POS_TOL
CTOL=0.02
STOLF=4
G1 X... Y... Z...               ; od tohoto místa se uplatňuje tolerance kontury 0,02 mm
X... Y... Z...
X... Y... Z...
G0 X... Y... Z...
X... Y... Z...                  ; od tohoto místa se uplatňuje toleranční faktor G0 rov-
                                nající se 4, tedy tolerance kontury rovnající se 0,08 mm
...
```

Příklad 2: Přizpůsobení absolutní hodnoty tolerancí pro pohyby s příkazem G0

Ve strojních parametrech mají být předem nastaveny, následující absolutní hodnoty tolerance pohybů s příkazem G0:

- Tolerance kontury při pohybech s příkazem G0: 0,1
- Tolerance orientace při pohybech s příkazem G0: 1.0

Ve výrobním programu mají být tyto tolerance dočasně přizpůsobeny:

```
Programový kód                  Komentář
COMPCAD G645 G1 F10000          ; funkce kompresoru COMPCAD
X... Y... Z...                  ; Od tohoto místa jsou v platnosti tolerance pro opra-
                                cování obrobku nastavené v konfiguraci.
```

Programový kód	**Komentář**
X... Y... Z...	
X... Y... Z...	
G0 X... Y... Z...	
G0 X... Y... Z...	; Zde jsou v platnosti absolutní tolerance G0 nastavené v konfiguraci.
CTOLG0=0.2 OTOLG0=2.0	; Naprogramování absolutní hodnoty tolerancí pro pohyby s příkazem G0.
G1 X... Y... Z...	
X... Y... Z...	
X... Y... Z...	
G0 X... Y... Z...	
X... Y... Z...	; Od tohoto místa jsou v platnosti naprogramované tolerance G0.
...	

Další informace

Načtení tolerančního faktoru pro pohyby s G0

Momentálně platný toleranční faktor je možné načíst pomocí systémových proměnných pro pohyby rychlým posuvem:

- V synchronních akcích nebo se zastavením předběžného zpracování ve výrobním programu pomocí systémové proměnné:

  $AC_STOLF — Aktivní toleranční faktor G0

  Toleranční faktor G0, který byl v platnosti při přípravě aktuálního bloku v hlavním zpracování.

- Bez zastavení předběžného zpracování ve výrobním programu pomocí systémové proměnné:

  $P_STOLF — Naprogramovaný toleranční faktor G0

Jestliže v aktivním výrobním programu není pro funkci STOLF naprogramována žádná hodnota, je v obou těchto systémových proměnných uložena hodnota nastavená ve strojním parametru.

Pokud v daném bloku není aktivní žádný rychlý posuv (G0), obsahují tyto systémové proměnné vždy hodnotu 1.

Načtení absolutní hodnoty tolerancí pro pohyby s příkazem G0

Momentálně platné absolutní hodnoty tolerancí pro pohyby rychlým posuvem je možno načíst pomocí systémových proměnných:

- V synchronních akcích nebo se zastavením předběžného zpracování ve výrobním programu pomocí systémových proměnných:

  $AC_CTOL_G0_ABS — Aktivní tolerance kontury při pohybech s příkazem G0

  Tolerance kontury při pohybech s příkazem G0, která byla v platnosti při přípravě aktuálního bloku v hlavním zpracování.

$AC_OTOL_ G0_ABS	Aktivní tolerance orientace při pohybech s příkazem G0 Tolerance orientace při pohybech s příkazem G0, která byla v platnosti při přípravě aktuálního bloku v hlavním zpracování.

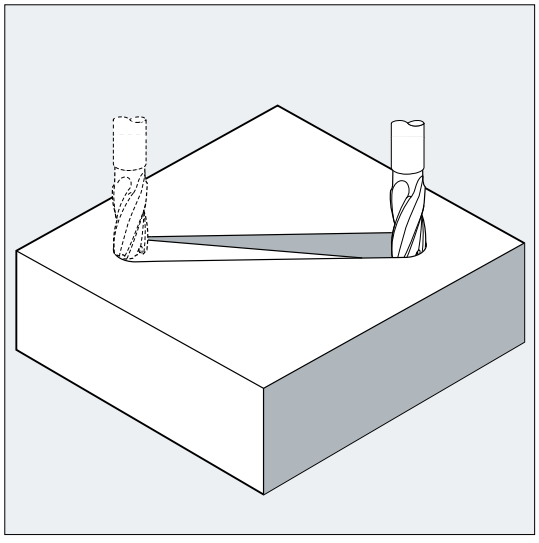
- Bez zastavení předběžného zpracování ve výrobním programu pomocí systémových proměnných:

$P_CTOL_ G0_ABS	Naprogramovaná tolerance kontury při pohybech s příkazem G0
$P_OTOL_ G0_ABS	Naprogramovaná tolerance orientace při pohybech s příkazem G0

Jestliže v aktivním výrobním programu nejsou pomocí příkazů CTOLG0 a OTOLG0 naprogramovány žádné absolutní hodnoty tolerancí pro pohyby s příkazem G0, potom jsou v těchto systémových proměnných uloženy hodnoty nastavené ve strojních parametrech.

3.9.5 Přímková interpolace (G1)

Pomocí funkce `G1` se nástroj pohybuje po přímkách rovnoběžných s osami, ležících šikmo nebo umístěných libovolně v prostoru. Přímková interpolace umožňuje výrobu 3D ploch, drážek atd.



Syntaxe

```
G1 X... Y... Z ... F...
G1 AP=... RP=... F...
```

Význam

`G1`:	Přímková interpolace (lineární interpolace s pracovním posuvem)
`X... Y... Z...`:	Koncový bod v kartézských souřadnicích
`AP=...`:	Koncový bod v polárních souřadnicích, zde polární úhel

`RP=...:`	Koncový bod v polárních souřadnicích, zde polární rádius
`F...:`	Rychlost pracovního posuvu v mm/min. Nástroj se pohybuje posuvem F po přímce z momentálního počátečního bodu do naprogramovaného cílového bodu. Cílový bod zadáváte v kartézských nebo v polárních souřadnicích. Na této dráze nástroj provádí obrábění. Příklad: `G1 G94 X100 Y20 Z30 A40 F100` Na koncový bod X, Y, Z se bude najíždět s posuvem 100 mm/min. Kruhová osa A se bude jako synchronizovaná osa pohybovat tak, aby všechny čtyři pohyby byly ukončeny ve stejném časovém okamžiku.

Poznámka

Příkaz `G1` má modální působnost.

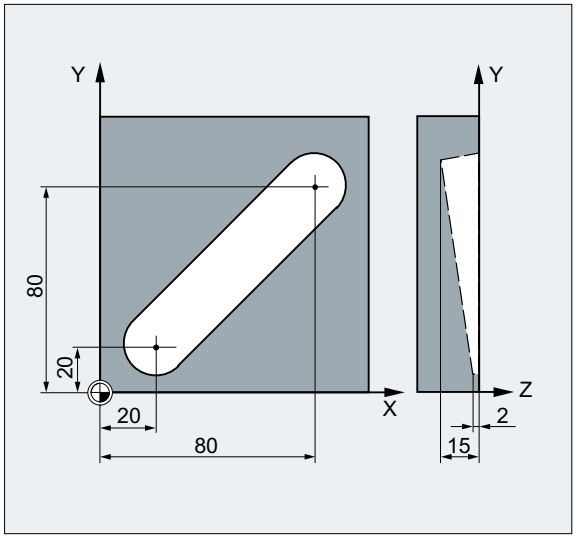
Za účelem opracování musí být zadány otáčky vřetena `S` a směr otáčení vřetena `M3`/`M4`.

Pomocí příkazu `FGROUP` mohou být definovány skupiny os, pro které platí dráhový posuv `F`. Pokud budete potřebovat další informace, viz "Chování při pohybu po dráze".

Příklady

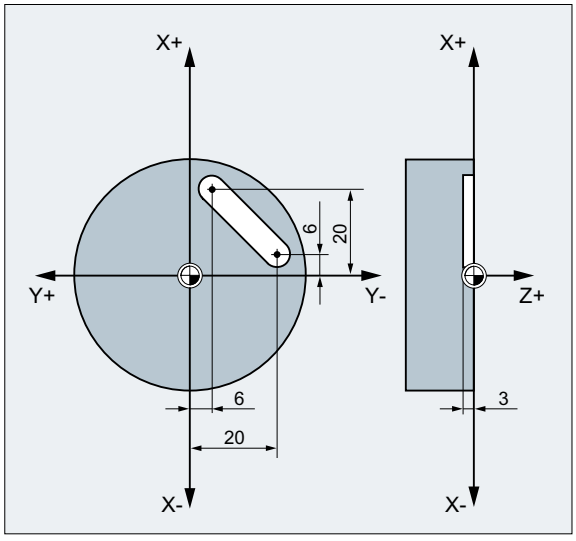
Příklad 1: Výroba drážky (frézování)

Nástroj se pohybuje z počátečního bodu do koncového bodu ve směru X/Y. Současně se provádí přísuv v ose Z.



Programový kód	**Komentář**
`N10 G17 S400 M3`	`; volba pracovní roviny, vřeteno se otáčí vpravo`
`N20 G0 X20 Y20 Z2`	`; najíždění na počáteční pozici`
`N30 G1 Z-2 F40`	`; přísuv nástroje`
`N40 X80 Y80 Z-15`	`; posuv po šikmo ležící přímce`
`N50 G0 Z100 M30`	`; volné najíždění na bod pro výměnu nástroje`

Příklad 2: Výroba drážky (soustružení)

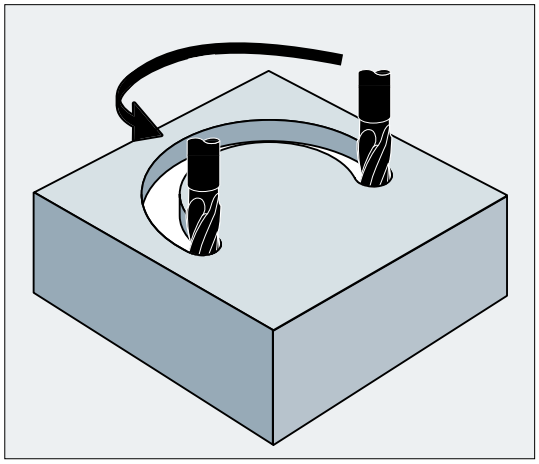


Programový kód	Komentář
N10 G17 S400 M3	; volba pracovní roviny, vřeteno se otáčí vpravo
N20 G0 X40 Y-6 Z2	; najíždění na počáteční pozici
N30 G1 Z-3 F40	; přísuv nástroje
N40 X12 Y-20	; posuv po šikmo ležící přímce
N50 G0 Z100 M30	; volné najíždění na bod pro výměnu nástroje

3.9.6 Kruhová interpolace

3.9.6.1 Přehled

Kruhová interpolace umožňuje výrobu celých kružnic nebo kruhových oblouků.



Obrázek 3-3 Příklad použití: Frézování kruhové drážky

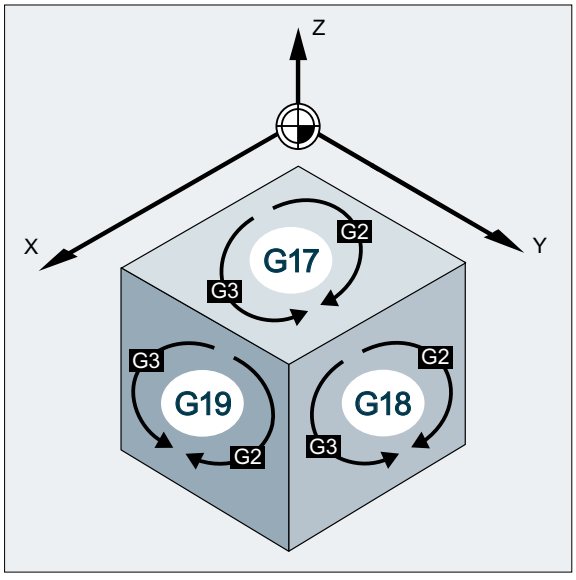
Varianta programování

Řídící systém nabízí různé možnosti, jak programovat kruhové pohyby. Jejich prostřednictvím může uživatel přímo do programu převést prakticky jakýkoli druh kótování z výkresu.

- Kruhová interpolace se středem a koncovým bodem (G2/G3, X... Y... Z..., I... J... K...) (Strana 191)
- Kruhová interpolace s rádiusem a koncovým bodem (G2/G3, X... Y... Z..., CR) (Strana 194)
- Kruhová interpolace s úhlem kruhové výseče a koncovým bodem/středem (G2/G3, X... Y... Z... / I... J... K..., AR) (Strana 196)
- Kruhová interpolace pomocí polárních souřadnic (G2/G3, AP, RP) (Strana 198)
- Kruhová interpolace s vnitřním a koncovým bodem (CIP, X... Y... Z..., I1... J1... K1...) (Strana 199)
- Kruhová interpolace s tangenciálním přechodem (CT, X... Y... Z...) (Strana 202)

Rovina pro kruhovou interpolaci

Pro výpočet směru opisování kružnice – `G2` ve směru nebo `G3` proti směru hodinových ručiček – potřebuje řídící systém zadání pracovní roviny (Strana 147).



Výjimka:

Kruhové útvary je také možné vyrábět i mimo zvolenou pracovní rovinu (nikoli při zadání úhlu výseče). V tomto případě určují rovinu kruhu adresy os, které programátor zadal jako koncový bod kruhu.

3.9.6.2 Kruhová interpolace se středem a koncovým bodem (G2/G3, X... Y... Z..., I... J... K...)

Varianta kruhové interpolace, při které se pro interpolaci používají **střed** a **koncový bod** prvku kontury kruhového tvaru.

Pokud je naprogramován kruh bez koncového bodu, vznikne celá kružnice.

Syntaxe

```
G2/G3 X... Y... Z... I... J... K...
G2/G3 X... Y... Z... I=AC(...) J=AC(...) K=(AC...)
```

Význam

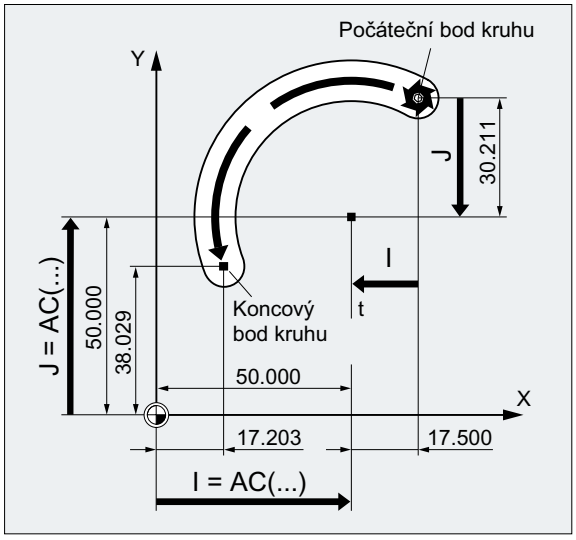
G2:	Kruhová interpolace ve směru hodinových ručiček	
	Platnost:	modální
G3:	Kruhová interpolace proti směru hodinových ručiček	
	Platnost:	modální
X... Y... Z...:	Koncový bod kruhu v kartézských souřadnicích V závislosti na momentálně platném nastavení zadávání rozměrů G90/G91, příp. ...=AC(...) / ...=IC(...), jsou souřadnice koncového bodu kruhu interpretovány buď v absolutních nebo v inkrementálních rozměrech.	
I... J... K...:	Interpolační parametry pro zadávání souřadnic středu kruhu ve směrech X, Y, Z Souřadnice středu kruhu se standardně zadávají v inkrementálních rozměrech vzhledem k počátečnímu bodu kruhu. Jestliže mají být souřadnice středu kruhu v absolutních rozměrech vztaženy na počátek souřadného systému obrobku, je potřeba interpolační parametry I, J, K naprogramovat následujícím způsobem: I=AC(...) J=AC(...) K=AC(...) **Upozornění:** Jeden interpolační parametr s hodnotou 0 může být vypuštěn, druhý související parametr však musí být v každém případě zadán.	

Poznámka

Předvolba G90/G91 (absolutní nebo inkrementální rozměry) je platná pouze pro koncový bod kruhu.

Příklady

Příklad 1: Frézování



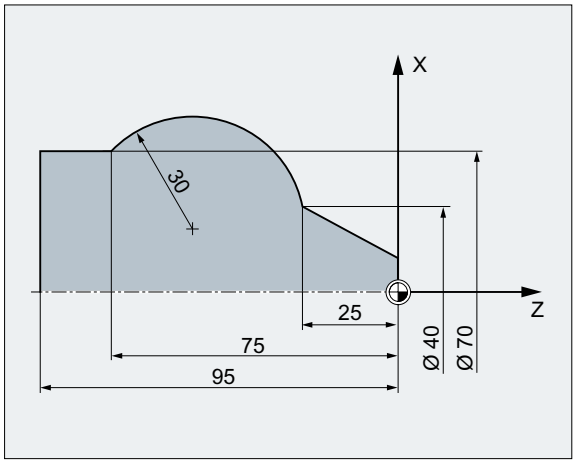
Zadání středu v inkrementálních rozměrech

```
N10 G0 X67.5 Y80.211
N20 G3 X17.203 Y38.029 I-17.5 J-30.211 F500
```

Zadání středu v absolutních rozměrech

```
N10 G0 X67.5 Y80.211
N20 G3 X17.203 Y38.029 I=AC(50) J=AC(50)
```

Příklad 2: Soustružení



Zadání středu v inkrementálních rozměrech

```
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
```

Zadání středu v absolutních rozměrech

```
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
```

3.9.6.3 Kruhová interpolace s rádiusem a koncovým bodem (G2/G3, X... Y... Z..., CR)

Varianta kruhové interpolace, při které se pro interpolaci používají **rádius** a **koncový bod** prvku kontury kruhového tvaru.

Poznámka

Úplnou kružnici (úhel kruhové dráhy 360°) **není možné** pomocí této varianty naprogramovat.

Syntaxe

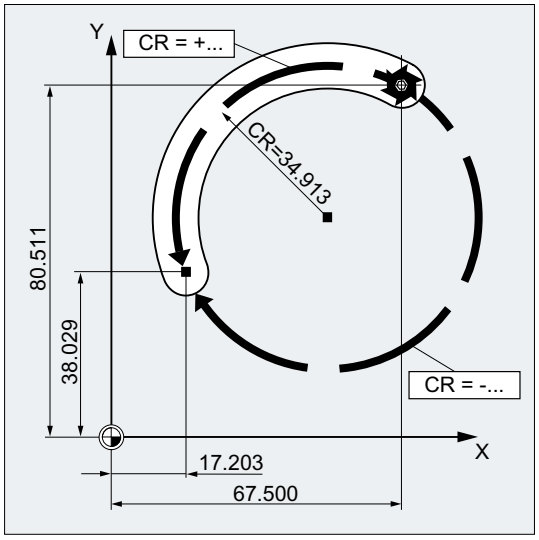
```
G2/G3 X… Y… Z… CR=±...
```

Význam

G2:	Kruhová interpolace ve směru hodinových ručiček	
	Platnost:	modální
G3:	Kruhová interpolace proti směru hodinových ručiček	
	Platnost:	modální
X... Y... Z...:	Koncový bod kruhu v kartézských souřadnicích V závislosti na momentálně platném nastavení zadávání rozměrů G90/G91, příp. ...=AC(...) / ...=IC(...), jsou souřadnice koncového bodu interpretovány buď v absolutních nebo v inkrementálních rozměrech.	
CR=±...:	Rádius kruhu Prostřednictvím znaménka se udává, zda opisovaný úhel má být větší nebo menší než 180°. Kladné znaménko je možné vypustit.	
	CR=+...:	Úhel kruhové dráhy ≤ 180°
	CR=-...:	Úhel kruhové dráhy > 180°
	Upozornění Neexistuje žádné praktické omezení pro velikost maximálního naprogramovatelného rádiusu.	

Příklady

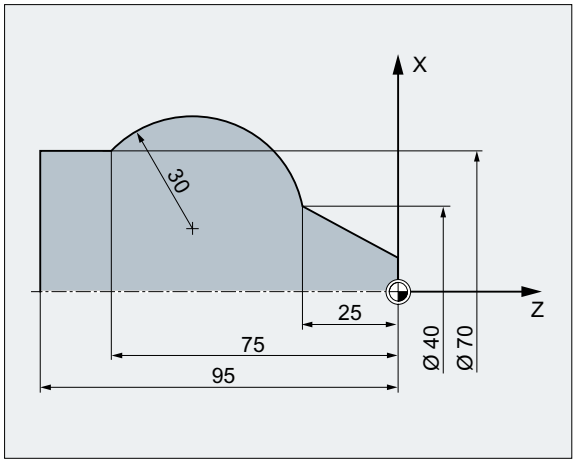
Příklad 1: Frézování



Programový kód

```
N10 G0 X67.5 Y80.511
N20 G3 X17.203 Y38.029 CR=34.913 F500
...
```

Příklad 2: Soustružení



Programový kód

```
...
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 CR=30
N135 G1 Z-95
...
```

3.9.6.4 Kruhová interpolace s úhlem kruhové výseče a koncovým bodem/středem (G2/G3, X... Y... Z... / I... J... K..., AR)

Varianta kruhové interpolace, při které se pro interpolaci používají **úhel kruhové výseče** a **střed** nebo **koncový bod** prvku kontury kruhového tvaru.

Poznámka

Úplnou kružnici (úhel kruhové dráhy 360°) **není možné** pomocí této varianty naprogramovat.

Syntaxe

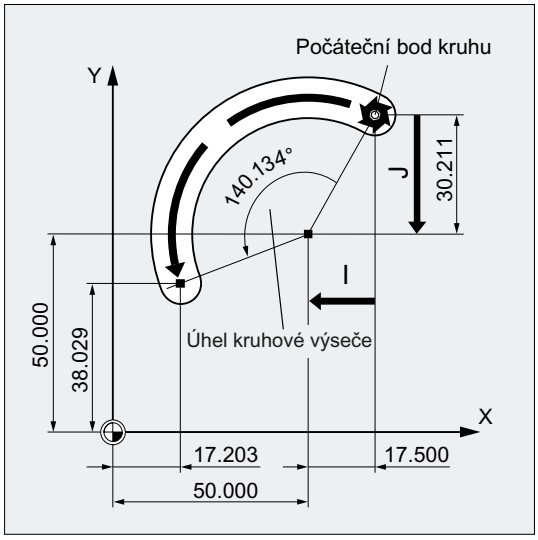
```
G2/G3 X… Y… Z… AR=...
G2/G3 I… J… K… AR=...
```

Význam

G2:	Kruhová interpolace ve směru hodinových ručiček	
	Platnost:	modální
G3:	Kruhová interpolace proti směru hodinových ručiček	
	Platnost:	modální
X... Y... Z...:	Koncový bod kruhu v kartézských souřadnicích V závislosti na momentálně platném nastavení zadávání rozměrů G90/G91, příp. ...=AC(...) / ...=IC(...), jsou souřadnice koncového bodu kruhu interpretovány buď v absolutních nebo v inkrementálních rozměrech.	
I... J... K...:	Interpolační parametry pro zadávání souřadnic středu kruhu ve směrech X, Y, Z Souřadnice středu kruhu se standardně zadávají v inkrementálních rozměrech vzhledem k počátečnímu bodu kruhu. Jestliže mají být souřadnice středu kruhu v absolutních rozměrech vztaženy na počátek souřadného systému obrobku, je potřeba interpolační parametry I, J, K naprogramovat následujícím způsobem: I=AC(...) J=AC(...) K=AC(...) **Upozornění:** Jeden interpolační parametr s hodnotou 0 může být vypuštěn, druhý související parametr však musí být v každém případě zadán.	
AR=...:	Úhel kruhové výseče	
	Rozsah hodnot:	0° ... 360°

Příklady

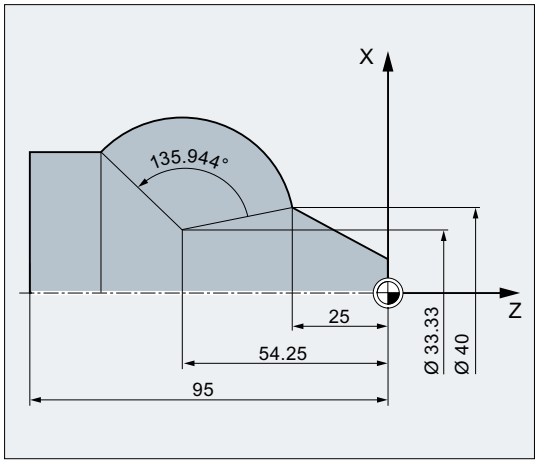
Příklad 1: Frézování



Programový kód

```
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
```

Příklad 2: Soustružení



Programový kód

```
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
```

Programový kód
`N135 G1 Z-95`

3.9.6.5 Kruhová interpolace pomocí polárních souřadnic (G2/G3, AP, RP)

Varianta kruhové interpolace, při které se pro interpolaci používá **koncový bod kruhu v polárních souřadnicích**.

Přitom platí následující konvence:

- Pól se nachází ve středu kruhu.
- Polární rádius odpovídá rádiusu kruhu.

Syntaxe

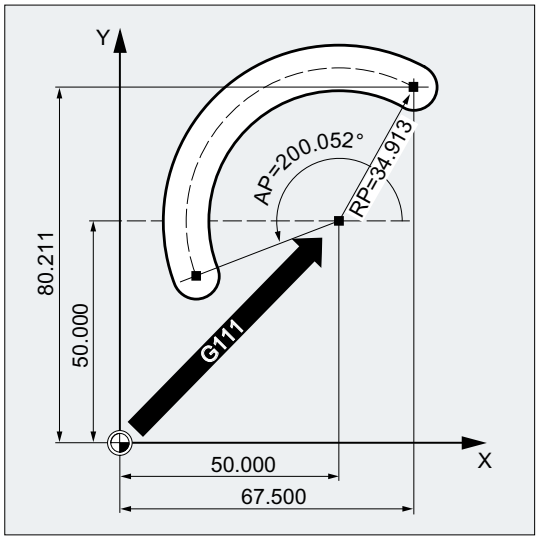
```
G2/G3 AP=... RP=...
```

Význam

`G2`:	Kruhová interpolace ve směru hodinových ručiček	
	Platnost:	modální
`G3`:	Kruhová interpolace proti směru hodinových ručiček	
	Platnost:	modální
`AP=... RP=...`:	Koncový bod kruhu v polárních souřadnicích	
	`AP=...`:	Polární úhel
	`RP=...`:	Polární rádius (≙ rádius kruhu)

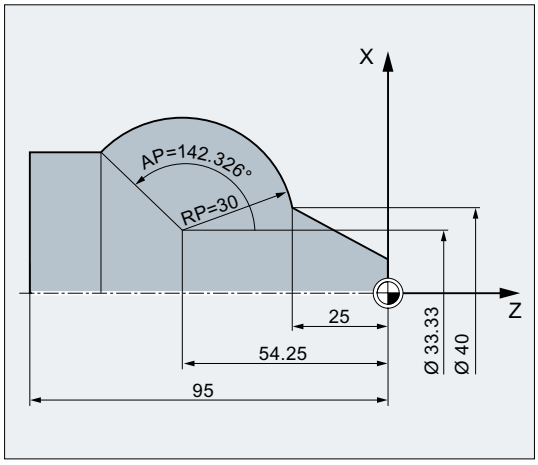
Příklady

Příklad 1: Frézování



Programový kód
N10 G0 X67.5 Y80.211
N20 G111 X50 Y50
N30 G3 RP=34.913 AP=200.052 F500

Příklad 2: Soustružení



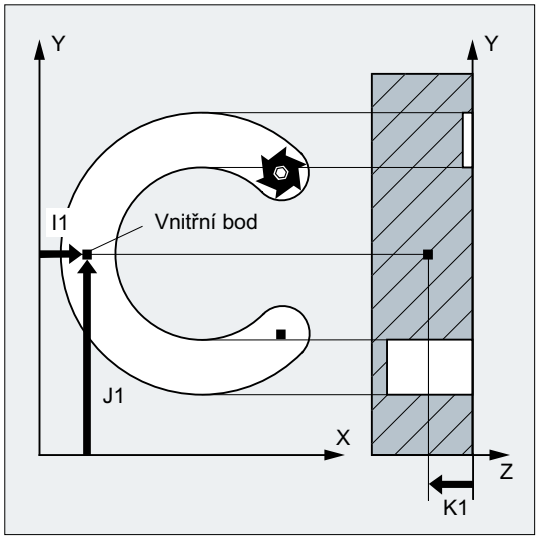
Programový kód
N125 G1 X40 Z-25 F0.2
N130 G111 X33.33 Z-54.25
N135 G3 RP=30 AP=142.326
N140 G1 Z-95

3.9.6.6 Kruhová interpolace s vnitřním a koncovým bodem (CIP, X... Y... Z..., I1... J1... K1...)

Varianta kruhové interpolace naprogramovaná pomocí příkazu G-funkce `CIP` umožňuje interpolaci kruhových oblouků ležících šikmo v prostoru.

Pohyb po kruhové dráze je popsán prostřednictvím **vnitřního bodů** a **koncového bodu** kruhové kontury.

Směr posuvu vyplývá z posloupnosti počáteční bod, → vnitřní bod, → koncový bod.



Syntaxe

```
CIP X... Y... Z... I1=AC(...) J1=AC(...) K1=(AC...)
```

Význam

`CIP`:	Kruhová interpolace přes vnitřní bod	
	Platnost:	modální
`X... Y... Z...`:	Koncový bod kruhu v kartézských souřadnicích V závislosti na momentálně platném nastavení zadávání rozměrů `G90`/`G91`, příp. `...=AC(...)` / `...=IC(...)`, jsou souřadnice koncového bodu kruhu interpretovány buď v absolutních nebo v inkrementálních rozměrech.	
`I1... J1... K1...`:	Interpolační parametry pro zadávání souřadnic vnitřního bodu kruhu ve směrech X, Y, Z V závislosti na momentálně platném nastavení zadávání rozměrů `G90`/`G91`, příp. `...=AC(...)` / `...=IC(...)`, jsou souřadnice vnitřního bodu kruhu interpretovány buď v absolutních nebo v inkrementálních rozměrech. **Upozornění:** Jeden interpolační parametr s hodnotou 0 může být vypuštěn, druhý související parametr však musí být v každém případě zadán.	

Poznámka

Předvolba `G90`/`G91` (zadávání absolutních nebo inkrementálních rozměrů) je platná pro vnitřní i pro koncový bod kruhu.

Když je aktivní zadávání inkrementálních rozměrů pomocí příkazů `G91`, příp. `...=IC(...)`, platí jako vztažný bod pro vnitřní a koncový bod počáteční bod kruhového oblouku.

Poznámka

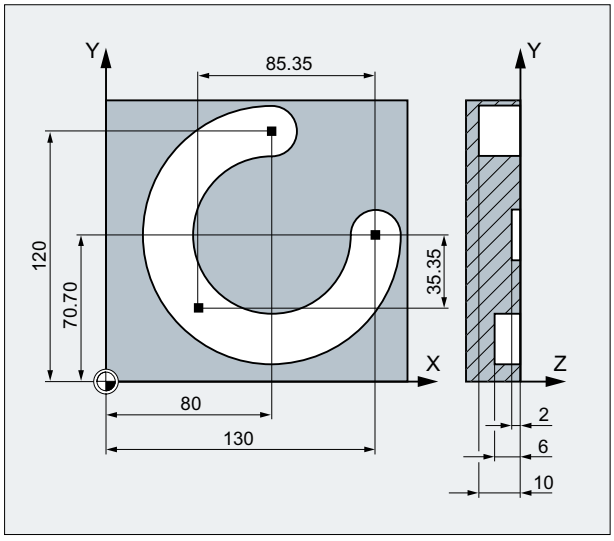
Technologie soustružení

Programování průměrů při zadávání interpolačních parametrů pro příčnou osu není pro programování kruhových oblouků pomocí příkazu `CIP` podporováno. Interpolační parametry pro příčnou osu je proto potřeba programovat v **rádiusech**.

Příklady

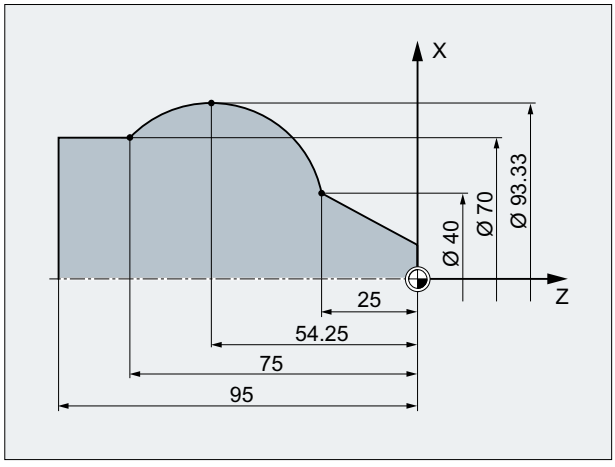
Příklad 1: Frézování

Pro výrobu šikmo v prostoru ležící kruhové drážky je popisován kruh zadáním vnitřního bodu se třemi interpolačními parametry a koncový bod rovněž se třemi souřadnicemi.



Programový kód	**Komentář**
`N10 G0 G90 X130 Y70.70 S800 M3`	`; najetí na počáteční bod`
`N20 G17 G1 Z-2 F100`	`; přísuv nástroje`
`N30 CIP X80 Y120 Z-10 I1=IC(-85.35) J1=IC(-35.35) K1=-6`	`; koncový bod a vnitřní bod kruhového oblouku`
	`; souřadnice pro všechny 3 geometrické osy`
`N40 M30`	`; konec programu`

Příklad 2: Soustružení



Programový kód	**Komentář**
...	
N125 G1 G90 X40 Z-25 F0.2	
N130 CIP X70 Z-75 **I1=IC(26.665)** K1=IC(-29.25)	; interpolační parametr I1 pro příčnou osu musí být naprogramován jako rádius
; příp.	
; N130 CIP X70 Z-75 **I1=46.665** K1=-54.25	
N135 G1 Z-95	

3.9.6.7 Kruhová interpolace s tangenciálním přechodem (CT, X... Y... Z...)

Varianta kruhové interpolace naprogramovaná pomocí příkazu G-funkce `CT` umožňuje interpolaci kruhových oblouků, které se tangenciálně napojují na předtím naprogramovaný konturový prvek.

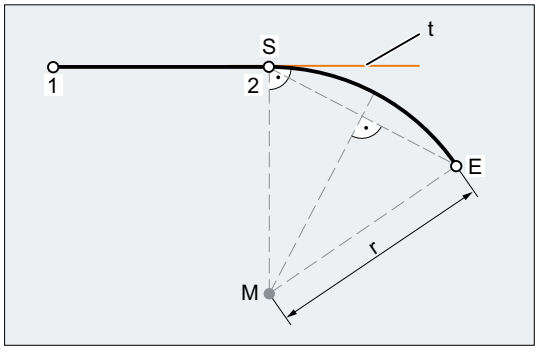
Kruh je přitom definován pomocí **počátečního a koncového bodu** a **směrnice tečny v počátečním bodě**.

Poznámka

Směrnice tečny v počátečním bodě

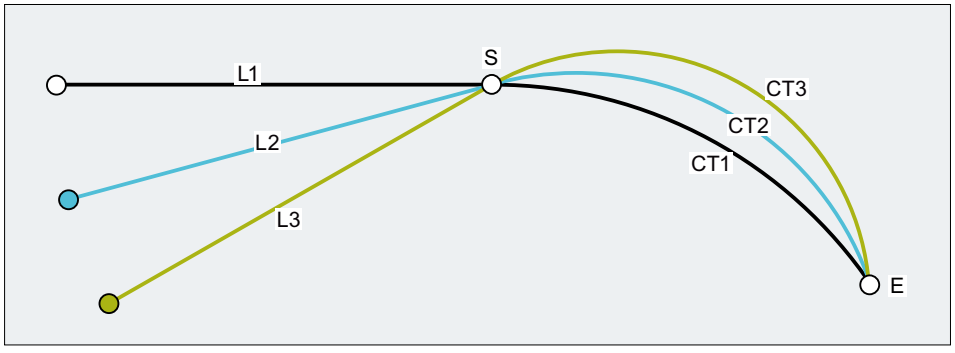
Směr tečny v počátečním bodě bloku s příkazem CT se určuje z koncové tečny naprogramované kontury v posledním předešlém bloku s příkazem posuvu.

Mezi tímto blokem a aktuálním blokem se může nacházet libovolný počet bloků bez informací o posuvu.



S Počáteční bod
E Koncový bod
M Střed kruhu
r Rádius kruhu
t Směr koncové tečny naprogramované kontury v posledním předešlém bloku s příkazem posuvu

Obrázek 3-4 Na přímkový úsek 1-2 tangenciálně navazující kruhová dráha S-E



Obrázek 3-5 Tangenciálně navazující kruhová dráha závisí na předcházejícím konturovém prvku

Syntaxe

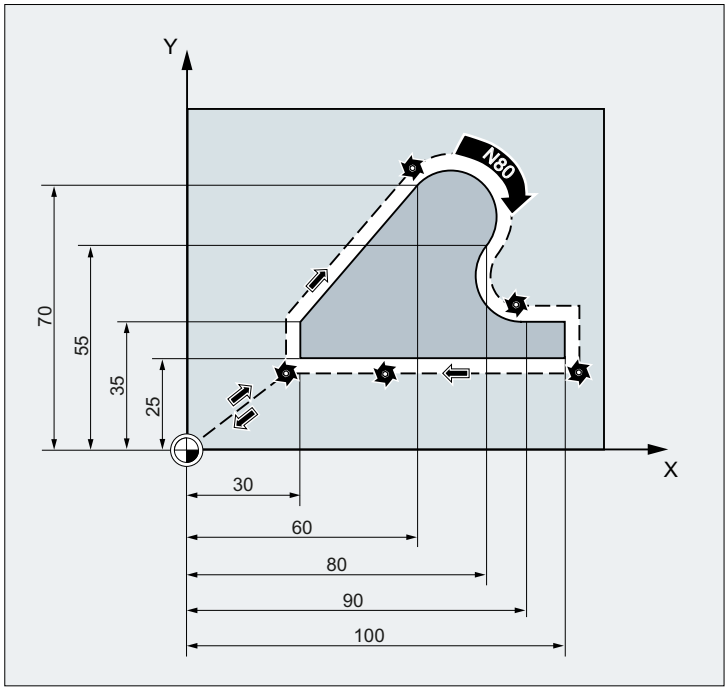
```
CT X... Y... Z...
```

Význam

`CT`:	Kruhová interpolace s tangenciálním přechodem	
	Platnost:	modální
`X... Y... Z...`:	Koncový bod kruhu v kartézských souřadnicích V závislosti na momentálně platném nastavení zadávání rozměrů `G90`/`G91`, příp. `...=AC(...)` / `...=IC(...)`, jsou souřadnice koncového bodu kruhu interpretovány buď v absolutních nebo v inkrementálních rozměrech.	

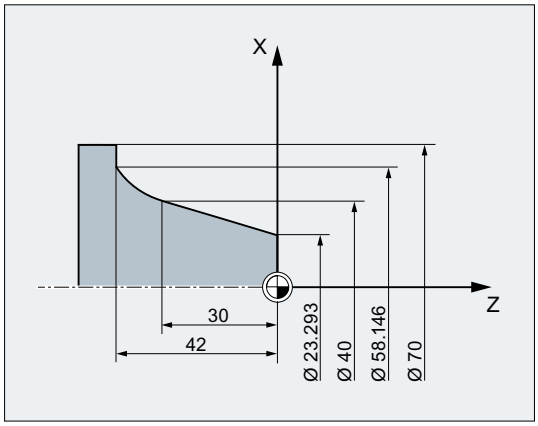
Příklady

Příklad 1: Frézování



Programový kód	Komentář
N10 G0 Z100	
N20 G17 T1 M6	
N30 G0 X0 Y0 Z2 M3 S300 D1	
N40 Z-5 F1000	; přísuv nástroje
N50 G41 X30 Y25 G1 F1000	; Aktivování korekce rádiusu nástroje
N60 Y35	; frézování kontury
N70 X60 Y70	
N80 CT X80 Y55	**; programování kruhového oblouku s tangenciálním přechodem**
N90 X90 Y35	
N100 G1 X100	
N110 Y25	
N120 X30	
N130 G0 G40 X0 Y0	; deaktivování korekce rádiusu nástroje
N140 Z100	; vyjíždění nástroje
N140 M30	

Příklad 2: Soustružení




```
Programový kód                  Komentář
...
N110 G1 X23.293 Z0 F10
N115 X40 Z-30 F0.2
N120 CT X58.146 Z-42            ; programování kruhového oblouku s tangenciálním pře-
                                chodem
N125 G1 X70
...
```

Další informace

Spliny

V případě splinů bude směr tečny určen přímkou vedenou posledními dvěma body. Obecně platí, že tento směr není u A-splinů a C-splinů při aktivních příkazech ENAT a EAUTO shodný se směrem v koncovém bodě splinu.

Přechod v případě B-splinů je vždy tangenciální, přičemž směr tečny je definován stejně jako u A-splinů a C-splinů a při aktivním příkazu ETAN.

Změna framu

Pokud se mezi blokem, který definuje tečku, a blokem s příkazem CT uskutečňuje změna framu, bude i tečna podléhat změně framu.

Mezní případ

Prochází-li prodloužení počáteční tečny koncovým bodem, vznikne namísto kruhu přímka (mezní případ kruhu s nekonečným rádiusem). V tomto speciálním případě nesmí být příkaz TURN vůbec naprogramován nebo musí být zadáno TURN=0.

Poznámka

Při přibližování se tomuto meznímu případu vznikají kruhy s libovolně velkým rádiusem, takže když se TURN nerovná nule, je zpracování programu obvykle přerušeno alarmem kvůli narušení softwarových mezních hodnot.

Poloha roviny kruhu

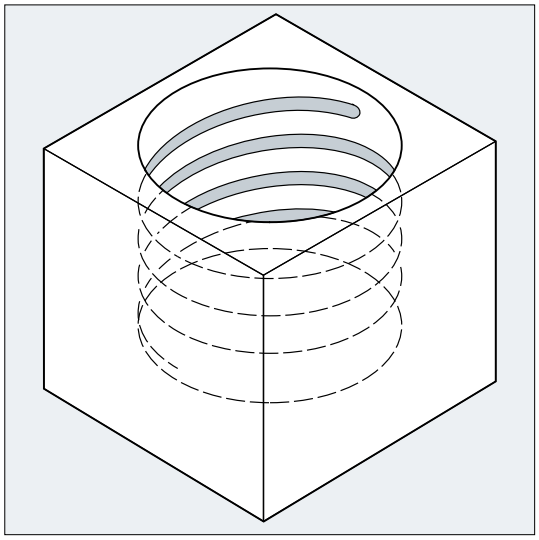
Poloha roviny kruhu závisí na aktivní rovině (G17-G19).

Pokud tečna z předcházejícího bloku neleží v aktivní rovině, použije se její průmět do aktivní roviny.

Jestliže souřadnicové složky polohy počátečního a koncového bodu kolmé k aktivní rovině nejsou stejné, namísto kruhu se bude vytvářet šroubovice.

3.9.7 Spirální interpolace (G2/G3, TURN)

Spirální interpolace (po šroubovici) umožňuje například výrobu závitů nebo mazacích drážek.



Při spirální interpolaci jsou superponovány a paralelně uskutečňovány dva pohyby:

- kruhový pohyb v rovině
- kolmý lineární pohyb

Syntaxe

```
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=
```

Význam

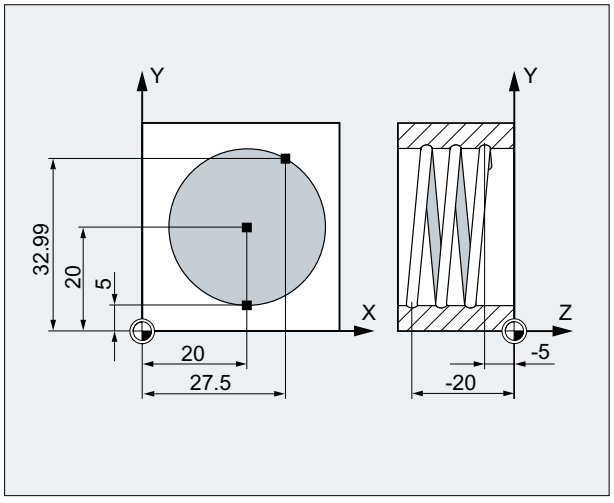
`G2`:	Pohyb po kruhové dráze ve směru hodinových ručiček
`G3`:	Interpolace po kruhové dráze proti směru hodinových ručiček
`X Y Z`:	Koncový bod v kartézských souřadnicích
`I J K`:	Střed kruhu v kartézských souřadnicích

AR:	Úhel kruhové výseče
TURN= :	Počet oběhů kružnice v rozsahu 0 až 999
AP= :	Polární úhel
RP= :	Polární rádius

Poznámka

Příkazy G2 a G3 mají modální platnost.

Příklad



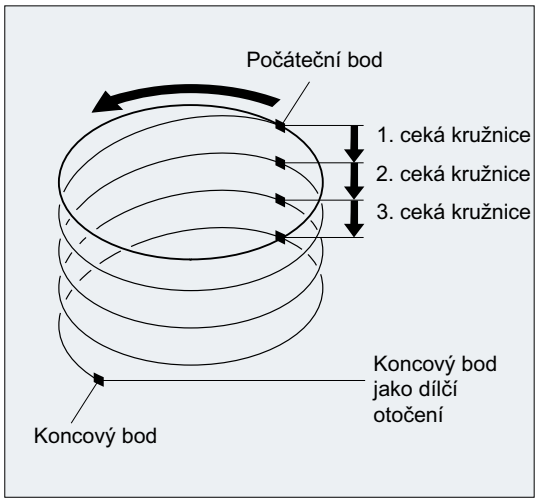
Programový kód	Komentář
N10 G17 G0 X27.5 Y32.99 Z3	; najíždění na počáteční pozici
N20 G1 Z-5 F50	; přísuv nástroje
N30 G3 X20 Y5 Z-20 I=AC(20) J=AC(20) TURN=2	; spirála s následujícími parametry: Od počáteční pozice se uskuteční 2 celé kružnice, pak se najíždí na konečný bod.
N40 M30	; konec programu

Další informace

Posloupnost pohybů

1. Najetí na počáteční bod
2. Uskutečnění celých kružnic naprogramovaných pomocí příkazu TURN=.
3. Najíždění na koncový bod kruhu, např. posuvem o část otáčky.
4. Uskutečnění bodů 2 a 3 po celé hloubce přísuvu.

Z počtu celých kružnic plus naprogramovaného koncového bodu kruhu (které se provádí po celé přísuvné hloubce) vyplývá stoupání, se kterým se má celá šroubovice vyrobit.



Programování koncového bodu spirální interpolace

Pokud budete potřebovat podrobné vysvětlení interpolačních parametrů, nahlédněte do kapitoly věnované kruhové interpolaci.

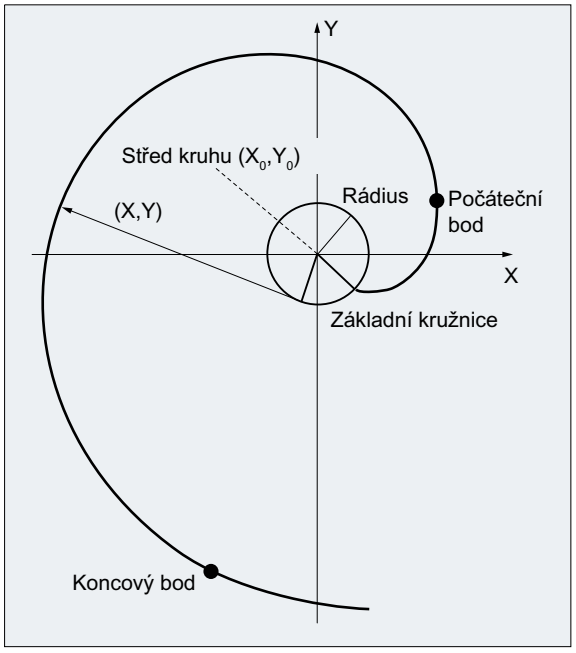
Naprogramovaný posuv

Při spirální interpolaci se doporučuje zadání naprogramované korekce posuvu (override) (`CFC`). Pomocí příkazu `FGROUP` můžete definovat, které osy se mají pohybovat naprogramovaným posuvem. Pokud budete potřebovat další informace, viz "Chování při pohybu po dráze".

3.9.8 Evolventní interpolace (INVCW, INVCCW)

Evolventa kruhu je křivka, která je popsána koncovým bodem pevného napnutého vlákna odvíjejícího se z kružnice.

Evolventní interpolace umožňuje dráhové křivky podél evolventy. Pohyb se uskutečňuje v rovině, ve které je definována základní kružnice, a probíhá z naprogramovaného počátečního bodu do naprogramovaného koncového bodu.



Programování koncového bodu může být realizováno dvěma způsoby:

1. Prostřednictvím kartézských souřadnic
2. Nepřímo zadáním úhlu kruhové výseče (viz také programování úhlu kruhové výseče pro programování kruhových oblouků).

Jestliže počáteční a koncový bod v rovině základní kružnice neleží, vznikne analogicky ke spirální interpolaci u kruhů superpozice křivky v prostoru.

Jestliže je ještě navíc zadán dráhový pohyb kolmo na aktivní rovinu, je možné (podobně jako v případě spirální interpolace u kruhů) definovat evolventu v prostoru.

Syntaxe

```
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=...
```

Význam

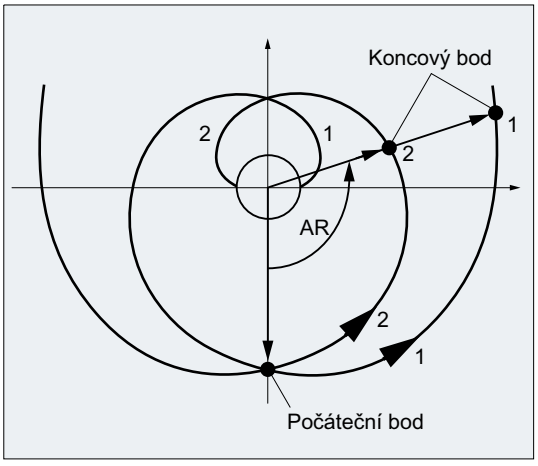
`INVCW`:	Příkaz pro pohyb evolventě ve směru hodinových ručiček
`INVCCW`:	Příkaz pro pohyb po evolventě proti směru hodinových ručiček
`X... Y... Z...`:	Přímé programování koncového bodu v kartézských souřadnicích
`I... J... K...`:	Interpolační parametry pro popis středu základní kružnice v kartézských souřadnicích **Upozornění:** Údaje jednotlivých souřadnic se vztahují na počáteční bod evolventy.
`CR=...`:	Rádius základní kružnice

AR=... :	Nepřímé programování koncového bodu zadáním úhlu kruhové výseče (úhel otáčení) Počátek úhlu kruhové výseče je přímka spojující střed kruhu a počáteční bod.	
	AR > 0:	Bod se pohybuje po evolventní dráze **pryč od základní kružnice**.
	AR < 0:	Bod se pohybuje po evolventní dráze **směrem k základní kružnici**. Pro AR < 0 je tedy maximální úhel omezen tím, že se koncový bod musí vždy nacházet mimo základní kružnici.

Nepřímé programování koncového bodu zadáním úhlu kruhové výseče

UPOZORNĚNÍ
Úhel kruhové výseče není definován V případě nepřímého programování koncového bodu zadáním úhlu kruhové výseče AR je potřeba věnovat pozornost znaménku úhlu, protože změna tohoto znaménka by měla za následek jinou evolventu a tím pádem i jinou dráhu.

Tyto záležitosti by měly být objasněny pomocí následujícího příkladu:

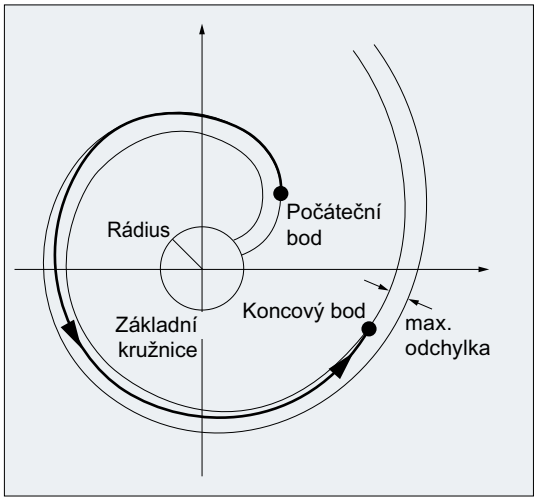


Zadané údaje pro rádius a střed základní kružnice, ale i počáteční bod a směr otáčení (INVCW / INVCCW), jsou pro evolventy 1 a 2 stejné. Jediný rozdíl spočívá ve znaménku úhlu kruhové výseče:

- Jestliže je AR > 0, pohybuje se bod po dráze evolventy 1, takže se najede do koncového bodu 1.
- Jestliže je AR < 0, pohybuje se bod po dráze evolventy 2, takže se najede do koncového bodu 2.

Okrajové podmínky

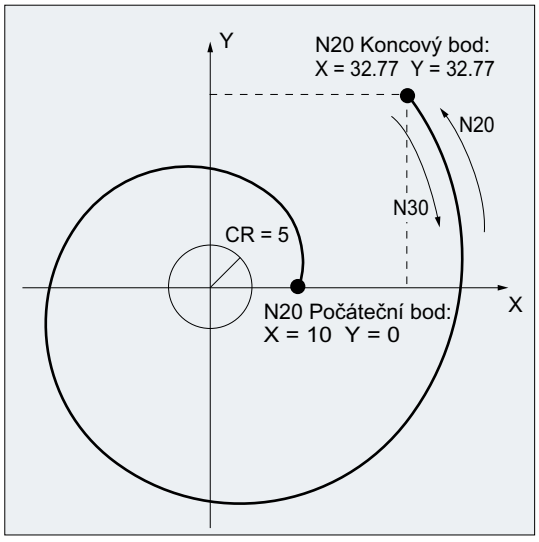
- Jak počáteční, tak i koncový bod musí ležet mimo plochu základní kružnice evolventy (kruh s rádiusem CR okolo středu definovaného souřadnicemi I, J, K). Pokud tato podmínka není splněna, je generován alarm a zpracování programu se přeruší.
- Obě možnosti naprogramování koncového bodu (přímo prostřednictvím kartézských souřadnic nebo nepřímo zadáním úhlu kruhové výseče) se vzájemně vylučují. V jednom bloku se proto smí použít jen jedna z těchto dvou možností programování.
- Jestliže naprogramovaný koncový bod neleží přesně na evolventě definované počátečním bodem a základní kružnicí, bude se provádět mezi oběma evolventami, které jsou definovány počátečním a koncovým bodem, interpolace (viz následující obrázek).



Maximální odchylka koncového bodu je stanovena strojním parametrem (--> výrobce stroje). Pokud je odchylka naprogramovaného koncového bodu v radiálním směru větší, než je hodnota daná tímto strojním parametrem, potom se generuje alarm a zpracování programu se přeruší.

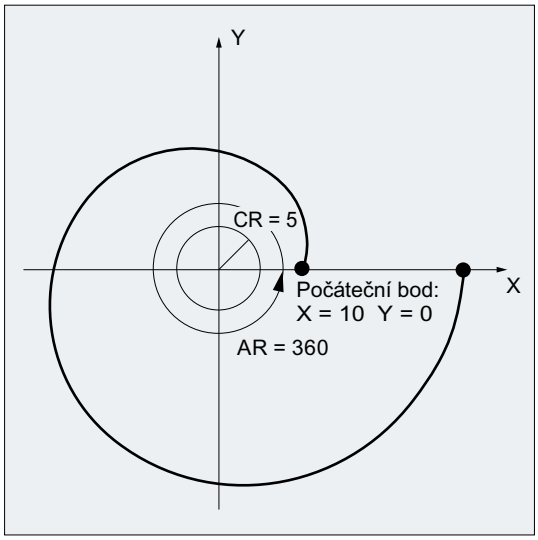
Příklady

Příklad 1: Levotočivá evolventa z počátečního bodu do naprogramovaného koncového bodu a pravotočivá evolventa pro zpětný pohyb zase zpátky



Programový kód	**Komentář**
N10 G1 X10 Y0 F5000	; Najíždění na počáteční pozici
N15 G17	; Volba pracovní roviny X/Y.
N20 INVCCW X32.77 Y32.77 CR=5 I-10 J0	; Evolventa proti směru hodinových ručiček, koncový bod v kartézských souřadnicích.
N30 INVCW X10 Y0 CR=5 I-32.77 J-32.77	; Evolventa ve směru hodinových ručiček, počátečním bodem je koncový bod z bloku N20, nový koncový bod je počáteční bod z bloku N20, nový střed kružnice se vztahuje na nový počáteční bod a rovná se starému středu kružnice.
...	

Příklad 2: Levotočivá evolventa s nepřímým naprogramováním koncového bodu zadáním úhlu kruhové výseče



Programový kód	**Komentář**
N10 G1 X10 Y0 F5000	; Najíždění na počáteční pozici
N15 G17	; Volba pracovní roviny X/Y.
N20 INVCCW CR=5 I-10 J0 AR=360	; Evolventa proti směru hodinových ručiček a směrem od základní kružnice (proto má zadaný úhel kladnou hodnotu) o jednu celou otáčku (360 stupňů).
...	

3.9.9 Definice kontur

3.9.9.1 Programování konturové křivky

Funkce

Programování průběhu kontury slouží pro rychlé zadávání jednoduchých kontur.

Zadáváním kartézských souřadnic a/nebo úhlů je možno programovat kontury s 1, 2, 3 nebo i více body a s přechodovými prvky, jimiž mohou být fasety nebo zaoblení (ANG příp. ANG1 a ANG2).

V blocích, které popisují konturu, je možno použít libovolné další NC adresy, jako např. adresová písmena pro další osy (jednotlivé osy nebo osu, která je kolmá na pracovní rovinu), příkazy pomocných funkcí, příkazy G-funkcí, zadání rychlosti atd.

Poznámka

Konturový počítač

Programování průběhu kontury je možno jednoduchým způsobem provádět také pomocí konturového počítače. Konturový počítač je nástroj uživatelského rozhraní, který umožňuje programování a grafické zobrazování jednoduchých a složitých kontur obrobku. Kontury naprogramované pomocí konturového počítače se přebírají do výrobního programu.

Další informace: Příručka pro obsluhu

Dosazení parametrů

Identifikátory pro úhel, rádius a fasetu jsou definovány prostřednictvím strojních parametrů:

MD10652 $MN_CONTOUR_DEF_ANGLE_NAME (název úhlu pro průběh kontury)

MD10654 $MN_RADIUS_NAME (název rádiusu pro průběh kontury)

MD10656 $MN_CHAMFER_NAME (název fasety pro průběh kontury)

Poznámka

Věnujte prosím pozornost informacím od výrobce stroje.

3.9.9.2 Definice kontur: Přímka

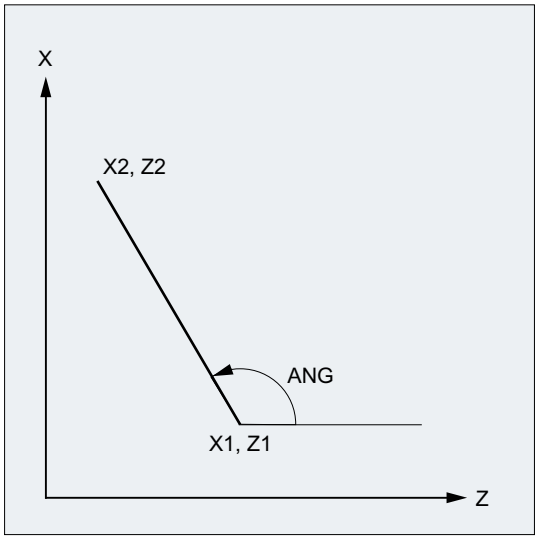
Poznámka

V následujícím popisu se vychází z následujících předpokladů:

- Je aktivní příkaz G18 (==> aktivní pracovní rovinou je rovina Z/X).
  (Programování průběhů kontur je však samozřejmě možné bez jakýchkoli omezení i v rovinách G17 nebo G19.)
- Pro úhel, rádius a fasetu jsou definovány následující identifikátory:
  - ANG (úhel)
  - RND (rádius)
  - CHR (faseta)

Koncový bod přímky je definován pomocí následujících údajů:

- Úhel ANG
- **Jedna** kartézská souřadnice koncového bodu (X2 nebo Z2)



ANG: Úhel přímky
X1, Z1: Souřadnice počátečního bodu
X2, Z2: Souřadnice koncového bodu přímky

Syntaxe

```
X... ANG=...
Z... ANG=...
```

Význam

`X...`:	Souřadnice koncového bodu ve směru X
Z...	Souřadnice koncového bodu ve směru Z
`ANG`:	Identifikátor pro programování úhlu Zadaná hodnota (úhel) je vztažena k abscise aktivní pracovní roviny (v případě roviny `G18` je to osa Z).

Příklad:

Programový kód	**Komentář**
`N10 X5 Z70 F1000 G18`	`; najíždění na počáteční pozici`
`N20 X88.8 ANG=110`	`; přímka se zadáním úhlu`
`N30 ...`	

Příp.:

Programový kód	**Komentář**
`N10 X5 Z70 F1000 G18`	`; najíždění na počáteční pozici`
`N20 Z39.5 ANG=110`	`; přímka se zadáním úhlu`
`N30 ...`	

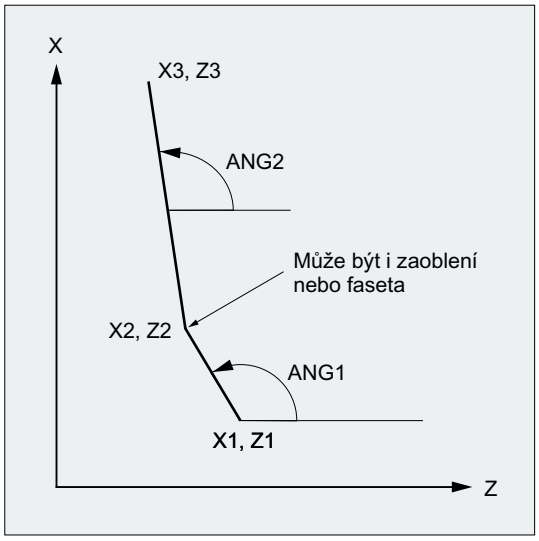
3.9.9.3 Definice kontur: Dvě přímky

Poznámka

V následujícím popisu se vychází z následujících předpokladů:

- Je aktivní příkaz G18 (==> aktivní pracovní rovinou je rovina Z/X).
  (Programování průběhů kontur je však samozřejmě možné bez jakýchkoli omezení i v rovinách G17 nebo G19.)
- Pro úhel, rádius a fasetu jsou definovány následující identifikátory:
  – ANG (úhel)
  – RND (rádius)
  – CHR (faseta)

Koncový bod první přímky může být naprogramován zadáním kartézských souřadnic nebo zadáním úhlu, který obě přímky svírají. Koncový bod druhé přímky musí být vždy naprogramován v kartézských souřadnicích. Průsečík obou přímek může být vyhotoven jako roh, zaoblení nebo faseta.



ANG1:	Úhel první přímky
ANG2:	Úhel druhé přímky
X1, Z1:	Souřadnice počátečního bodu první přímky
X2, Z2:	Souřadnice koncového bodu první přímky, resp. souřadnice počátečního bodu druhé přímky
X3, Z3:	Souřadnice koncového bodu druhé přímky

Syntaxe

Naprogramování koncového bodu první přímky zadáním úhlu

- Přechodovým prvkem mezi oběma přímkami je roh:

```
ANG=…
X… Z… ANG=…
```

- Přechodovým prvkem mezi oběma přímkami je zaoblení:

```
ANG=… RND=...
X… Z… ANG=…
```

- Přechodovým prvkem mezi oběma přímkami je faseta:

```
ANG=… CHR=...
X… Z… ANG=…
```

Naprogramování koncového bodu první přímky zadáním souřadnic

- Přechodovým prvkem mezi oběma přímkami je roh:

```
X… Z…
X… Z…
```

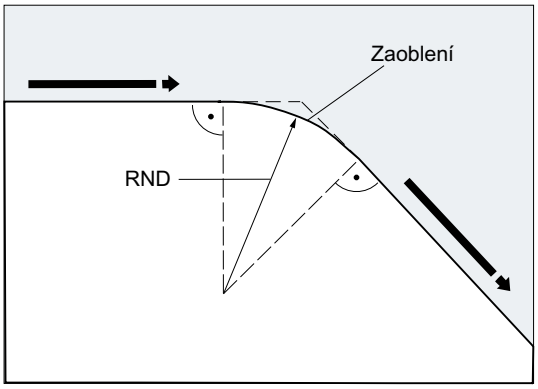
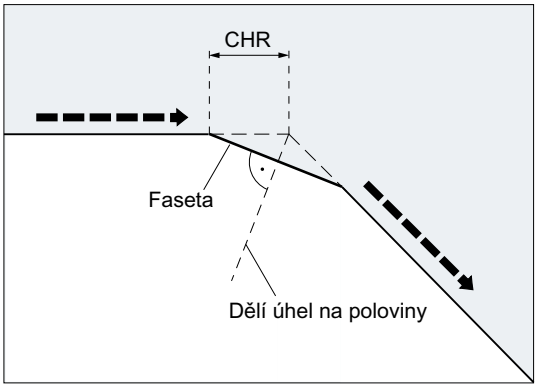
- Přechodovým prvkem mezi oběma přímkami je zaoblení:

```
X… Z… RND=...
X… Z…
```

- Přechodovým prvkem mezi oběma přímkami je faseta:

```
X… Z… CHR=...
X… Z…
```

Význam

`ANG=...` :	Identifikátor pro programování úhlu Zadaná hodnota (úhel) je vztažena k abscise aktivní pracovní roviny (v případě roviny `G18` je to osa Z).
`RND=...` :	Identifikátor pro programování zaoblení Uvedená hodnota odpovídá rádiusu zaoblení: 
`CHR=...` :	Identifikátor pro programování fasety Uvedená hodnota odpovídá šířce fasety ve směru pohybu: 
X...	Souřadnice ve směru X
Z...	Souřadnice ve směru Z

Poznámka

Pokud budete potřebovat podrobnější informace o programování faset nebo zaoblení, viz kapitola " Faseta, zaoblení (CHF, CHR, RND, RNDM, FRC, FRCM) (Strana 247) ".

Příklad:

Programový kód	Komentář
N10 X10 Z80 F1000 G18	; najíždění na počáteční pozici
N20 ANG=148.65 CHR=5.5	; přímka se zadáním úhlu a fasety
N30 X85 Z40 ANG=100	; přímka se zadáním úhlu a koncového bodu
N40 ...	

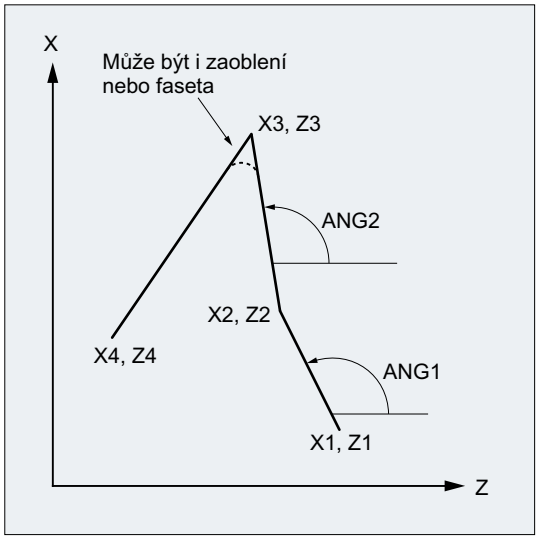
3.9.9.4 Definice kontur: Tři přímky

Poznámka

V následujícím popisu se vychází z následujících předpokladů:

- Je aktivní příkaz G18 (==> aktivní pracovní rovinou je rovina Z/X).
  (Programování průběhů kontur je však samozřejmě možné bez jakýchkoli omezení i v rovinách G17 nebo G19.)
- Pro úhel, rádius a fasetu jsou definovány následující identifikátory:
  – ANG (úhel)
  – RND (rádius)
  – CHR (faseta)

Koncový bod první přímky může být naprogramován zadáním kartézských souřadnic nebo zadáním úhlu, který obě přímky svírají. Koncový bod druhé a třetí přímky musí být vždy naprogramován v kartézských souřadnicích. Průsečík přímek může být vyhotoven jako roh, zaoblení nebo faseta.



ANG1:	Úhel první přímky
ANG2:	Úhel druhé přímky
X1, Z1:	Souřadnice počátečního bodu první přímky
X2, Z2:	Souřadnice koncového bodu první přímky, resp. souřadnice počátečního bodu druhé přímky
X3, Z3:	Souřadnice koncového bodu druhé přímky, resp. souřadnice počátečního bodu třetí přímky
X4, Z4:	Souřadnice koncového bodu třetí přímky

Poznámka

Způsob programování, který je zde vysvětlen pro případ kontury se 3 body, je možné libovolně rozšířit i pro kontury skládající se z více než tří bodů.

Syntaxe

Naprogramování koncového bodu první přímky zadáním úhlu

- Přechodovým prvkem mezi oběma přímkami je roh:

```
ANG=…
X… Z… ANG=…
X… Z…
```

- Přechodovým prvkem mezi oběma přímkami je zaoblení:

```
ANG=… RND=...
X… Z… ANG=… RND=...
X… Z…
```

- Přechodovým prvkem mezi oběma přímkami je faseta:

```
ANG=… CHR=...
X… Z… ANG=… CHR=...
X… Z…
```

Naprogramování koncového bodu první přímky zadáním souřadnic

- Přechodovým prvkem mezi oběma přímkami je roh:

```
X… Z…
X… Z…
X… Z…
```

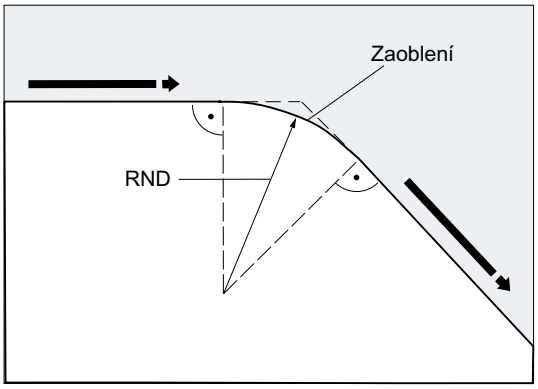
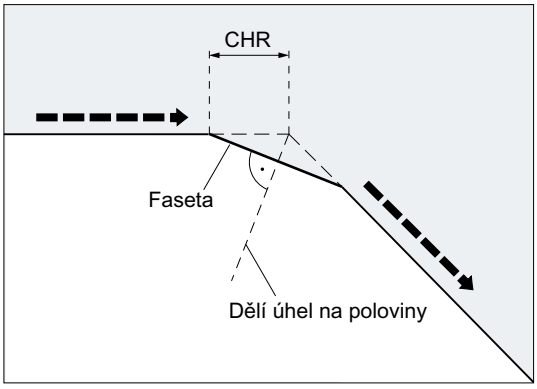
- Přechodovým prvkem mezi oběma přímkami je zaoblení:

```
X… Z… RND=...
X… Z… RND=...
X… Z…
```

- Přechodovým prvkem mezi oběma přímkami je faseta:

```
X… Z… CHR=...
X… Z… CHR=...
X… Z…
```

Význam

`ANG=...:`	Identifikátor pro programování úhlu Zadaná hodnota (úhel) je vztažena k abscise aktivní pracovní roviny (v případě roviny `G18` je to osa Z).
`RND=...:`	Identifikátor pro programování zaoblení Uvedená hodnota odpovídá rádiusu zaoblení: 
`CHR=...:`	Identifikátor pro programování fasety Uvedená hodnota odpovídá šířce fasety ve směru pohybu: 
`X...:`	Souřadnice ve směru X
`Z...:`	Souřadnice ve směru Z

Poznámka

Pokud budete potřebovat podrobnější informace týkající se programování fasety nebo rádiusu, viz " Faseta, zaoblení (CHF, CHR, RND, RNDM, FRC, FRCM) (Strana 247) ".

Příklad:

Programový kód	Komentář
N10 X10 Z100 F1000 G18	; najíždění na počáteční pozici
N20 ANG=140 CHR=7.5	; přímka se zadáním úhlu a fasety
N30 X80 Z70 ANG=95.824 RND=10	; přímka do vnitřního bodu se zadáním úhlu a zaoblení
N40 X70 Z50	; přímka do koncového bodu

3.9.9.5 Definice kontur: Programování koncového bodu pomocí úhlu

Funkce

Pokud se objeví v NC bloku adresové písmeno A, mohou být naprogramovány žádná, jedna nebo obě osy aktivní roviny.

Počet naprogramovaných os

- Jestliže není naprogramována **žádná osa** aktivní roviny, jedná se buď o první nebo o druhý blok kontury, která se skládá ze dvou bloků.
  Jestliže se jedná o druhý blok takové kontury, znamená to, že počáteční a koncový bod v aktivní rovině jsou identické. Kontura se potom skládá nanejvýš z jednoho pohybu kolmého na aktivní rovinu.
- Jestliže je naprogramována **právě jedna osa** aktivní roviny, jedná se buď o jednu přímku, jejíž koncový bod je jednoznačně určen úhlem a naprogramovanou kartézskou souřadnicí, nebo je to druhý blok v definici kontury skládající se ze dvou bloků. Ve druhém případě bude chybějící souřadnice rovna poslední dosažené (modální) poloze.
- Jestliže jsou naprogramovány **dvě osy** aktivní roviny, jedná se o druhý blok kontury skládající se ze dvou bloků. Pokud aktuálnímu bloku nepředcházel blok s naprogramovaným úhlem a bez zadaných os v aktivní rovině, je takový blok nepřípustný.

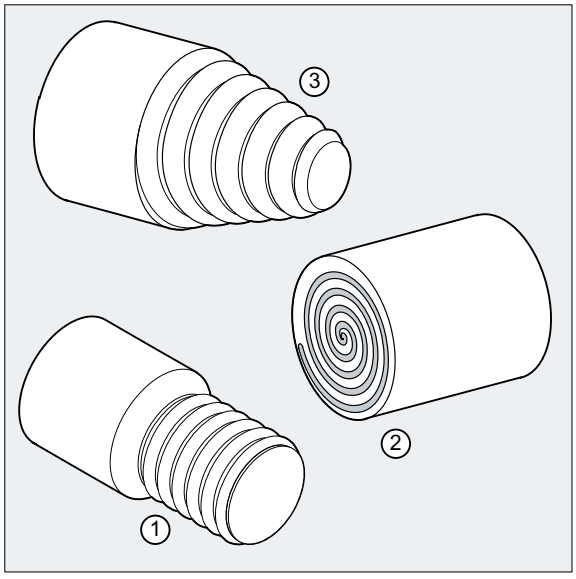
Úhel A smí být naprogramován pouze pro lineární nebo splinovou interpolaci.

3.9.10 Řezání závitu

3.9.10.1 Řezání závitu s konstantním stoupáním (G33, SF)

Pomocí příkazu G33 je možné vyrábět závity s konstantním stoupáním:

- Válcový závit ①
- Rovinný závit ②
- Kuželový závit ③

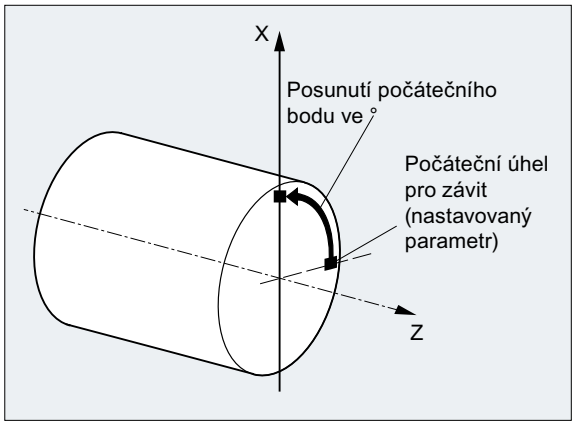


Poznámka

Technickým předpokladem pro tento způsob řezání závitů pomocí příkazu `G33` je vřeteno s regulací otáček se systémem pro měření dráhy.

Vícechodý závit

Vícechodé závity (závity s přesazenými řezy) je možné vyrábět pomocí zadání posunutí počátečního bodu. Programování se uskutečňuje v bloku s příkazem `G33` a s adresou `SF`.

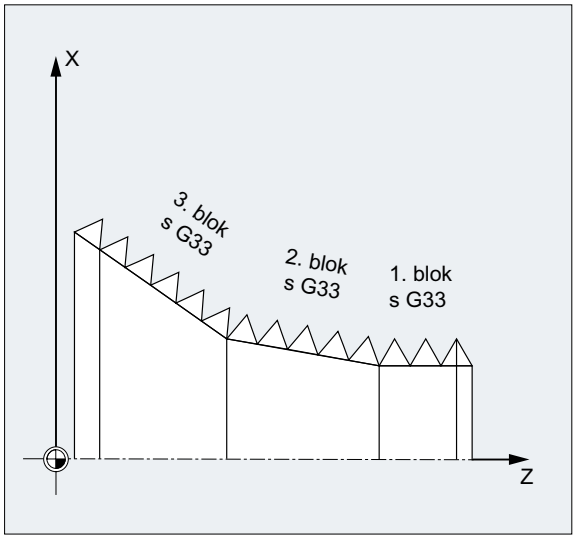


Poznámka

Jestliže žádné posunutí počátečního bodu není definováno, použije se „Počáteční úhel pro závity“ v nastavovaných parametrech.

Řetězec závitů

Prostřednictvím většího počtu za sebe naprogramovaných bloků s příkazem `G33` lze vyrábět řetězce závitů.



Poznámka

Pomocí příkazu `G64` (režim řízení pohybu po dráze) se jednotlivé bloky díky předvídání hodnoty rychlosti na několik bloků dopředu pospojují tak, aby nevznikly žádné skokové změny rychlosti.

Směr otáčení závitu

Směr otáčení závitu se určuje stanovením směru otáčení vřetena:

- Zadáním směru otáčení vpravo příkazem `M3` se vyrobí pravý závit.
- Zadáním směru otáčení vlevo `M4` se vyrobí levý závit.

Syntaxe

Válcový závit:
```
G33 Z… K…
G33 Z… K … SF=…
```

Rovinný závit:
```
G33 X… I…
G33 X… I… SF=…
```

Kuželový závit:
```
G33 X… Z… K…
G33 X… Z… K… SF=…
G33 X… Z… I…
G33 X… Z… I… SF=…
```

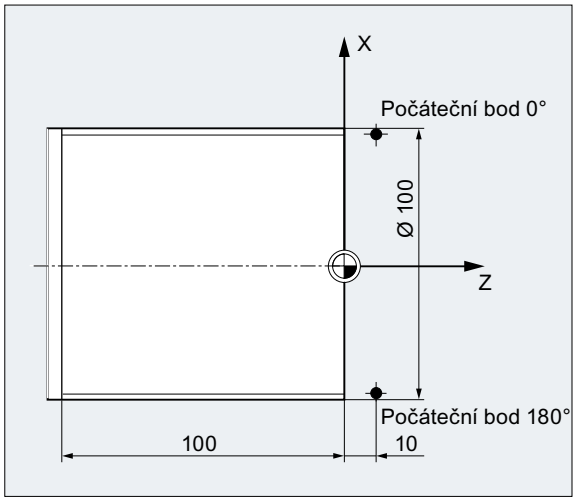
Význam

`G33`:	Příkaz pro řezání závitů s konstantním stoupáním
`X... Y... Z...`:	Koncový bod nebo body v kartézských souřadnicích
`I...`:	Stoupání závitu ve směru X

J...:	Stoupání závitu ve směru Y	
K...:	Stoupání závitu ve směru Z	
Z:	Podélná osa	
X:	Příčná osa	
Z... K...:	Délka závitu a stoupání závitu pro válcové závity	
X... I...:	Průměr závitu a stoupání závitu pro příčné závity	
I... nebo K...	Stoupání závitu pro kuželový závit To, který parametr je zadán (I... nebo K...), se řídí podle úhlu kužele.	
	< 45°:	Stoupání závitu se zadává pomocí parametru K... (stoupání závitu v podélném směru).
	> 45°:	Stoupání závitu se zadává pomocí parametru I... (stoupání závitu v příčném směru).
	= 45°:	Stoupání závitu je možno zadat pomocí parametru I... nebo K....
SF=...:	Posunutí počátečního bodu (potřebné jen v případě vícechodých závitů) Posunutí počátečního bodu se zadává jako absolutní úhlová pozice.	
	Rozsah hodnot:	0.0000 až 359.999 stupňů

Příklady

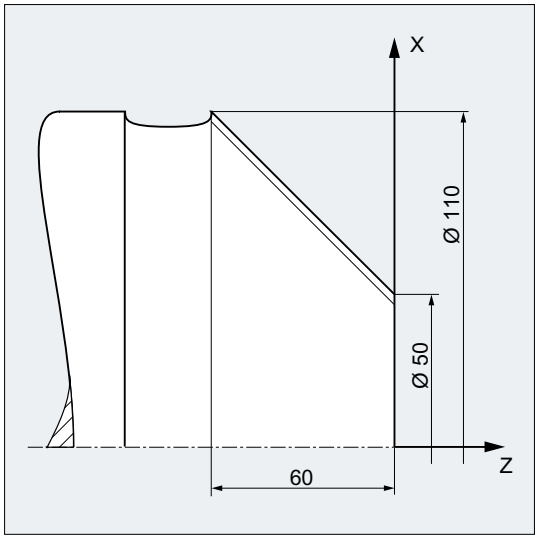
Příklad 1: Příklad dvouchodého válcového závitu s posunutím počátečního bodu o 180°



Programový kód	Komentář
N10 G1 G54 X99 Z10 S500 F100 M3	; posunutí počátku, najíždění na počáteční bod, zapnutí vřetena
N20 G33 Z-100 K4	; válcový závit: koncový bod v ose Z
N30 G0 X102	; návrat na počáteční pozici
N40 G0 Z10	
N50 G1 X99	
N60 G33 Z-100 K4 SF=180	; 2. chod: posunutí počátečního bodu o 180°
N70 G0 X110	; vyjíždění nástroje

Programový kód	Komentář
N80 G0 Z10	
N90 M30	; konec programu

Příklad 2: Kuželový závit s úhlem menším než 45°

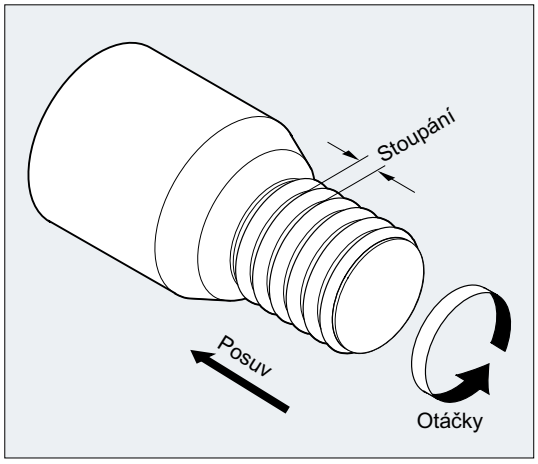


Programový kód	Komentář
N10 G1 X50 Z0 S500 F100 M3	; najíždění na počáteční bod, zapnutí vřetena
N20 G33 X110 Z-60 K4	; kuželový závit: Koncový bod v osách X a Z, zadání stoupání závitu pomocí parametru K... ve směru osy Z (protože úhel < 45°).
N30 G0 Z0 M30	; odjíždění nástroje, konec programu

Další informace

Posuv při řezání závitu pomocí příkazu G33

Na základě naprogramovaných otáček vřetena a stoupání závitu řídící systém vypočítá potřebný posuv, se kterým se bude soustružnický nůž pohybovat po délce závitu v podélném a/nebo příčném směru. Posuv `F` v případě příkazu `G33` není zohledňován, řídícím systémem je sledováno omezení na maximální rychlost os (rychlý posuv).



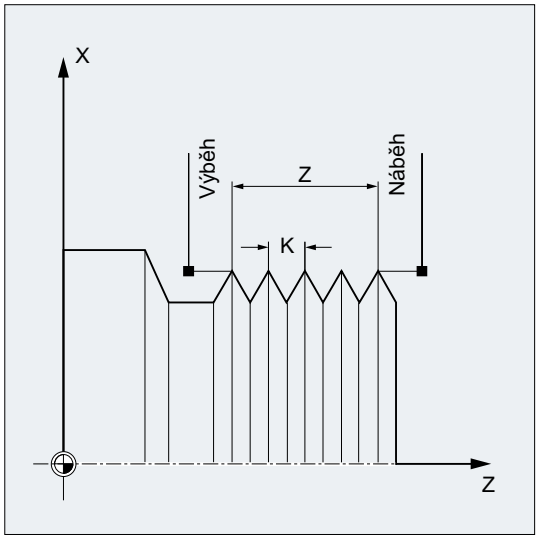
Válcové závity

Válcový závit je popsán následujícími parametry:

- Délka závitu
- Stoupání závitu

Délka závitu se zadává jednou z kartézských souřadnic X, Y nebo Z v absolutních nebo inkrementálních rozměrech (v případě soustruhu nejlépe ve směru osy Z). Navíc je potřeba brát ohled také na náběh a výběh závitu, na kterých posuv narůstá nebo se snižuje.

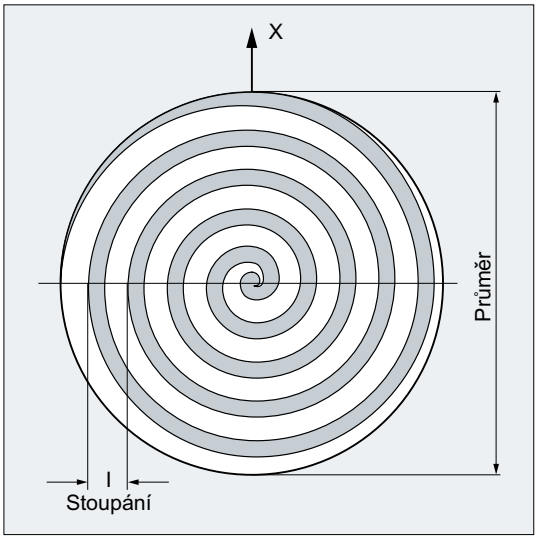
Stoupání závitu se zadává do adres `I`, `J`, `K` (u soustruhů nejlépe pomocí `K`).



Rovinné závity

Rovinný závit je popsán následujícími parametry:

- Průměr závitu (nejlépe ve směru X)
- Stoupání závitu (nejlépe pomocí `I`)



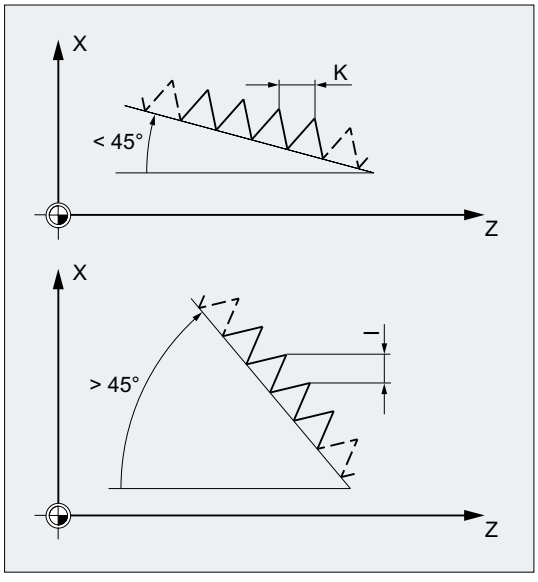
Kuželové závity

Kuželový závit je popsán následujícími parametry:

- Koncový bod v podélném a v příčném směru (kontura kuželu)
- Stoupání závitu

Kontura kuželu se zadává v kartézských souřadnicích X, Z, Z v absolutních nebo inkrementálních rozměrech, při obrábění na soustruzích nejraději ve směrech X a Z. Navíc je potřeba brát ohled také na náběh a výběh závitu, na kterých posuv narůstá nebo se snižuje.

Způsob zadání stoupání se řídí podle úhlu kuželu (úhel mezi podélnou osou a pláštěm kuželu):



3.9.10.2 Řezání závitů s narůstajícím nebo s klesajícím stoupáním (G34, G35)

Pomocí příkazů G34 a G35 byla funkce G33 rozšířena o možnost naprogramovat pomocí adresy F navíc ještě i změnu stoupání závitu. V případě příkazu G34 se jedná o lineární zvyšování, v případě příkazu G35 o lineární snižování hodnoty stoupání závitu. Příkazy G34 a G35 je tedy možné používat pro výrobu samosvorných závitů.

Syntaxe

Válcový závit se zvyšujícím se stoupáním:
```
G34 Z... K... F...
```
Válcový závit se snižujícím se stoupáním:
```
G35 Z... K... F...
```
Rovinný závit se zvyšujícím se stoupáním:
```
G34 X... I... F...
```
Rovinný závit s klesajícím stoupáním:
```
G35 X... I... F...
```
Kuželový závit se zvyšujícím se stoupáním:
```
G34 X... Z... K... F...
G34 X... Z... I... F...
```
Kuželový závit s klesajícím stoupáním:
```
G35 X... Z... K... F...
G35 X... Z... I... F...
```

Význam

`G34`:	Příkaz pro řezání závitů s lineárně **se zvyšujícím** stoupáním	
`G35`:	Příkaz pro řezání závitů s lineárně **se snižujícím** stoupáním	
`X... Y... Z...`:	Koncový bod nebo body v kartézských souřadnicích	
`I...`:	Stoupání závitu ve směru X	
`J...`:	Stoupání závitu ve směru Y	
`K...`:	Stoupání závitu ve směru Z	
F...	Změna stoupání závitu Jestliže je počáteční a koncové stoupání závitu známo, může být změna stoupání závitu, kterou je zapotřebí naprogramovat, vypočítána podle následující rovnice: $F = \frac{k_e^2 - k_a^2}{2 * l_G}$ [mm/ot^2] Přitom platí:	
	k_e:	koncové stoupání závitu (stoupání závitu v cílovém bodě souřadné osy [mm/ot])
	k_a:	počáteční stoupání závitu (naprogramované pomocí I, J nebo K) [mm/ot])
	l_G:	délka závitu [mm]

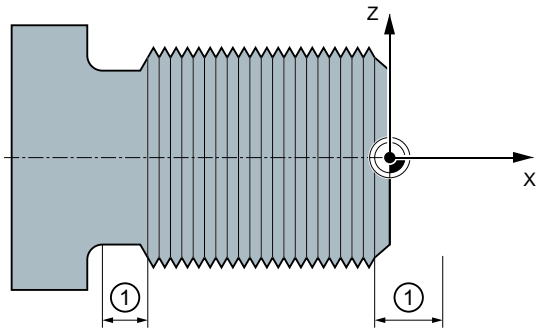
Příklad:

Programový kód	**Komentář**
N1608 M3 S10	; zapnutí vřetena
N1609 G0 G64 Z40 X216	; najetí na počáteční bod
N1610 G33 Z0 K100 SF=R14	; řezání závitu s konstantním stoupáním (100 mm/ot)
N1611 G35 Z-200 K100 F17.045455	; zmenšení stoupání závitu: 17,0454 mm/U2
	Stoupání na konci bloku: 50 mm/ot
N1612 G33 Z-240 K50	; blok závitu bez trhavých pohybů
N1613 G0 X218	
N1614 G0 Z40	
N1615 M17	

3.9.10.3 Programovatelný náběh a výběh závitu pomocí příkazů G33, G34 a G35 (DITS, DITE)

Prostřednictvím adres DITS a DITE je možno ve výrobním programu zadat dráhu náběhu a výběhu závitu.

V rámci této předem zadané dráhy se uskutečňuje zrychlení, příp. zabrždění osy závitu.



① Dráha náběhu, resp. výběhu, v závislosti na směru obrábění

Kratší dráha pro náběh

V oblasti náběhu závitu je příliš málo místa pro náběžnou hranu charakteristiky rychlosti nástroje.
Proto je nutné zadat tuto dráhu kratší pomocí příkazu DITS.

Kratší dráha pro výběh

V oblasti výběhu závitu je příliš málo místa pro brzdnou hranu charakteristiky rychlosti nástroje, v důsledku čehož vzniká nebezpečí kolize mezi obrobkem a břitem nástroje. Prostřednictvím příkazu DITE je možné zadat kratší brzdnou rampu. Kvůli setrvačnosti mechaniky může ale přesto dojít ke kolizi.

Pomoc: Závit naprogramujte kratší, snižte otáčky vřetena.

Poznámka

Příkaz `DITE` působí na konci závitu jako přibližná vzdálenost. Díky tomu se dosáhne hladké změny pohybu osy.

Důsledky

Nárůst zrychlení se uskutečňuje výlučně na naprogramovaném úseku náběhu a výběhu dráhy. Jestliže byl jeden z obou úseků zadán delší, než je pro osu závitu s aktivním zrychlením zapotřebí, bude osa závitu zrychlována nebo zpomalována s maximálním zrychlením.

Syntaxe

```
DITS=<hodnota> DITE=<hodnota>
```

Význam

`DITS`:	Definice náběžné dráhy závitu
`DITE`:	Definice výběhu závitu
`<hodnota>`:	Do parametrů `DITS` a `DITE` jsou programovány výlučně úseky dráhy, nikoli pozice! Naprogramovaná dráha náběhu/výběhu je vyhodnocována v souladu s nastavením měřicích jednotek (mm/palce).

Příklad:

Programový kód	**Komentář**
`...`	
`N40 G90 G0 Z100 X10 SOFT M3 S500`	
`N50 G33 Z50 K5 SF=180 DITS=1 DITE=3`	`; začátek zaoblování rohu na Z=53`
`N60 G0 X20`	

Další informace

SD42010 $SC_THREAD_RAMP_DISP

Při přechodu na další blok s příkazem `DITS` a/nebo `DITE` v hlavní větvi programu se naprogramovaná dráha náběhu, příp. výběhu, přenese do nastavovaného parametru SD 42010 $SC_THREAD_RAMP_DISP:

- SD42010 $SC_THREAD_RAMP_DISP[**0**] = naprogramovaná hodnota `DITS`
- SD42010 $SC_THREAD_RAMP_DISP[**1**] = naprogramovaná hodnota `DITE`

Jestliže není předem nebo v prvním bloku závitu naprogramována žádná dráha náběhu nebo výběhu, použije se momentální hodnota uložená v nastavovaném parametru.

Chování po resetu kanálu/BAG/konci programu

Hodnoty parametru SD42010, které byly přepsány prostřednictvím příkazů `DITS` a/nebo `DITE`, zůstávají zachovány i po resetu kanálu/BAG a po skončení programu.

Chování po teplém startu

V případě teplého startu jsou nastavovaným parametrům znovu dosazeny hodnoty, které byly v platnosti před jejich přepsáním pomocí příkazů `DITS` a/nebo `DITE` (standardní chování).

Jestliže by ale měly oproti tomu hodnoty naprogramované pomocí příkazů `DITS` a `DITE` zůstat v platnosti i po teplém startu, musí být nastavovaný parametr SD42010 $SC_THREAD_RAMP_DISP uveden ve strojním parametru MD10710 $MN_PROG_SD_RESET_SAVE_TAB:

MD10710 $MN_PROG_SD_RESET_SAVE_TAB[<n>] = 42010

Chování v případě velmi krátkých drah náběhu a/nebo výběhu

Při velmi krátkých náběžných a/nebo výběžných drahách je zrychlení v ose závitu větší, než je hodnota nastavená v konfiguraci, což způsobuje přetížení zrychlení v dané ose.

Pro náběh závitu se potom aktivuje alarm 22280 „Příliš krátká náběžná dráha“ (při odpovídající konfiguraci ve strojním parametru MD11411 $MN_ENABLE_ALARM_MASK). Tento alarm je čistě informativní a nemá žádný vliv na zpracování výrobního programu.

3.9.10.4 Rychlý zpětný pohyb během řezání závitu (LFON, LFOF, DILF, ALF, LFTXT, LFWP, LFPOS, POLF, POLFMASK, POLFMLIN)

Funkce "Rychlý zpětný pohyb během řezání závitu (G33)" umožňuje bezproblémové přerušení procesu řezání závitu v následujících situacích:

- Bylo aktivováno NC-Stop prostřednictvím DB21, ... DBX7.3
- NC-Stop bylo aktivováno implicitně prostřednictvím alarmu
- Přepnutí rychlého vstupu (Strana 549)

Zpětný pohyb je možné naprogramovat prostřednictvím:

- Dráhy a směru zpětného pohybu (relativně)
- Polohy pro zpětný pohyb (absolutní)

Poznámka

Signály NC-Stop

Následující signály NS-Stop nespouštějí v průběhu řezání závitu žádný rychlý zpětný pohyb:

- DB21, ... DBX3.4 (NC-Stop os a vřeten)
- DB21, ... DBX7.2 (NC-Stop na konci bloku)

Vrtání závitů

Funkci "Rychlý zpětný pohyb" **není** možné použít v případě **vrtání** závitů (G331/G332).

Syntaxe

Uvolnění rychlého zpětného pohybu, zpětný pohyb pomocí dráhy a směru zpětného pohybu:

```
G33 ... LFON DILF=<hodnota> LFTXT/LFWP ALF=<hodnota>
```

Uvolnění rychlého zpětného pohybu, zpětný pohyb zadán pomocí koncové polohy tohoto pohybu:

```
POLF[<identifikátor osy>]=<hodnota> LFPOS
POLFMASK/POLFMLIN(<název osy 1>,<název osy 2>,...)
G33 ... LFON
```

Zablokování rychlého zpětného pohybu během řezání závitů:
`LFOF`

Význam

`LFON`:	Uvolnění rychlého zpětného pohybu v průběhu řezání závitu (`G33`)	
`LFOF`:	Zablokování rychlého zpětného pohybu v průběhu řezání závitu (`G33`)	
`DILF=` :	Definice délky dráhy zpětného pohybu	
	Hodnota předem nastavená v konfiguraci pomocí strojního parametru (MD21200 $MC_LIFTFAST_DIST) může být ve výrobním programu změněna naprogramováním příkazu `DILF`. **Upozornění:** Po resetu NC systému je vždy aktivní hodnota nastavená v konfiguraci pomocí MD.	
`LFTXT` `LFWP`:	Směr návratového pohybu je ve spojení s příkazem `ALF` řízen příkazy G-funkcí `LFTXT` a `LFWP`.	
	`LFTXT`:	Rovina, v níž se uskutečňuje zpětný pohyb, se vypočítává z tečny ke dráze a ze směru nástroje (standardní nastavení).
	`LFWP`:	Rovina, ve které se provádí zpětný pohyb, je aktivní pracovní rovina.
`ALF=` :	Směr se programuje v diskrétních úhlových krocích pomocí příkazu `ALF` v rovině zpětného pohybu.	
	V případě příkazu `LFTXT` je pro `ALF=1` definován návrat ve směru nástroje. V případě `LFWP` je směr v pracovní rovině přiřazen podle následujícího uspořádání: • `G17` (rovina X/Y) `ALF=1` ; zpětný pohyb ve směru osy X `ALF=3` ; zpětný pohyb ve směru osy Y • `G18` (rovina X/Z) `ALF=1` ; zpětný pohyb ve směru osy Z `ALF=3` ; zpětný pohyb ve směru osy X • `G19` (rovina Y/Z) `ALF=1` ; zpětný pohyb ve směru osy Y `ALF=3` ; zpětný pohyb ve směru osy Z	
	Pokud budete potřebovat další informace o možnostech programování s příkazem `ALF`, viz kapitola „Směr pohybu při rychlém pozvednutí od kontury (Strana 551)“.	
`LFPOS`:	Návrat osy stanovené příkazem `POLFMASK` nebo `POLFLIN` na absolutní pozici naprogramovanou pomocí příkazu `POLF`.	
`POLFMASK`:	Uvolnění os `(<název osy 1>,<název osy 1>,...)` pro nezávislý zpětný pohyb na absolutní pozici	
`POLFMLIN`:	Uvolnění os pro návrat na absolutní pozici v lineárním vztahu **Upozornění:** Lineárního vztahu nemůže být vždy dosaženo kvůli dynamickému chování všech podílejících se os až po dokončení zpětného pohybu.	

POLF[]:	Stanovení absolutní pozice zpětného pohybu geometrické osy, příp. osy stroje, která je uvedena v indexu.	
	Platnost:	modální
	=<hodnota>:	V případě geometrických os je přiřazená hodnota interpretována jako poloha v souřadném systému obrobku (WCS), v případě os stroje jako poloha v souřadném systému stroje (MCS). Přiřazení hodnoty může být naprogramováno i pomocí zadání inkrementálních rozměrů: =IC<hodnota>
<identifikátor osy>:	Identifikátor geometrické nebo strojní osy	

Poznámka

Příkazy LFON, příp. LFOF mohou být naprogramovány kdykoli, vyhodnocování se však provádí výlučně při řezání závitu (G33).

Poznámka

Příkaz POLF s příkazy POLFMASK/POLFMLIN nejsou omezeny na použití při řezání závitů.

Příklady

Příklad 1: Uvolnění rychlého zpětného pohybu v průběhu řezání závitů

Programový kód	Komentář
N55 M3 S500 G90 G18	; aktivní rovina obrábění
...	; najíždění na počáteční pozici
N65 MSG ("Řezání závitu")	; přísuv nástroje
MM_THREAD:	
N67 $AC_LIFTFAST=0	; vynulování před zahájením závitu
N68 G0 Z5	
N68 X10	
N70 G33 Z30 K5 LFON DILF=10 LFWP ALF=7	; uvolnění rychlého zpětného pohybu v průběhu řezání závitů Dráha zpětného pohybu = 10 mm Návratová rovina: Z/X (kvůli G18) Směr zpětného pohybu: -X (když ALF=3: směr zpětného pohybu +X)
N71 G33 Z55 X15	
N72 G1	; deaktivování řezání závitu
N69 IF $AC_LIFTFAST GOTOB MM_THREAD	; pokud došlo k přerušení řezání závitu
N90 MSG("")	
...	
N70 M30	

Příklad 2: Deaktivování rychlého zpětného pohybu před vrtáním závitu

Programový kód	Komentář
N55 M3 S500 G90 G0 X0 Z0	
...	
N87 MSG ("Vrtání závitu")	
N88 LFOF	; deaktivování rychlého zpětného pohybu před vrtáním závitu
N89 CYCLE...	; cyklus vrtání závitu s příkazem G33
N90 MSG("")	
...	
N99 M30	

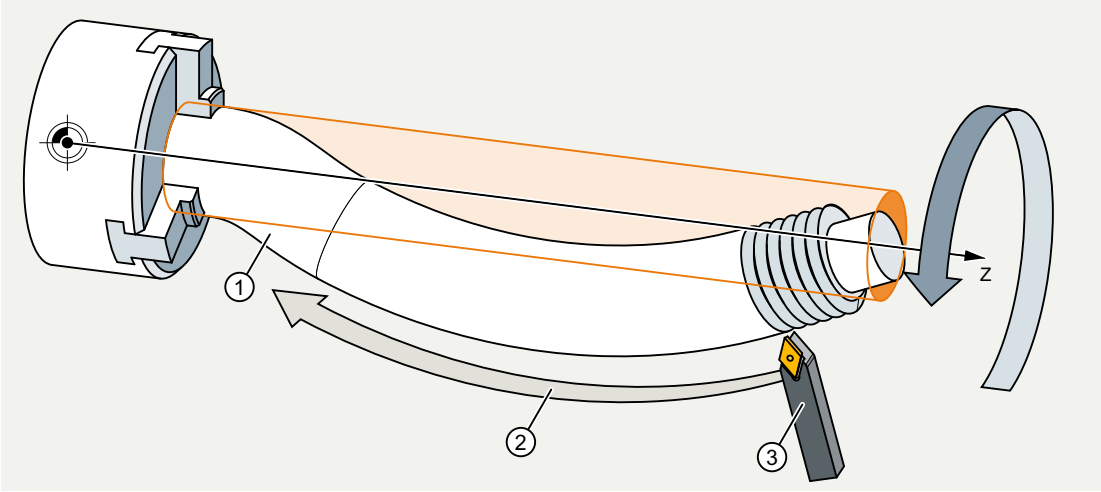
Příklad 3: Rychlý zpětný pohyb na absolutní návratovou pozici

V případě zastavení je dráhová interpolace osy X potlačena a místo toho se max. rychlostí uskutečňuje pohyb na pozici POLF[X]. Pohyb ostatních os bude i nadále určen naprogramovanou konturou, příp. stoupáním závitu a otáčkami vřetena.

Programový kód	Komentář
N10 G0 G90 X200 Z0 S200 M3	
N20 G0 G90 X170	
N22 POLF[X]=210 LFPOS	
N23 POLFMASK(X)	; aktivování (odblokování) rychlého pozvednutí osy 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()	; pozvednutí je pro všechny osy blokováno
M30	

3.9.10.5 Sudovitý závit (G335, G336)

Prostřednictvím příkazů G-funkcí G335 a G336 máte možnost soustružit sudovité (= lišící se od válcovitého tvaru) závity. Používají se pro obrábění extrémně velkých součástí, které se v důsledku své vlastní hmotnosti prohýbají. Pokud by byl závit opracováván rovnoběžně s osou, chod závitu by byl ve středu součásti příliš malý. Pomocí sudovitého závitu je možno toto prohnutí kompenzovat.



Obrázek 3-6 Soustružení sudovitého závitu

Programování

Soustružení sudovitého závitu je možno naprogramovat pomocí příkazů `G335` nebo `G336`:

`G335`:	Soustružení sudovitého závitu, přičemž nástroj se bude pohybovat po kruhovém oblouku **ve směru hodinových ručiček**.
`G336`:	Soustružení sudovitého závitu, přičemž nástroj se bude pohybovat po kruhovém oblouku **proti směru hodinových ručiček**.

Programování se napřed uskutečňuje stejně jako při lineárním závitu, to znamená, že se zadávají body odpovídající koncům bloku pro jednotlivé osy a stoupání prostřednictvím parametrů `I`, `J` a `K` (viz "Řezání závitu s konstantním stoupáním (G33, SF) (Strana 223)").

Navíc je potřeba ještě zadat kruhový oblouk. Tento kruhový oblouk je možno naprogramovat stejně jako v případě příkazů `G2`/`G3` prostřednictvím zadání středu, rádiusu, úhlu kruhové výseče nebo vnitřního bodu oblouku (viz "Kruhová interpolace (Strana 190)"). Při programování sudovitého závitu pomocí naprogramování středu je potřeba mít na paměti následující: Protože se parametry `I`, `J` a `K` při řezání závitů používají pro stoupání, musí být parametry kruhu při programování jeho středu zadány pomocí příkazů `IR=...`, `JR=...` a `KR=...`.

`IR=...`:	Kartézská souřadnice pro střed kruhu ve směru osy X
`JR=...`:	Kartézská souřadnice pro střed kruhu ve směru osy Y
`KR=...`:	Kartézská souřadnice pro střed kruhu ve směru osy Z

Poznámka

IR, JR a KR jsou standardní hodnoty pro názvy interpolačních parametrů pro sudovité závity, které jsou nastavitelné pomocí strojního parametru
(MD10651 $MN_IPO_PARAM_THREAD_NAME_TAB).

Odchylky od těchto standardních hodnot si musíte zjistit v dokumentaci od výrobce stroje!

Pokud se přejete, můžete pomocí parametru `SF` zadat ještě i posunutí počátečního bodu (viz "Řezání závitu s konstantním stoupáním (G33, SF) (Strana 223)").

Syntaxe

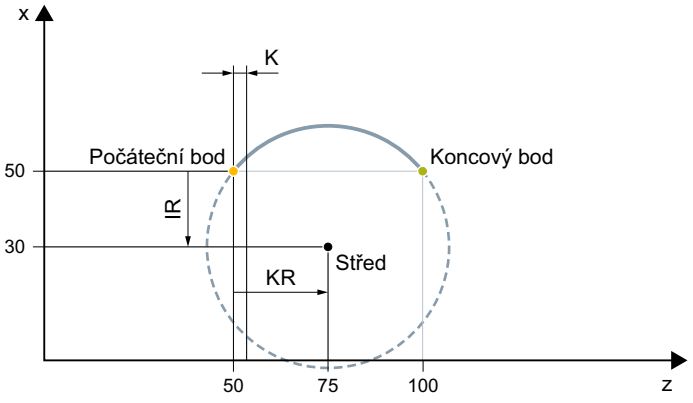
Obecná syntaxe při programování sudovitého závitu je tedy následující:

```
G335/G336 <souřadnice cílového bodu os)> <stoupání> <kruhový oblouk>
[<posunutí počátečního bodu>]
```

Příklady

Příklad 1: Sudovitý závit opracovávaný ve směru hodinových ručiček naprogramovaný pomocí koncového bodu a středu

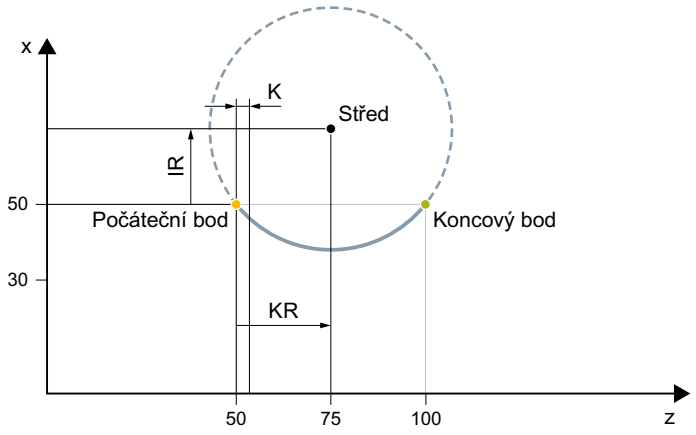
Programový kód	Komentář
N5 G0 G18 X50 Z50	; najíždění na počáteční bod
N10 **G335 Z100** K=3.5 **KR=25 IR=-20** SF=90	; soustružení sudovitého závitu ve směru hodinových ručiček



Obrázek 3-7 Sudovitý závit opracovávaný ve směru hodinových ručiček naprogramovaný pomocí koncového bodu a středu

Příklad 2: Sudovitý závit opracovávaný proti směru hodinových ručiček naprogramovaný pomocí koncového bodu a středu

Programový kód	Komentář
N5 G0 G18 X50 Z50	; najíždění na počáteční bod
N10 **G336 Z100** K=3.5 **KR=25 IR=20** SF=90	; soustružení sudovitého závitu proti směru hodinových ručiček

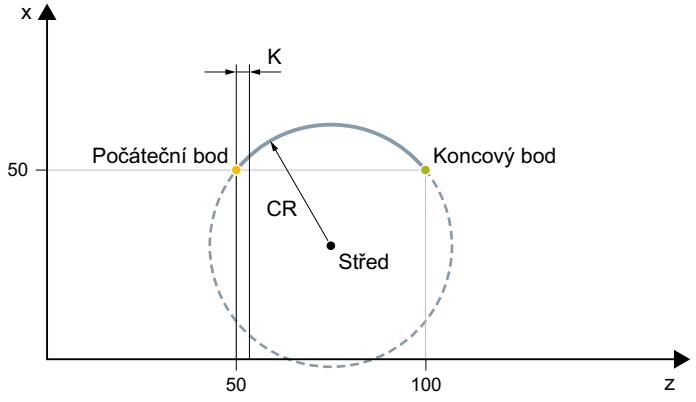


Obrázek 3-8 Sudovitý závit opracovávaný proti směru hodinových ručiček naprogramovaný pomocí koncového bodu a středu

Příklad 3: Sudovitý závit opracovávaný ve směru hodinových ručiček naprogramovaný pomocí koncového bodu a rádiusu

Programový kód

```
N5 G0 G18 X50 Z50
N10 G335 Z100 K=3.5 CR=32 SF=90
```

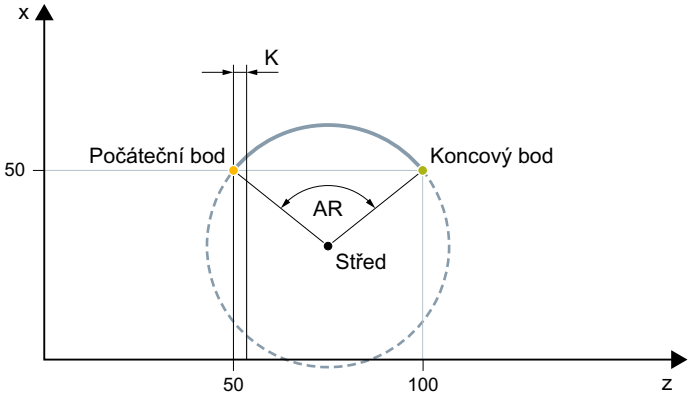


Obrázek 3-9 Sudovitý závit opracovávaný ve směru hodinových ručiček naprogramovaný pomocí koncového bodu a rádiusu

Příklad 4: Sudovitý závit opracovávaný ve směru hodinových ručiček naprogramovaný pomocí koncového bodu a úhlu kruhové výseče

Programový kód

```
N5 G0 G18 X50 Z50
N10 G335 Z100 K=3.5 AR=102.75 SF=90
```

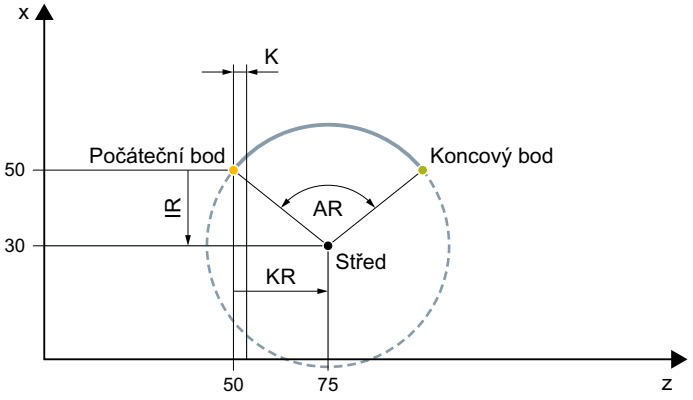


Obrázek 3-10 Sudovitý závit opracovávaný ve směru hodinových ručiček naprogramovaný pomocí koncového bodu a úhlu kruhové výseče

Příklad 5: Sudovitý závit opracovávaný ve směru hodinových ručiček naprogramovaný pomocí středu a úhlu kruhové výseče

Programový kód

```
N5 G0 G18 X50 Z50
N10 G335 K=3.5 KR=25 IR=-20 AR=102.75 SF=90
```

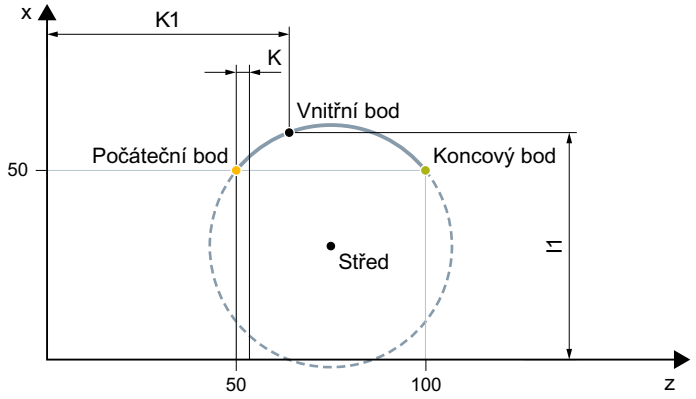


Obrázek 3-11 Sudovitý závit opracovávaný ve směru hodinových ručiček naprogramovaný pomocí středu a úhlu kruhové výseče

Příklad 6: Sudovitý závit opracovávaný ve směru hodinových ručiček naprogramovaný pomocí koncového bodu a vnitřního bodu kruhového oblouku

Programový kód

```
N5 G0 G18 X50 Z50
N10 G335 Z100 K=3.5 I1=60 K1=64
```

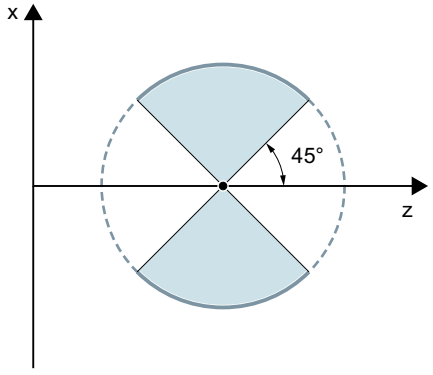


Obrázek 3-12 Sudovitý závit opracovávaný ve směru hodinových ručiček naprogramovaný pomocí koncového bodu a vnitřního bodu kruhového oblouku

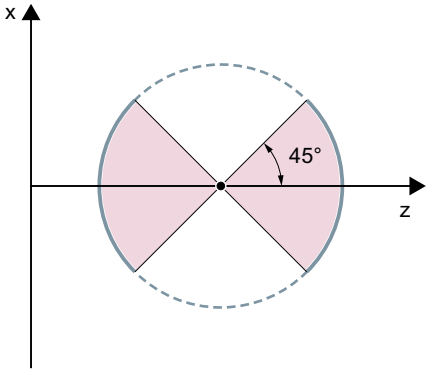
Další informace

Přípustné rozsahy kruhového oblouku

Kruhový oblouk naprogramovaný pomocí příkazů `G335`/`G336` musí ležet v rozsahu, ve kterém má specifikovaná hlavní osa závitu (`I`, `J` nebo `K`) v celém kruhovém oblouku složku hlavní osy na kruhovém oblouku:

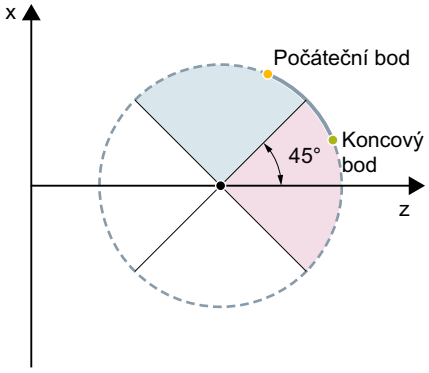


Přípustné rozsahy pro osu **Z** (stoupání naprogramováno pomocí parametru `K`)



Přípustné rozsahy pro osu **X** (stoupání naprogramováno pomocí parametru `I`)

Změna hlavní osy závitu, jak ukazuje následující obrázek, je **nepřípustná**:



Obrázek 3-13 Sudovitý závit: Přípustná oblast

Framy

Příkazy G335 a G336 je možno používat, i když jsou aktivní framy. Je ale potřeba dávat pozor, aby byly dodrženy přípustné rozsahy kruhových oblouků v základním souřadném systému (BCS).

Okrajové podmínky pro programování kruhových oblouků

Pro programování kruhových oblouků pomocí příkazů G335/G336 platí stejné okrajové podmínky, jaké jsou popsány pro programování kruhů pomocí příkazů G2/G3 (viz "Kruhová interpolace (Strana 190)").

3.9.11 Vrtání závitů bez vyrovnávací hlavičky

3.9.11.1 Vrtání závitů bez vyrovnávací hlavičky a zpětného pohybu (G331, G332)

Při vrtání závitů bez vyrovnávací hlavičky se pomocí příkazů `G331` a `G332` uskutečňují následující pohyby pracovním posuvem:

- `G331`: Vrtání závitu ve směru vyvrtané díry až do koncového bodu závitu
- `G332`: Zpětný pohyb vůči bloku pro vrtání závitu `G331` s automatickým obrácením směru otáčení vřetena

Syntaxe

```
G331 <osa> <stoupání závitu>
G331 <osa> <stoupání závitu> S...
G332 <osa> <stoupání závitu>
```

Význam

G331:	Vrtání závitů Vrtání závitu je popsáno prostřednictvím pohybu osy pracovním posuvem (vrtaná hloubka) a stoupáním závitu.	
	Platnost:	modální
G332:	Zpětný pohyb při vrtání závitu Zpětný pohyb musí být uskutečněn se stejným stoupáním jako při vrtání závitu (G331). Přepnutí směru otáčení vřetena se uskutečňuje automaticky.	
	Platnost:	modální
<osa>:	Dráha posuvu/poloha geometrické osy (X, Y nebo Z) vzhledem k počátečnímu bodu závitu, např. Z50	
<Stoupání závitu>:	Stoupání závitu I (X), J (Y) nebo K (Z): • Kladné stoupání: Pravý závit, např. K1,25 • Záporné stoupání: Levý závit, např. K-1,25	
	Rozsah hodnot:	± 0.001 až ± 2000.00 mm/otáčku
S...:	Otáčky vřetena Pokud nejsou udány žádné otáčky vřetena, bude se pohyb uskutečňovat s otáčkami vřetena, které byly v platnosti jako poslední.	

Poznámka

Datový blok druhého stupně převodovky

Aby bylo možné při vrtání závitů dosáhnout efektivního přizpůsobení otáček vřetena a momentu motoru, ale také vyššího zrychlení, může být prostřednictvím osových strojních parametrů předdefinován datový blok druhého převodového stupně pro další dvojici spínacích prahových hodnot nastavitelnou v konfiguraci (maximální a minimální rychlost). Tento blok se může lišit od datového bloku prvního převodového stupně a prahových hodnot pro přepnutí rychlosti. Je potřeba mít na paměti informace od výrobce stroje.

Pokud budete potřebovat další informace, viz Příručka k funkcím, „Osy a vřetena".

Příklady

- Příklad: Vrtání závitů pomocí příkazu G331 / G332 (Strana 244)
- Příklad: Výstup naprogramovaných otáček při vrtání s aktuálním převodovým stupněm (Strana 244)
- Příklad: Použití datového bloku druhého převodového stupně (Strana 245)
- Příklad: Žádné naprogramované otáčky, monitorování převodového stupně (Strana 245)
- Příklad: Změna převodového stupně není možná, monitorování převodového stupně (Strana 245)
- Příklad: Programování bez příkazu SPOS (Strana 246)

3.9.11.2 Příklad: Vrtání závitů pomocí příkazu G331 / G332

Programový kód	Komentář
N10 SPOS[n]=0	; vřeteno: režim regulace polohy
	; počáteční poloha 0 stupňů
N20 G0 X0 Y0 Z2	; osy: Najíždění na počáteční pozici
N30 G331 Z-50 K-4 S200	; vrtání závitu ve směru osy Z,
	; stoupání K-4 záporné =>
	; otáčení vřetena: vlevo,
	; otáčky vřetena 200 ot/min
N40 G332 Z3 K-4	; zpětný pohyb ve směru osy Z,
	; stoupání K-4 záporné (otáčení vlevo)
	; autom. obrácení směru otáčení =>
	; vřeteno se otáčí vpravo
N50 G1 F1000 X100 Y100 Z100 S300 M3	; vřeteno v režimu vřetena

3.9.11.3 Příklad: Výstup naprogramovaných otáček při vrtání s aktuálním převodovým stupněm

Programový kód	Komentář
N05 M40 S500	; naprogramované otáčky vřetena: 500 ot/min =>
	; stupeň převodovky 1 (20 až 1028 ot/min)
...	
N55 SPOS=0	; nastavení polohy vřetena
N60 G331 Z-10 K5 S800	; vrtání závitu
	; otáčky vřetena 800 ot/min => stupeň převodovky 1

Převodový stupeň v příkazu `M40`, který vyhovuje naprogramovaným otáčkám vřetena `S500`, je zjišťován na základě datového bloku prvního převodového stupně. Naprogramované otáčky pro vrtání `S800`, jsou na výstupu z momentálně zvoleného převodového stupně a v případě potřeby jsou omezeny na maximální otáčky daného převodového stupně. Po operaci s příkazem `SPOS` není automatická změna převodového stupně možná. Předpokladem pro automatickou změnu převodového stupně je, že se vřeteno nachází v režimu regulace otáček.

Poznámka

Pokud má být při otáčkách vřetena 800 ot/min zvolen 2. převodový stupeň, musí být v konfiguraci pro tento účel nastaveny spínací prahové hodnoty pro maximální a minimální otáčky v příslušných strojních parametrech datového bloku druhého převodového stupně (viz následující příklady).

3.9.11.4 Příklad: Použití datového bloku druhého převodového stupně

U příkazů G331/G332 a při naprogramování hodnoty S pro aktivní řídící vřeteno jsou vyhodnocovány spínací prahové hodnoty z datového bloku druhého převodového stupně, pokud jde o jeho maximální a minimální otáčky. Automatická změna převodového stupně M40 musí být aktivní. Takto zjištěný převodový stupeň je porovnán s momentálně zařazeným převodovým stupněm. Existuje-li mezi nimi rozdíl, provede se změna převodového stupně.

Programový kód	**Komentář**
N05 M40 S500	; naprogramované otáčky vřetena: 500 ot/min
...	
N50 G331 S800	; řídící vřeteno: Je zvolen převodový stupeň 2.
N55 SPOS=0	; nastavení polohy vřetena
N60 G331 Z-10 K5	; vrtání závitu
	; zrychlení vřetena z datového bloku 2. převodového stupně

3.9.11.5 Příklad: Žádné naprogramované otáčky, monitorování převodového stupně

Pokud je použit datový blok druhého převodového stupně a pokud v příkazu G331 nejsou naprogramovány žádné otáčky, bude se závit vyrábět s naposled naprogramovanými otáčkami. Přepínání na jiný převodový stupeň se neprovádí. V tomto případě se však kontroluje, že naprogramované otáčky leží v předem specifikovaném rozsahu (spínací mezní hodnoty pro maximální a minimální otáčky) momentálně zvoleného převodového stupně. Jinak se aktivuje alarm 16748.

Programový kód	**Komentář**
N05 M40 S800	; naprogramované otáčky vřetena: 800 ot/min
...	
N55 SPOS=0	; nastavení polohy vřetena
N60 G331 Z-10 K5	; vrtání závitu
	; monitorování otáček vřetena 800 ot/min
	; je aktivní 1. převodový stupeň
	; musel by být aktivní 2. stupeň převodovky => alarm 16748.

3.9.11.6 Příklad: Změna převodového stupně není možná, monitorování převodového stupně

Jestliže jsou v případě použití datového bloku druhého převodového stupně v příkazu G331 naprogramovány kromě geometrie ještě i otáčky vřetena a jestliže tyto otáčky neleží v předepsaném intervalu otáček (spínací mezní hodnoty pro maximální a minimální otáčky) momentálně zvoleného převodového stupně, může se stát, že se přepnutí převodového stupně neuskuteční, protože by nemohl být dodržen požadovaný pohyb vřetena a přísuvné osy po dráze.

Stejně jako v předcházejícím příkladu jsou v bloku s G331 monitorovány otáčky a převodový stupeň a v případě nutnosti se aktivuje alarm 16748.

Programový kód	**Komentář**
N05 M40 S500	; naprogramované otáčky vřetena: 500 ot/min =>
	; stupeň převodovky 1

Programový kód	Komentář
...	
N55 SPOS=0	; nastavení polohy vřetena
N60 G331 Z-10 K5 S800	; vrtání závitu
	; změna převodového stupně není možná
	; monitorování otáček vřetena 800 ot/min
	; s datovým blokem 1. převodového stupně: 2. stupeň převodovky
	; musel by být aktivní => alarm 16748

3.9.11.7 Příklad: Programování bez příkazu SPOS

Programový kód	Komentář
N05 M40 S500	; naprogramované otáčky vřetena: 500 ot/min =>
	; stupeň převodovky 1 (20 až 1028 ot/min)
...	
N50 G331 S800	; řídící vřeteno: Je zvolen převodový stupeň 2.
N60 G331 Z-10 K5	; vrtání závitu
	; zrychlení vřetena z datového bloku 2. převodového stupně

Od momentální pozice začíná pro vřeteno závitová interpolace, která závisí na předtím zpracovávaném výrobním programu, např. jestliže byla uskutečněna změna převodového stupně. Dopracování závitu v případě potřeby není z tohoto důvodu možné.

Poznámka

Je potřeba mít na paměti, že při obrábění s více vřeteny musí být vřeteno, které provádí vrtání, řídícím vřetenem. Jestliže potřebujete vrtající vřeteno nastavit jako řídící vřeteno, naprogramujte příkaz `SETMS(<číslo vřetena>)`.

3.9.12 Vrtání závitů s vyrovnávací hlavičkou

3.9.12.1 Vrtání závitů s vyrovnávací hlavičkou a se zpětným pohybem (G63)

Při vrtání závitů s vyrovnávací hlavičkou se pomocí příkazu `G63` uskutečňují následující pohyby pracovním posuvem:

- `G63`: Vrtání závitu ve směru vyvrtané díry až do koncového bodu závitu
- `G63`: Zpětný pohyb s naprogramovaným obrácením směru otáčení vřetena

Poznámka

Po bloku s příkazem `G63` se znovu aktivuje druh interpolace `G0`, `G1`, `G2`, který byl naposled v platnosti.

Syntaxe

```
G63 <osa> <směr otáčení> <otáčky> <posuv>
```

Význam

G63:	Vrtání závitů s vyrovnávací hlavičkou	
	Platnost:	bloková
<osa>:	Dráha posuvu/poloha geometrické osy (X, Y nebo Z) vzhledem k počátečnímu bodu závitu, např. Z50	
<Směr otáčení>:	Směr otáčení vřetena: • M3: Otáčení ve směru hodinových ručiček, pravý závit • M4: Otáčení proti směru hodinových ručiček, levý závit	
<Otáčky>:	Maximální otáčky vřetena, které jsou v průběhu vrtání závitu přípustné, např. S100	
<Posuv>:	Posuv F osy provádějící vrtání, přičemž F = otáčky vřetena * stoupání závitu	

Příklad:

Vrtání závitu M5:

- Stoupání závitu podle normy: 0,8 mm/ot
- Otáčky vřetena S: 200 ot/min
- Posuv F = 200 ot/min * 0,8 mm/ot = 160 mm/min.

Programový kód	**Komentář**
N10 G1 X0 Y0 Z2 F1000 S200 M3	; najetí na počáteční bod
	; vřeteno ve směru hodinových ručiček, 200 ot/min
N20 G63 Z-50 F160	; vrtání závitu s vyrovnávací hlavičkou
	; vrtaná hloubka: absolutní Z=50 mm
	; posuv: 160 mm/min
N30 G63 Z3 M4	; zpětný pohyb: absolutní Z=3 mm
	; změna směru otáčení
	; vřeteno proti směru hodinových ručiček, 200 ot/min

3.9.13 Faseta, zaoblení (CHF, CHR, RND, RNDM, FRC, FRCM)

Rohy kontury nacházející se v aktivní pracovní rovině mohou být vyhotoveny jako zaoblení nebo jako faseta.

Kvůli optimalizaci jakosti povrchu je možné pro obráběné faset/zaoblení naprogramovat samostatnou hodnotu posuvu. Jestliže žádný posuv není naprogramován, je v platnosti normální posuv po dráze F.

Pomocí funkce "Modální zaoblení" může být větší počet rohů kontury ležících za sebou zaoblen stejným způsobem.

Syntaxe

Faseta v rohu kontury:
```
G... X... Z... CHR/CHF=<hodnota> FRC/FRCM=<hodnota>
G... X... Z...
```

Zaoblení rohů kontury:
```
G... X... Z... RND=<hodnota> FRC=<hodnota>
G... X... Z...
```

Modální zaoblení:

```
G... X... Z... RNDM=<hodnota> FRCM=<hodnota>
...
RNDM=0
```

Poznámka

Technologie (posuv, typ posuvu, příkazy M-funkcí...) pro obrábění faset/rádiusů závisí na nastavení bitu 0 ve strojním parametru MD20201 $MC_CHFRND_MODE_MASK (chování na fasetě/rádiusu) a je odvozeno od chování buď v předcházejícím nebo v následujícím bloku. Při doporučeném nastavení je chování odvozeno od předcházejícího bloku (Bit 0 = 1).

Význam

CHF=... :	Faseta v rohu kontury	
	<hodnota> :	Délka fasety (měřicí jednotky podle G70/G71)
CHR=... :	Faseta v rohu kontury	
	<hodnota> :	Šířka fasety ve směru předcházejícího pohybu (měřicí jednotky podle G70/G71)
RND=... :	Zaoblení rohů kontury	
	<hodnota> :	Rádius zaoblení (měřicí jednotky podle G70/G71)
RNDM=... :	Modální zaoblení (stejným způsobem zaoblit větší počet po sobě následujících rohů kontury)	
	<hodnota> :	Rádius zaoblení (měřicí jednotky podle G70/G71)
		Příkazem RNDM=0 se modální zaoblování deaktivuje.
FRC=... :	Posuv pro obrábění faset/zaoblení s blokovou platností	
	<hodnota> :	Rychlost posuvu v mm/min (když je aktivní G94), příp. v mm/ot. (když je aktivní G95)

FRCM=... :	Posuv pro obrábění faset/zaoblení s modální platností	
	<hodnota> :	Rychlost posuvu v mm/min (když je aktivní G94), příp. v mm/ot. (když je aktivní G95)
		Pomocí příkazu FRCM=0 se posuv pro obrábění faset/zaoblení s modální platností deaktivuje, takže bude v platnosti posuv naprogramovaný pomocí příkazu F.

Poznámka

Faseta/zaoblení jsou příliš veliké

Pokud jsou naprogramované hodnoty pro fasetu (CHF/CHR) nebo zaoblení (RND/RNDM) příliš velké vzhledem k připojovaným konturovým prvkům, bude se faseta nebo zaoblení upravovat:

1. Jestliže je aktivován bit 4 parametru MD11411 $MN_ENABLE_ALARM_MASK, bude se aktivovat alarm 10833 „Faseta nebo rádius musí být zmenšeny" (alarm typu Cancel).
2. Faseta/zaoblení se zmenší natolik, aby byly přizpůsobeny rohu kontury. Přitom vznikne minimálně jeden blok neobsahující žádný pohyb. V tomto bloku se pohyb nutně zastaví.

Poznámka

Faseta/zaoblení nejsou možné

Nastane-li některá z následujících okolností, faseta/zaoblení se nebudou vkládat:

- V rovině není k dispozici žádná přímková nebo kruhová kontura.
- Pohyb se uskutečňuje mimo rovinu.
- Bylo provedeno přepnutí roviny.
- Došlo k překročení ve strojním parametru definovaného počtu bloků, které neobsahují žádné informace o pohybu (např. pouze příkazový výstup).

Poznámka

FRC/FRCM

FRC/FRCM se neuplatňují, pokud má být faseta vyrobena pomocí G0; při programování je možné používat jen odpovídající F-slovo bez chybového hlášení.

FRC má platnost jen tehdy, je-li v bloku, kde je naprogramována faseta/zaoblení, příp. když bylo aktivováno RNDM.

FRC přepisuje v aktuálním bloku hodnotu F, příp. hodnotu FRCM.

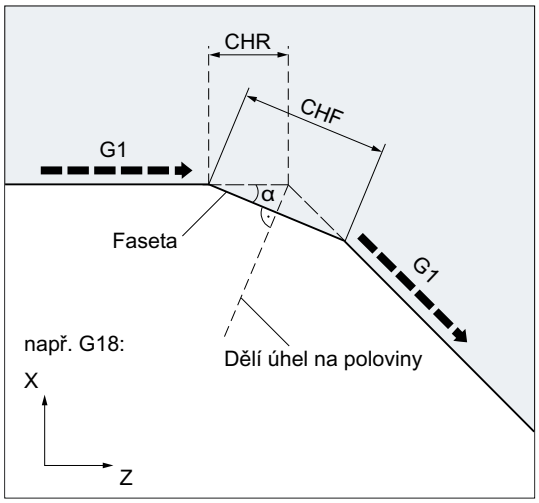
Posuv naprogramovaný do FRC musí být větší než nula.

Příkaz FRCM=0 aktivuje obrábění faset/zaoblení s posuvem naprogramovaným pomocí příkazu F.

Pokud je naprogramováno FRCM, musí být znovu naprogramována hodnota tohoto příkazu, analogicky k F, když dojde k přepnutí G94 <--> G95 atd. Pokud je naprogramována pouze nová hodnota F a hodnota FRCM byla před změnou typu posuvu > 0, bude aktivováno chybové hlášení.

Příklady

Příklad 1: Faseta mezi dvěma přímkami



- Bit 0 v MD20201 = 1 (chování je odvozováno z předcházejícího bloku)
- G71 je aktivní.
- Šířka fasety ve směru pohybu (CHR) by měla být 2 mm, posuv pro obrábění fasety činí 100 mm/min.

Programování může být realizováno dvěma způsoby:

- Programování pomocí příkazu CHR

Programový kód

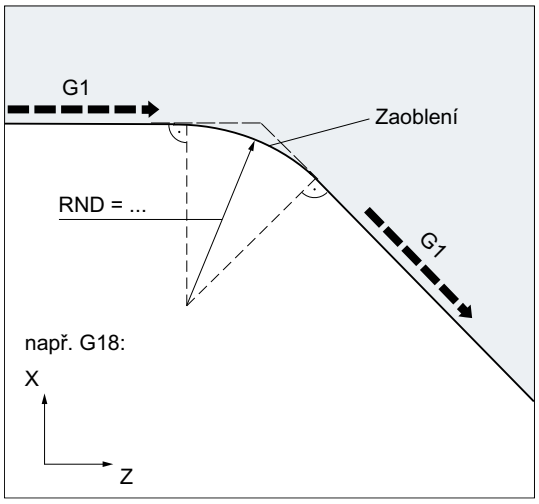
```
...
N30 G1 Z… CHR=2 FRC=100
N40 G1 X…
...
```

- Programování pomocí příkazu CHF

Programový kód

```
...
N30 G1 Z… CHF=2(cosα*2) FRC=100
N40 G1 X…
...
```

Příklad 2: Zaoblení mezi dvěma přímkami



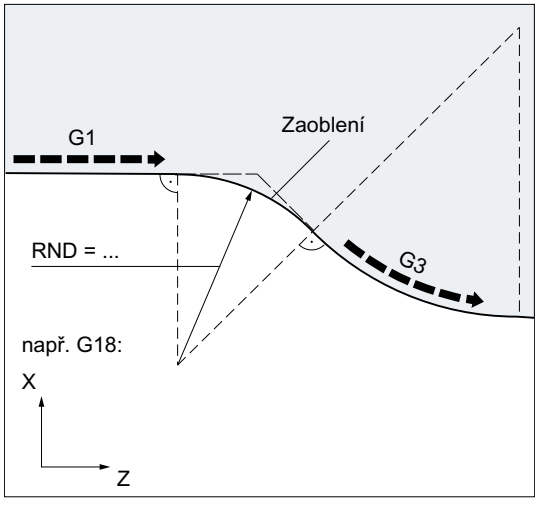
- Bit 0 v MD20201 = 1 (chování je odvozováno z předcházejícího bloku)
- G71 je aktivní.
- Rádius zaoblení má být 2 mm, posuv pro obrábění zaoblení činí 50 mm/min.

Programový kód

```
...
N30 G1 Z… RND=2 FRC=50
N40 G1 X…
...
```

Příklad 3: Zaoblení mezi přímkou a kruhovým obloukem

Mezi lineární a kruhové konturové prvky v libovolné kombinaci lze pomocí funkce RND vložit kruhový konturový prvek s tangenciálním napojením.



- Bit 0 v MD20201 = 1 (chování je odvozováno z předcházejícího bloku)
- G71 je aktivní.
- Rádius zaoblení má být 2 mm, posuv pro obrábění zaoblení činí 50 mm/min.

Programový kód

```
...
```

Programový kód
N30 G1 Z… RND=2 FRC=50
N40 G3 X… Z… I… K…
...

Příklad 4: Modální zaoblení kvůli skosení ostrých hran obrobku

Programový kód	**Komentář**
...	
N30 G1 X… Z… RNDM=2 FRCM=50	; aktivování modálního zaoblení
	Rádius zaoblení: 2mm
	Posuv pro zaoblení: 50 mm/min
N40...	
N120 RNDM=0	; deaktivování modálního zaoblení
...	

Příklad 5: Přebírání technologie z následujícího nebo z předcházejícího bloku

- MD20201 Bit 0 = 0: Chování je odvozováno od následujícího bloku (standardní nastavení!)

Programový kód	**Komentář**
N10 G0 X0 Y0 G17 F100 G94	
N20 G1 X10 CHF=2	; faseta N20-N30 s F=100 mm/min
N30 Y10 CHF=4	; faseta N30-N40 s FRC=200 mm/min
N40 X20 CHF=3 FRC=200	; faseta N40-N60 s FRCM=50 mm/min
N50 RNDM=2 FRCM=50	
N60 Y20	; modální zaoblení N60-N70 s FRCM=50 mm/min
N70 X30	; modální zaoblení N70-N80 s FRCM=50 mm/min
N80 Y30 CHF=3 FRC=100	; faseta N80-N90 s FRC=100 mm/min
N90 X40	; modální zaoblení N90-N100 s F=100 mm/min (FRCM deaktivováno)
N100 Y40 FRCM=0	; modální zaoblení N100-N120 s G95 FRC=1 mm/ot
N110 S1000 M3	
N120 X50 G95 F3 FRC=1	
...	
M02	

- MD20201 Bit 0 = 1: Chování odvozeno od předcházejícího bloku (doporučené nastavení!)

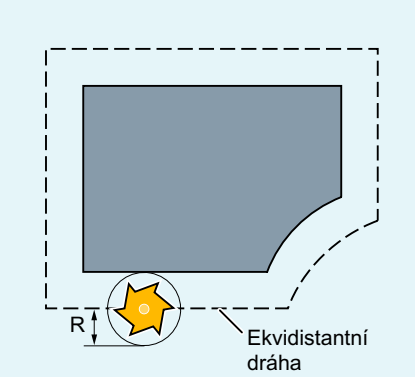
Programový kód	**Komentář**
N10 G0 X0 Y0 G17 F100 G94	
N20 G1 X10 CHF=2	; faseta N20-N30 s F=100 mm/min
N30 Y10 CHF=4 FRC=120	; faseta N30-N40 s FRC=120 mm/min
N40 X20 CHF=3 FRC=200	; faseta N40-N60 s FRC=200 mm/min

Programový kód	**Komentář**
N50 RNDM=2 FRCM=50	
N60 Y20	; modální zaoblení N60-N70 s FRCM=50 mm/min
N70 X30	; modální zaoblení N70-N80 s FRCM=50 mm/min
N80 Y30 CHF=3 FRC=100	; faseta N80-N90 s FRC=100 mm/min
N90 X40	; modální zaoblení N90-N100 s FRCM=50 mm/min
N100 Y40 FRCM=0	; modální zaoblení N100-N120 s F=100 mm/min
N110 S1000 M3	
N120 X50 CHF=4 G95 F3 FRC=1	; faseta N120-N130 s G95 FRC=1 mm/ot
N130 Y50	; modální zaoblení N130-N140 s F=3 mm/ot
N140 X60	
...	
M02	

3.10 Korekce rádiusu nástroje

3.10.1 Korekce rádiusu nástroje (G40, G41, G42, OFFN)

Když je korekce rádiusu nástroje aktivována, řídící systém automaticky vypočítává pro rozmanité nástroje příslušné ekvidistantní dráhy.



Syntaxe

G0/G1 X... Y… Z... **G41**/**G42** [**OFFN=<hodnota>**]	
...	
G40 X... Y… Z...	

Význam

G41:	Aktivování korekce rádiusu nástroje při směru opracovávání **vlevo** od kontury
G42:	Aktivování korekce rádiusu nástroje při směru opracovávání **vpravo** od kontury
OFFN=<hodnota>:	Přídavek rozměru k naprogramované kontuře (normální offset kontury) (zadání není nutné) Např . výroba ekvidistantních drah pro obrábění nahrubo.
G40:	Deaktivování korekce rádiusu nástroje

Poznámka

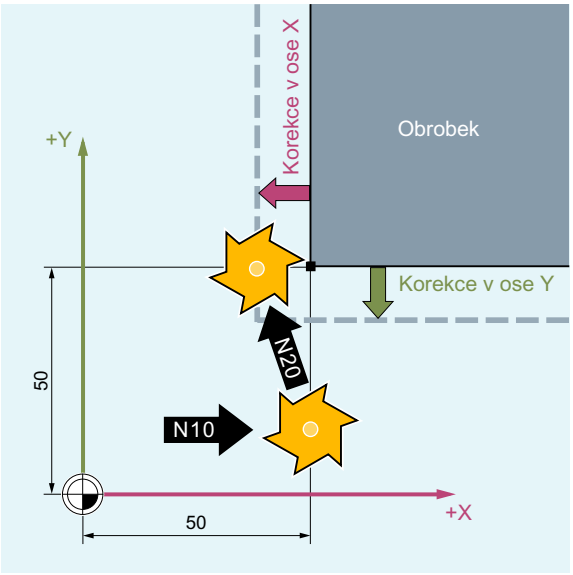
V NC bloku s příkazem G40/G41/G42 musí být aktivní příkaz G0 nebo G1 a musí být uvedena nejméně jedna osa ze zvolené pracovní roviny.

Jestliže je při aktivování zadána jen jedna osa, poslední pozice druhé osy bude automaticky doplněna a budou se pohybovat **obě** osy.

Obě osy musí být v kanálu aktivovány jako geometrické osy, což může být zabezpečeno tím, že budou naprogramovány pomocí příkazu GEOAX.

Příklady

Příklad 1: Frézování



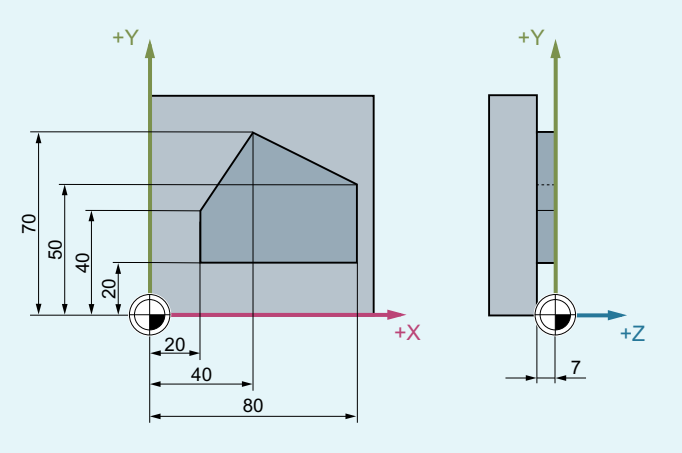
Programový kód	Komentář
N10 G0 X50 T1 D1	; bude aktivována pouze korekce délky nástroje Pohyb na X50 se uskuteční bez korekce.
N20 G1 G41 Y50 F200	; korekce rádiusu nástroje se aktivuje, takže na bod X50/Y50 se najíždí s korekcí
N30 Y100	

Programový kód	Komentář
...	

Příklad 2: "Klasický" postup na příkladu frézování

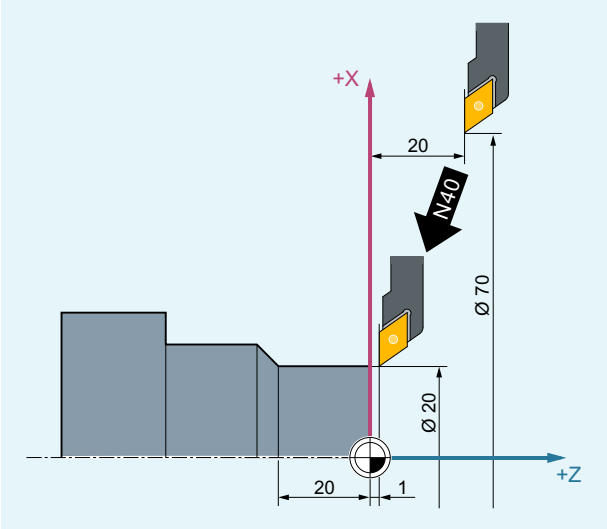
"Klasický" postup:

1. Volání nástroje
2. Výměna nástroje.
3. Aktivování pracovní roviny a korekce rádiusu nástroje.



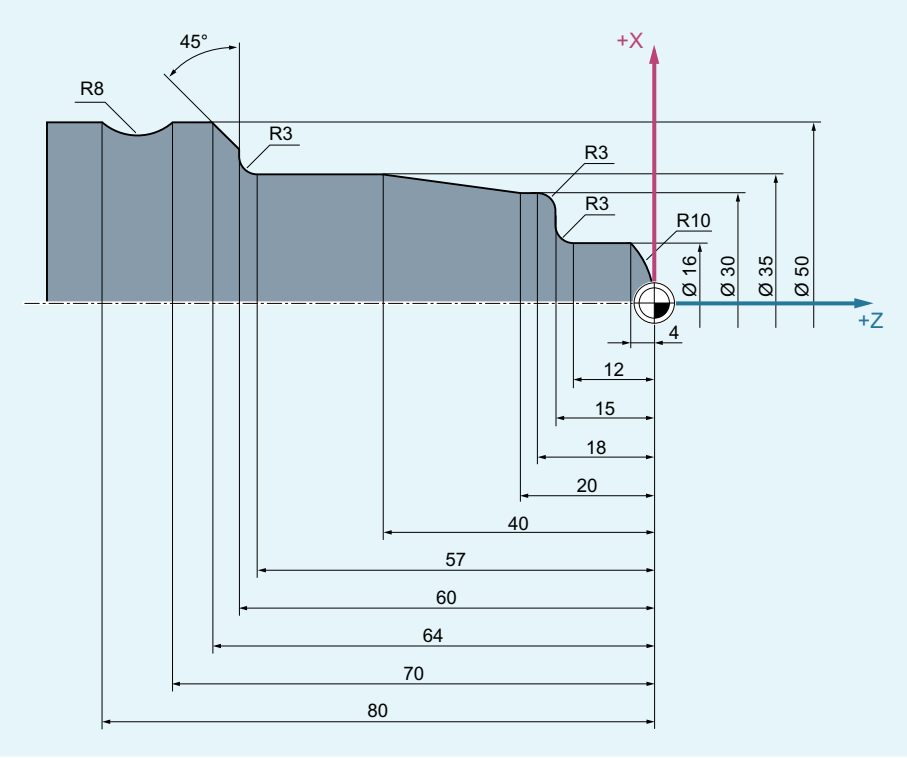
Programový kód	Komentář
N10 G0 Z100	; volné najíždění na bod pro výměnu nástroje
N20 G17 T1 M6	; výměna nástroje
N30 G0 X0 Y0 Z1 M3 S300 D1	; vyvolání korekčních hodnot parametrů nástroje, aktivování korekce délky
N40 Z-7 F500	; přísuv nástroje
N50 G41 X20 Y20	; aktivování korekce rádiusu nástroje, nástroj pracuje vlevo od kontury
N60 Y40	; frézování kontury
N70 X40 Y70	
N80 X80 Y50	
N90 Y20	
N100 X20	
N110 G40 G0 Z100 M30	; vyjíždění nástroje, konec programu

Příklad 3: Soustružení



Programový kód	Komentář
…	
N20 T1 D1	; bude aktivována pouze korekce délky nástroje
N30 G0 X70 Z20	; na bod X70 Z20 se najíždí bez korekce
N40 G42 X20 Z1	; aktivuje se korekce rádiusu nástroje, na bod X20 Z1 se najíždí s korekcí
N50 G1 Z-20 F0.2	
…	

Příklad 4: Soustružení



Programový kód	**Komentář**
N5 G0 G53 X280 Z380 D0	; počáteční bod
N10 TRANS X0 Z250	; posunutí počátku
N15 LIMS=4000	; omezení otáček (G96)
N20 G96 S250 M3	; aktivování konstantního posuvu
N25 G90 T1 D1 M8	; volba nástroje a aktivování korekčních parametrů
N30 G0 G42 X-1.5 Z1	; nastavení nástroje do potřebné polohy s korekcí rádiusu nástroje
N35 G1 X0 Z0 F0.25	
N40 G3 X16 Z-4 I0 K-10	; soustružení rádiusu 10
N45 G1 Z-12	
N50 G2 X22 Z-15 CR=3	; soustružení rádiusu 3
N55 G1 X24	
N60 G3 X30 Z-18 I0 K-3	; soustružení rádiusu 3
N65 G1 Z-20	
N70 X35 Z-40	
N75 Z-57	
N80 G2 X41 Z-60 CR=3	; soustružení rádiusu 3
N85 G1 X46	
N90 X52 Z-63	
N95 G0 G40 G97 X100 Z50 M9	; deaktivování korekce rádiusu nástroje a najetí na bod pro výměnu nástroje

Programový kód	Komentář
N100 T2 D2	; vyvolání nástroje a korekčních parametrů
N105 G96 S210 M3	; aktivování konstantní řezné rychlosti
N110 G0 G42 X50 Z-60 M8	; nastavení nástroje do potřebné polohy s korekcí rádiusu nástroje
N115 G1 Z-70 F0.12	; soustružení průměru 50
N120 G2 X50 Z-80 I6.245 K-5	; soustružení rádiusu 8
N125 G0 G40 X100 Z50 M9	; pozvednutí nástroje a deaktivování korekce rádiusu nástroje
N130 G0 G53 X280 Z380 D0 M5	; najetí na bod pro výměnu nástroje
N135 M30	; konec programu

Další informace

Výpočet dráhy nástroje

Pro výpočet dráhy nástroje potřebuje řídící systém následující informace:

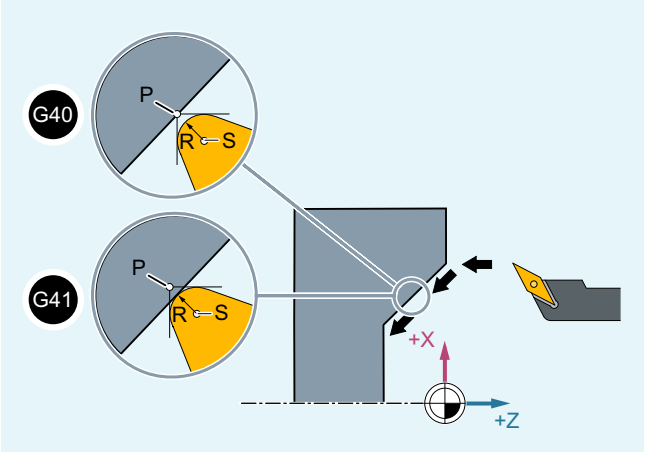
- Číslo nástroje (T...), číslo břitu (D...)
- Směr obrábění (G41/G42)
- Pracovní rovina (G17/G18/G19)

Číslo nástroje (T...), číslo břitu (D...)

Z rádiusu frézy, resp. břitu, a údajů o poloze břitu bude vypočítána vzdálenost mezi dráhou nástroje a konturou obrobku.

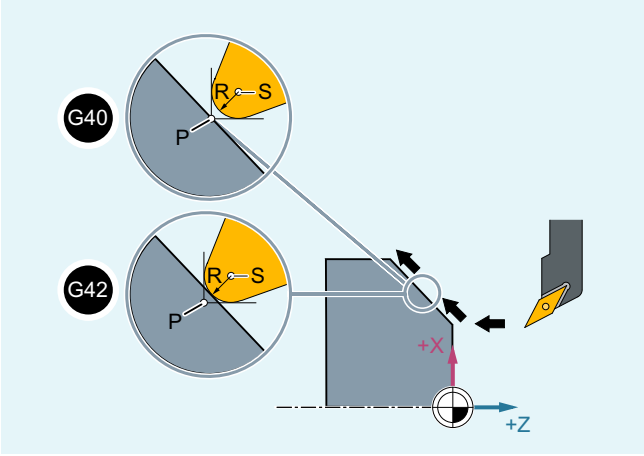
Směr obrábění (G41/G42)

Z tohoto údaje řídící systém rozpozná směr, ve kterém má být dráha nástroje posunuta.



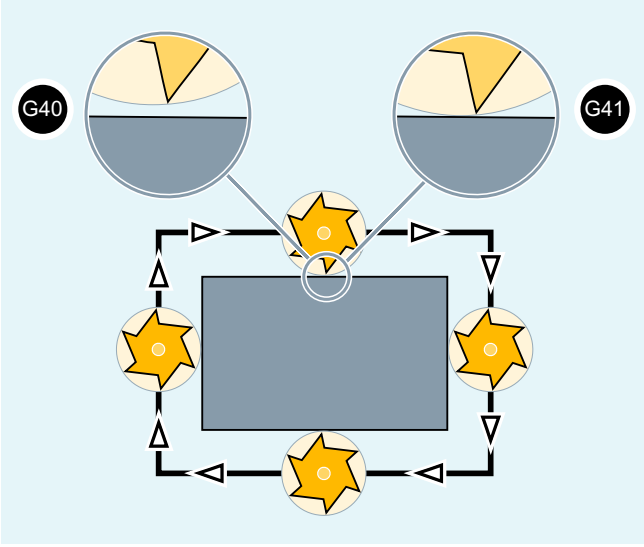
P Špička nástroje
S Střed břitu
R Rádius břitu

Obrázek 3-14 Korekce rádiusu nástroje při soustružení se směrem opracovávání vlevo od kontury (G41)

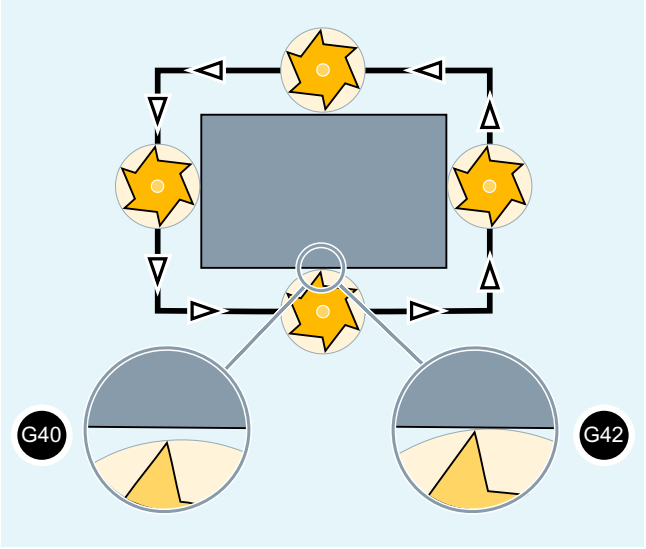


P Špička nástroje
S Střed břitu
R Rádius břitu

Obrázek 3-15 Korekce rádiusu nástroje při soustružení se směrem opracovávání vpravo od kontury (G42)



Obrázek 3-16 Korekce rádiusu nástroje při frézování se směrem opracovávání vlevo od kontury (G41)



Obrázek 3-17 Korekce rádiusu nástroje při frézování se směrem opracovávání vpravo od kontury (G42)

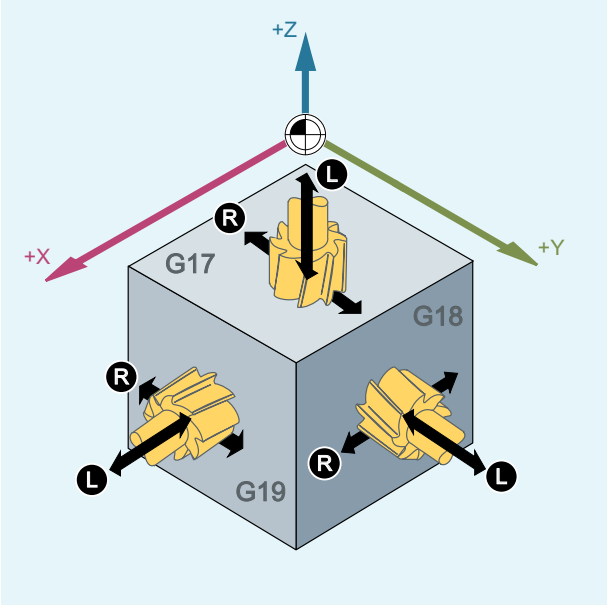
Poznámka

Záporná hodnota korekce má stejný význam jako změna strany pro korekci (G41 <---> G42).

Pracovní rovina (G17/G18/G19)

Z toho řídící systém rozpozná rovinu a tedy i osové směry, ve kterých má být korekce aplikována.

Příklad: fréza



L Délka

R Rádius

Programový kód	Komentář
...	
N10 G17 G41 …	; Korekce rádiusu nástroje se provádí v rovině X/Y, korekce délky nástroje bude aplikována v ose Z.
...	

Poznámka

U strojů se dvěma osami je možná korekce rádiusu nástroje pouze ve „skutečné" rovině, zpravidla G18.

Korekce délky nástroje

Parametr opotřebení přiřazený při volbě nástroje ose, ve které se měří průměr, může být pomocí strojního parametru definován jako hodnota průměru. Toto přiřazení se nemění automaticky, když je následně změna rovina. Aby se změna uskutečnila, po změně roviny je nutno nástroj znovu vybrat.

Chování korekce rádiusu nástroje v případě uzavřené kontury

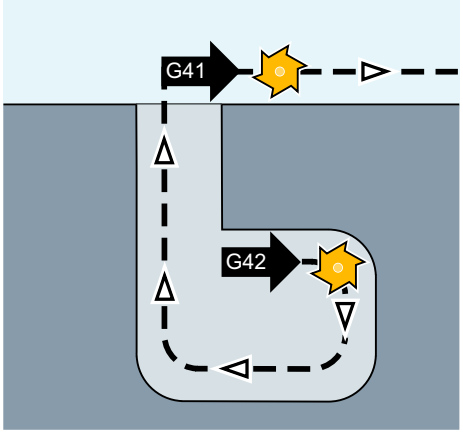
Chování korekce rádiusu nástroje v případě uzavřené kontury je možno definovat prostřednictvím nastavovaného parametru:

SD42496 $SC_CUTCOM_CLSD_CONT

Hodnota	Význam
FALSE	Jestliže u nějaké (skoro) uzavřené kontury, která se skládá ze dvou po sobě jdoucích kruhových bloků nebo z kruhového a lineárního bloku, existují při korekci na vnitřní straně dva průsečíky, pak bude v souladu se standardním postupem zvolen ten průsečík, který leží na první části kontury blíž ke konci bloku. Kontura je považována za (skoro) uzavřenou, jestliže vzdálenost mezi počátečním bodem prvního bloku a koncovým bodem druhého bloku je menší než 10% efektivního rádiusu korekce, ale ne větší než 1000 dráhových inkrementů (odpovídá 1 mm při 3 desetinných místech).
TRUE	Ve stejné situaci, jaká byla popsána výše, bude zvolen průsečík, který leží na první části kontury blíž k začátku bloku.

Změna směru korekce (G41 ↔ G42)

Změna směru korekce (G41 <--> G42) může být naprogramována, aniž by bylo potřeba mezi ně vložit jejich deaktivování pomocí příkazu G40.



Změna pracovní roviny

Když je aktivní příkaz G41/G42, změna pracovní roviny (G17/G18/G19) **není** možná.

Změna datového bloku korekčních parametrů nástroje (D...)

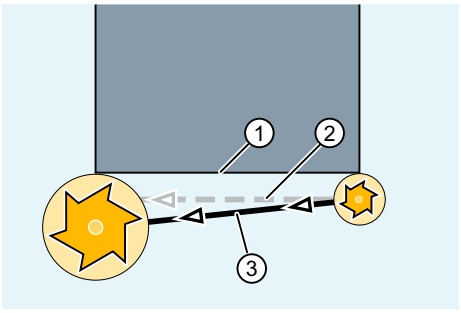
Datový blok korekčních parametrů nástroje může být v režimu práce s korekcí změněn.

Změněný rádius nástroje platí teprve od bloku, v němž se nachází nové D-číslo.

Poznámka

Změna rádiusu, příp. vyrovnávací pohyb, se uskutečňuje přes celý blok a nové ekvidistantní dráhy je dosaženo teprve v naprogramovaném koncovém bodu.

V případě lineárních pohybů se nástroj pohybuje po šikmé dráze mezi počátečním a koncovým bodem:



① Naprogramovaná kontura
② Dráha nástroje beze změny datového bloku korekčních parametrů nástroje v aktivním bloku
③ Dráha nástroje se změnou datového bloku korekčních parametrů nástroje v aktivním bloku a s tím spojená změna rádiusu nástroje

V případě kruhové interpolace vznikají spirální pohyby.

Změna rádiusu nástroje

Změna může být provedena např. pomocí systémových proměnných. Pro zpracování programu platí totéž jako při změně datového bloku korekčních parametrů nástroje (D...).

Poznámka

Změněné hodnoty budou v platnosti až po opětovném naprogramování T-čísla nebo D-čísla. Změna bude platit až v následujícím bloku.

Když je aktivováno započítávání korekcí

Režim započítávání korekcí smí být přerušen pouze určitým počtem po sobě jdoucích bloků nebo M-příkazů, které neobsahují žádné příkazy posuvu, příp. údaje dráhy v rovině korekce.

Poznámka

Maximální počet po sobě jdoucích bloků nebo příkazů M-funkcí znamenajících přerušení je nastavitelný pomocí strojního parametru (viz informace od výrobce stroje!).

Poznámka

Blok s dráhou posuvu rovnou nule se rovněž počítá jako přerušení!

Viz také

Najíždění na konturu a odjíždění od ní (NORM, KONT, KONTC, KONTT) (Strana 263)

3.10.2 Najíždění na konturu a odjíždění od ní (NORM, KONT, KONTC, KONTT)

Předpoklad

Příkazy `KONTC` a `KONTT` jsou k dispozici jen tehdy, pokud je v řídícím systému uvolněn volitelný doplněk "Polynomická interpolace".

Funkce

Prostřednictvím příkazů `NORM`, `KONT`, `KONTC` nebo `KONTT` je možné při aktivované korekci rádiusu nástroje (`G41`/`G42`) přizpůsobit dráhu pro najíždění a odjíždění nástroje na požadovanou konturu nebo na tvar surového obrobku.

Pokud jsou použity příkazy `KONTC` nebo `KONTT`, zůstávají podmínky spojitosti ve všech třech osách zachovány. Díky tomu je přípustné současné naprogramování komponent dráhy, které jsou kolmé na rovinu korekce.

Syntaxe

G41/G42 **NORM**/**KONT**/**KONTC**/**KONTT** X... Y... Z...	
...	
G40 X... Y... Z...	

Význam

NORM:	Aktivování přímého najíždění/odjíždění po přímce Nástroj se nastavuje kolmo na bod kontury.
KONT:	Aktivování najíždění/odjíždění s objížděním počátečního/koncového bodu podle naprogramovaného chování v rozích pomocí příkazů G450, příp. G451.
KONTC:	Aktivování najíždění/odjíždění s dodržením spojitého zakřivení
KONTT:	Aktivování tangenciálního najíždění/odjíždění

Poznámka

Jako původní bloky pro najíždění/odjíždění pro funkce KONTC a KONTT jsou přípustné pouze bloky s G1. Tyto bloky jsou řídícím systémem nahrazovány polynomem pro odpovídající najížděcí/odjížděcí dráhu.

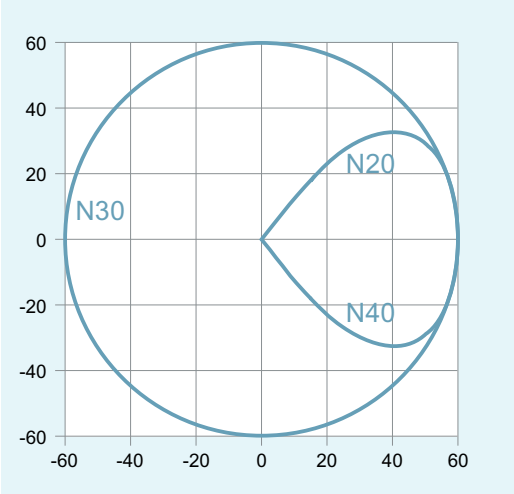
Okrajové podmínky

Příkazy KONTC a KONTT nejsou k dispozici při 3D variantách korekce rádiusu nástroje (CUT3DC, CUT3DCC, CUT3DF). Jestliže jsou přesto naprogramovány, uskuteční se interní přepnutí řídícím systémem na režim NORM.

Příklad:

KONTC

Má se najíždět na plnou kružnici, přičemž počáteční bod leží v jejím středu. Směr a rádius zakřivení najížděcí dráhy jsou v jejím koncovém bloku identické s těmito hodnotami pro navazující kruhový oblouk. Současně se uskutečňuje přísuv v ose Z jak v bloku najíždění, tak i v bloku odjíždění. Následující obrázek ukazuje vertikální projekci této dráhy nástroje.

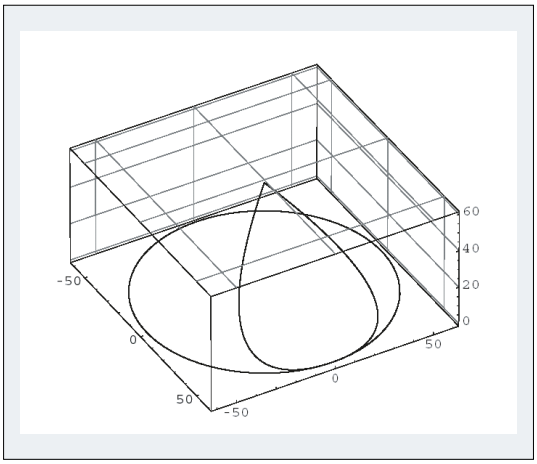


Obrázek 3-18 Kolmá projekce

Segment souvisejícího NC programu je následující:

Programový kód	**Komentář**
`$TC_DP1[1,1]=121`	`; fréza`
`$TC_DP6[1,1]=10`	`; rádius 10 mm`
`N10 G1 X0 Y0 Z60 G64 T1 D1 F10000`	
`N20 G41 KONTC X70 Y0 Z0`	`; najíždění`
`N30 G2 I-70`	`; celá kružnice`
`N40 G40 G1 X0 Y0 Z60`	`; odjíždění`
`N50 M30`	

Souběžně s přizpůsobováním zakřivení na kruhovou dráhu celé kružnice se provádí přísuv z výšky Z60 na rovinu kružnice Z0:

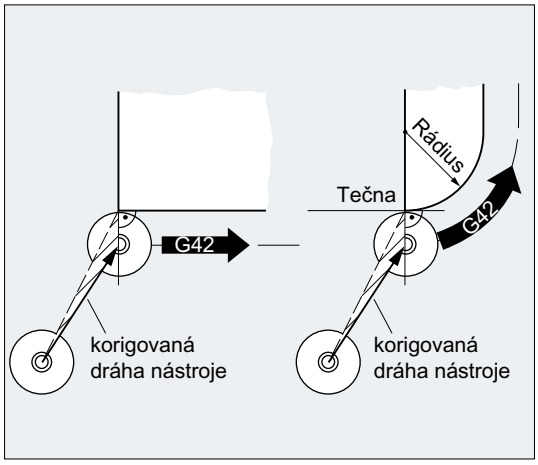


Obrázek 3-19 Prostorové zobrazení

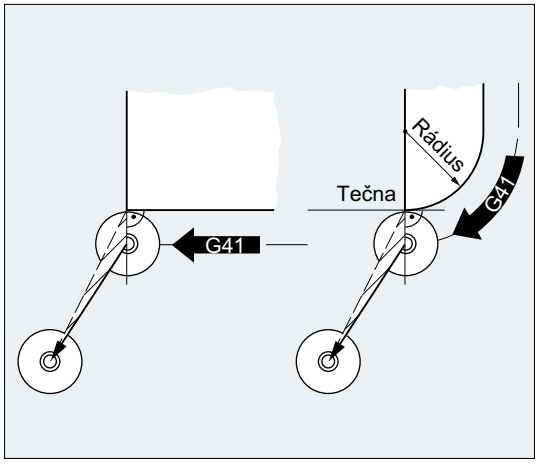
Další informace

Najíždění/odjíždění s příkazem NORM

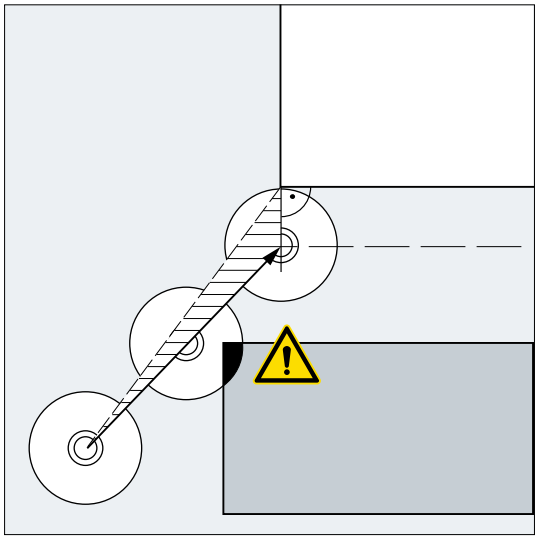
1. Najíždění:
   Když je aktivován příkaz NORM, pohybuje se nástroj rovně po přímce na počáteční pozici upravenou o korekci (bez ohledu na úhel najíždění zadaný dříve pomocí naprogramovaného pracovního pohybu) a nastaví se kolmo na tečnu ke dráze v tomto počátečním bodě.



2. Odjíždění:
   Nástroj se nachází kolmo na poslední koncový bod dráhy s korekcí a pak se pohybuje (bez ohledu na úhel najíždění zadaný dříve pomocí naprogramovaného pracovního pohybu) rovně po přímce na následující pozici bez korekce, např. na bod pro výměnu nástroje.



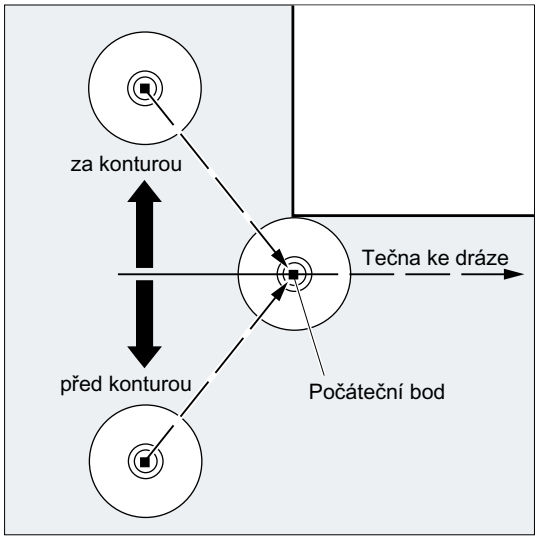
Změna úhlu najíždění/odjíždění představuje nebezpečí kolize:



UPOZORNĚNÍ
Nebezpečí kolize Aby se zabránilo případným kolizím, je nutno mít při programování na paměti změněný úhel najíždění/odjíždění.

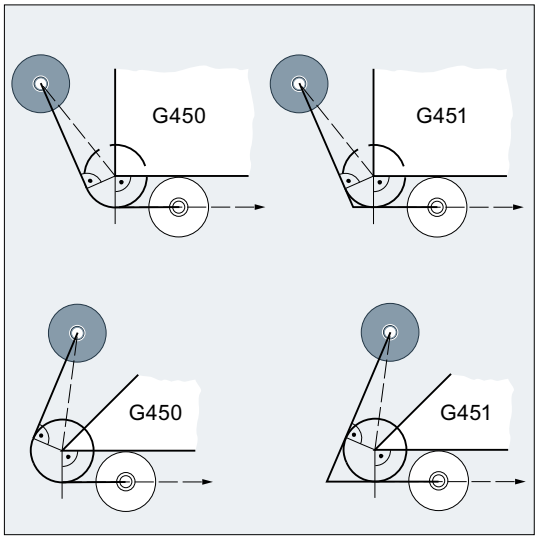
Najíždění/odjíždění s příkazem KONT

Před najížděním se nástroj může nacházet **před** nebo **za** konturou. Jako dělicí čára přitom platí tečna ke dráze v počátečním bodě:

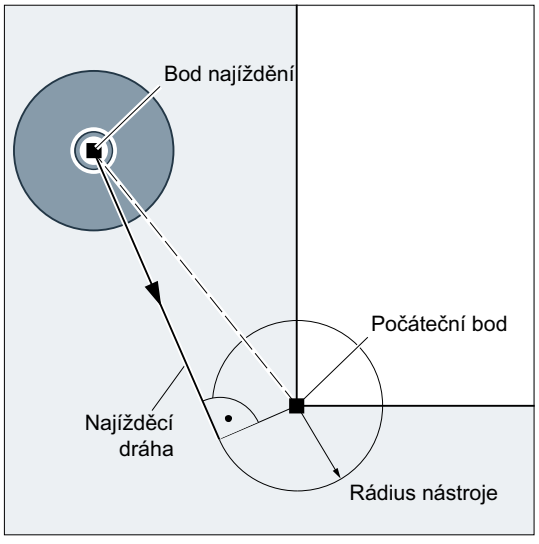


Podle toho je zapotřebí při najíždění/odjíždění pomocí příkazu KONT rozlišovat následující dva případy:

1. Nástroj se nalézá před konturou
   --> Strategie najíždění/odjíždění je stejná jako u příkazu NORM.
2. Nástroj se nalézá za konturou
   – Najíždění:
   Nástroj objíždí počáteční bod v závislosti na naprogramovaném chování v rohu (G450/G451) po kruhové dráze nebo přes průsečík ekvidistantních drah.
   Příkazy G450/G451 platí pro přechod z aktuálního bloku na příští blok:



V obou případech (G450/G451) bude vytvořena následující najížděcí dráha:



Z najížděcího bodu bez korekce bude vztyčena přímka, která se dotýká kružnice, jejíž poloměr je roven rádiusu nástroje. Střed této kružnice leží v počátečním bodě.

– Odjíždění:
Pro odjíždění platí v obráceném pořadí totéž co pro najíždění.

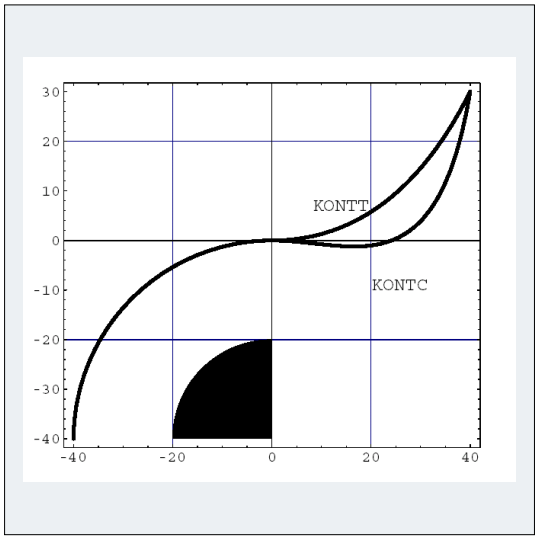
Najíždění/odjíždění s příkazem KONTC

Na bod kontury se najíždí nebo se od něj odjíždí po spojité křivce. Na bodu kontury se nevyskytuje žádné skokové zrychlení. Dráha z počátečního bodu na konturu je interpolována jako polynom.

Najíždění/odjíždění s příkazem KONTT

Na bod kontury se najíždí nebo se od něj odjíždí po spojité tečně. Na bodu kontury se může vyskytnout skokové zrychlení. Dráha z počátečního bodu na konturu je interpolována jako polynom.

Rozdíl mezi funkcemi KONTC a KONTT

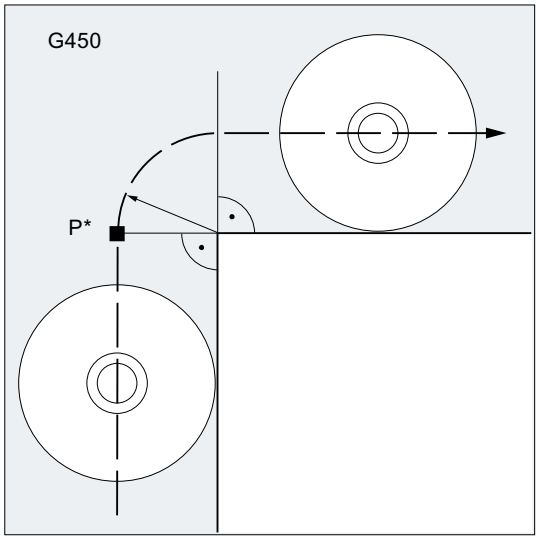


Na tomto obrázku je zobrazeno odlišné chování při najíždění/odjíždění při příkazech `KONTT` a `KONTC`. Provádí se korekce na vnější straně na kruhu s rádiusem 20 mm okolo středu v bodě X0 Y-40 s nástrojem o rádiusu 20 mm. Výsledkem tedy je kruhový pohyb středu nástroje s rádiusem 40 mm. Koncový bod odjížděcího bloku leží v bodě X40 Y30. Přechod mezi blokem kruhu a odjížděcím blokem leží v počátku. Kvůli prodloužení spojitého zakřivení v případě příkazu `KONTC` provádí odjížděcí blok napřed pohyb se zápornou složkou Y, což je často nežádoucí. Odjížděcí blok s `KONTT` toto chování nevykazuje. Jinak se v tomto případě na přechodu mezi bloky vyskytne skoková změna zrychlení.

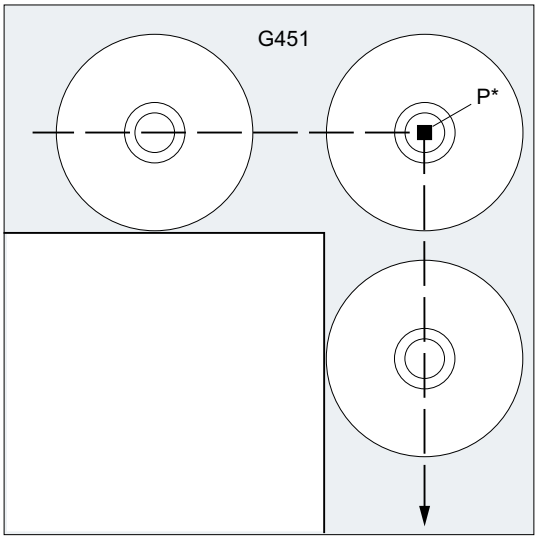
Jestliže je blok s `KONTT`, příp. `KONTC` nikoli blokem pro odjíždění, ale blokem pro najíždění, je výsledná kontura přesně stejná, pohyb se však uskutečňuje v opačném směru.

3.10.3 Korekce na vnějších rozích (G450, G451, DISC)

Prostřednictvím příkazů G450, příp. G451, je při aktivované korekci rádiusu nástroje (G41/G42) definován průběh korigované dráhy nástroje při objíždění vnějších rohů:



Když je použit příkaz G450, objíždí střed nástroje roh obrobku po kruhovém oblouku, jehož poloměr odpovídá rádiusu nástroje.



Když je použit příkaz G451, najíždí nástroj na průsečík obou ekvidistantních drah, které leží ve vzdálenosti rádiusu nástroje k naprogramované kontuře. G451 platí jen pro přímkové a kruhové úseky.

Poznámka

Pomocí příkazů G450/G451 je definována také najížděcí dráha, když je aktivní příkaz KONT a když se najížděcí bod nachází za konturou (viz "Najíždění na konturu a odjíždění od ní (NORM, KONT, KONTC, KONTT) (Strana 263)").

Pomocí příkazu DISC mohou být přechodové kružnice, jestliže je aktivní příkaz G450, deformovány, takže lze vyrábět ostré konturové rohy.

Syntaxe

```
G450 [DISC=<hodnota>]
G451
```

Význam

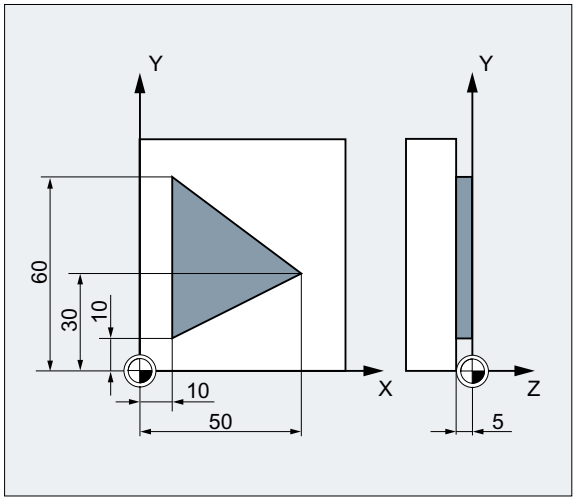
G450:	Pomocí příkazu G450 jsou rohy obrobku objížděny po kruhové dráze.			
DISC:	Flexibilní programování kruhové dráhy, když je aktivní příkaz G450 (zadání není nutné)			
	<hodnota>:	Typ:	INT	
		Rozsah hodnot:	0, 1, 2, ... 100	
		Význam:	0	Přechodový kruh
			100	Průsečík ekvidistantních drah (teoretická hodnota)
G451:	Když je zadán příkaz G451, bude se na rozích kontury najíždět na průsečík obou ekvidistantních křivek. Na rozích obrobku nástroj tedy řeže ve volném prostoru.			

Poznámka

Příkaz DISC se uplatňuje jen spolu s voláním příkazu G450, je možné jej však naprogramovat i v předcházejícím příkazu bez G450. Oba příkazy mají modální působnost.

Příklad:

V následujícím příkladu se na všech vnějších rozích vkládá jako přechodový prvek rádius (v souladu s naprogramovaným chováním v rozích v bloku N30). Tato zabraňujete situaci, kdy se nástroj musí kvůli změně směru zastavit a řezat naprázdno.



Programový kód	**Komentář**
N10 G17 T1 G0 X35 Y0 Z0 F500	; počáteční podmínky
N20 G1 Z-5	; přísuv nástroje
N30 G41 KONT **G450** X10 Y10	; aktivování korekce rádiusu nástroje a režimu najíždění/odjíždění KONT a **chování v rozích G450.**
N40 Y60	; frézování kontury
N50 X50 Y30	
N60 X10 Y10	

Programový kód	Komentář
N80 G40 X-20 Y50	; deaktivování režimu práce s korekcí, odjíždění po přechodové kružnici
N90 G0 Y100	
N100 X200 M30	

Další informace

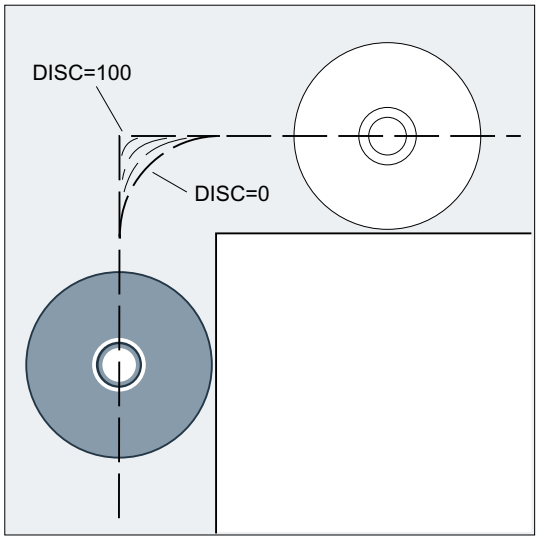
G450/G451

Ve vnitřním bodě P* uskutečňuje řídící systém příkazy, jako jsou např. přísuvné pohyby nebo spínací funkce. Tyto příkazy byly naprogramovány v blocích, které se nacházejí mezi dvěma bloky, jež vytvářejí roh.

Když je použit příkaz `G450`, z datového hlediska technicky patří přechodový kruh k navazujícímu příkazu pohybu.

DISC

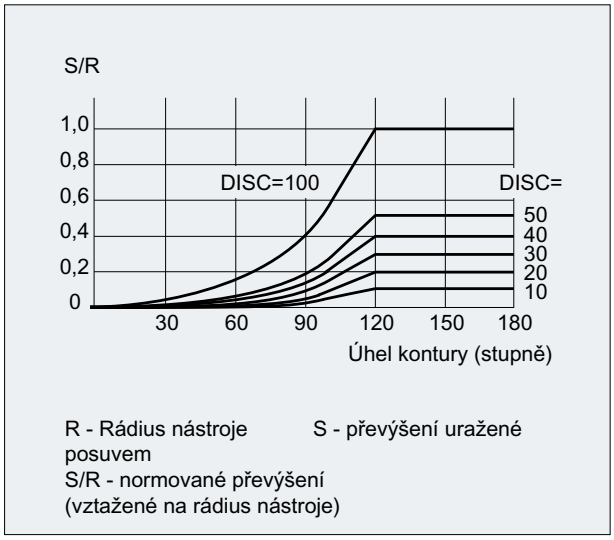
Při zadání hodnoty `DISC` větší než nula se přechodové kružnice zobrazují se zvětšenou výškou – to má za následek, že vznikají přechodové elipsy, paraboly nebo hyperboly:



Prostřednictvím strojního parametru je možné definovat maximální hodnotu – zpravidla se nastavuje DISC=50.

Chování při posuvu

Když je příkaz `G450` aktivní, v případě špičatých úhlů na kontuře a při vysokých hodnotách parametru `DISC` se nástroj na rozích kontury od ní vzdaluje. V případě úhlů kontury vyšších než 120° je kontura objížděna se stejnou rychlostí.

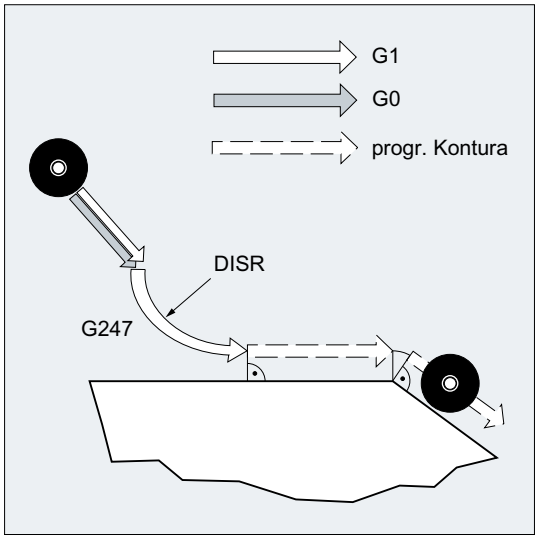


V případě ostrých konturových úhlů, když byl aktivován příkaz `G451`, mohou v důsledku pohybů se zvednutým nástrojem vznikat zbytečné dráhy nástroje naprázdno. Prostřednictvím strojního parametru lze definovat, že v takových případech se bude automaticky přepínat na přechodový kruh.

3.10.4 Měkké najíždění a odjíždění

3.10.4.1 Najíždění a odjíždění (G140 až G143, G147, G148, G247, G248, G347, G348, G340, G341, DISR, DISCL, DISRP, FAD, PM, PR)

Funkce "Měkké najíždění a odjíždění (WAB)" slouží k tomu, aby bylo možné tangenciálně najet na počáteční bod kontury nezávisle na poloze výchozího bodu.



Funkce se používá převážně ve spojení s korekcí rádiusu nástroje.

Při aktivování této funkce přebírá řídící systém úkol vypočítat polohu vnitřních bodů tak, aby se přechod na následující blok (příp. při odjíždění přechod od předcházejícího bloku) odpovídal zadaným parametrům.

Pohyb při najíždění se skládá z maximálně 4 dílčích pohybů. Počáteční bod pohybu je v následujícím textu označen jako P_0, koncový bod jako P_4. Mezi nimi se mohou nacházet maximálně tři vnitřní body P_1, P_2 a P_3. Body P_0, P_3 a P_4 jsou vždy definovány. Vnitřní body P_1 a P_2 mohou v závislosti na nastavení parametrů a geometrických poměrech odpadnout. Při odjíždění se průchod těmito body uskutečňuje v obráceném pořadí, tzn. pohyb začíná v bodě P_4 a končí v bodě P_0.

Syntaxe

Měkké najíždění:

- po přímkách:
  `G147 G340/G341 ... DISR=..., DISCL=..., DISRP=... FAD=...`
- po čtvrtkruhu/půlkruhu:
  `G247/G347 G340/G341 G140/G141/G142/G143 ... DISR=... DISCL=... DISRP=... FAD=...`

Měkké odjíždění:

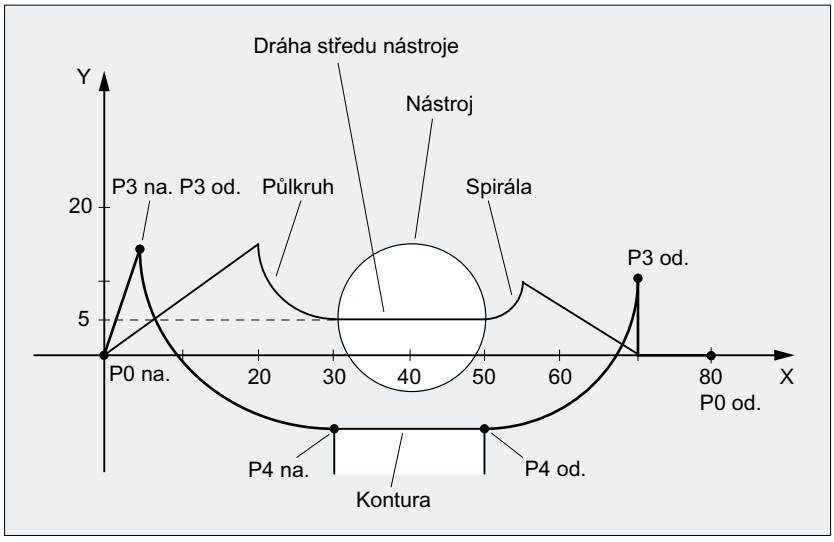
- po přímkách:
  `G148 G340/G341 ... DISR=..., DISCL=..., DISRP=... FAD=...`
- po čtvrtkruhu/půlkruhu:
  `G248/G348 G340/G341 G140/G141/G142/G143 ... DISR=... DISCL=... DISRP=... FAD=...`

Význam

`G147`:	Najíždění po přímce
`G148`:	Odjíždění po přímce
`G247`:	Najíždění po čtvrtkruhu
`G248`:	Odjíždění po čtvrtkruhu
`G347`:	Najíždění po půlkruhu
`G348`:	Odjíždění po půlkruhu
`G340`:	Najíždění a odjíždění v prostoru (základní nastavení)
`G341`:	Najíždění a odjíždění v rovině
`G140`:	Směr najíždění a odjíždění v závislosti na aktuální straně kontury (základní nastavení)
`G141`:	Najíždění zleva, příp. odjíždění vlevo
`G142`:	Najíždění zprava, příp. odjíždění vpravo
`G143`:	Směr najíždění, příp. odjíždění závisí na relativní poloze počátečního, resp. koncového bodu vůči směru tečny

`DISR=...:`	1. Při najíždění a odjíždění po přímkách (G147/G148): Vzdálenost hrany frézy od počátečního bodu kontury 2. Při najíždění a odjíždění po kruhových drahách (G247, G347/G248, G348): Rádius dráhy středu nástroje **Pozor:** U příkazu REPOS s půlkruhem popisuje parametr DISR průměr kruhu.
`DISCL=...:`	Vzdálenost koncového bodu rychlého přísuvného pohybu od roviny obrábění DISCL=AC(...) Zadání absolutní polohy koncového bodu rychlého přísuvného pohybu
`DISCL=AC(...):`	Zadání absolutní polohy koncového bodu rychlého přísuvného pohybu
`DISRP:`	Vzdálenost bodu P1 (návratová rovina) od roviny obrábění
`DISRP=AC(...):`	Údaj absolutní polohy bodu P1
`FAD=...:`	Rychlost pomalého přísuvného pohybu Naprogramovaná hodnota se uplatňuje v závislosti na aktivním typu posuvu (skupina G-funkcí č. 15).
`FAD=PM(...):`	Naprogramovaná hodnota je interpretována nezávisle na aktivním typu posuvu jako lineární posuv (jako G94).
`FAD=PR(...):`	Naprogramovaná hodnota je interpretována nezávisle na aktivním typu posuvu jako otáčkový posuv (jako G95).

Příklad:



- měkké najíždění (aktivuje blok N20)
- najížděcí pohyb po čtvrtkruhu (G247)
- směr najíždění nenaprogramován, platí G140, tzn. korekce rádiusu nástroje je aktivní (G41)
- offset kontury OFFN=5 (N10)

- aktuální rádius nástroje 10; efektivní rádius korekce je tedy pro korekci rádiusu nástroje = 15, rádius kontury WAB=25, takže rádius dráhy středu nástroje je roven DISR=10
- koncový bod kruhu vyplývá z N30, protože v N20 je naprogramována jen pozice Z
- Přísuvný pohyb
  - ze Z20 do Z7 (DISCL=AC(7) rychlým posuvem
  - potom do Z0 s FAD=200
  - Najíždějící kruh v rovině X-Y a následující bloky s F1500 (protože tato rychlost platí pro následující bloky, je nutno přepsat G0 v N30 rychlostí G1, jinak by kontura byla dále opracovávána rychlostí G0).
- měkké odjíždění (aktivuje blok N60)
- odjíždějící pohyb po čtvrtkruhu (G248) a spirále (G340)
- FAD není naprogramováno, protože u G340 nemá žádný význam
- Z=2 v počátečním bodě, Z=8 v koncovém bodě, proto DISCL=6
- při DISR=5 je rádius kontury WAB=20, dráha středu nástroje=5

posuv z Z8 do Z20 a pohyb rovnoběžně s rovinou X-Y do X70 Y0

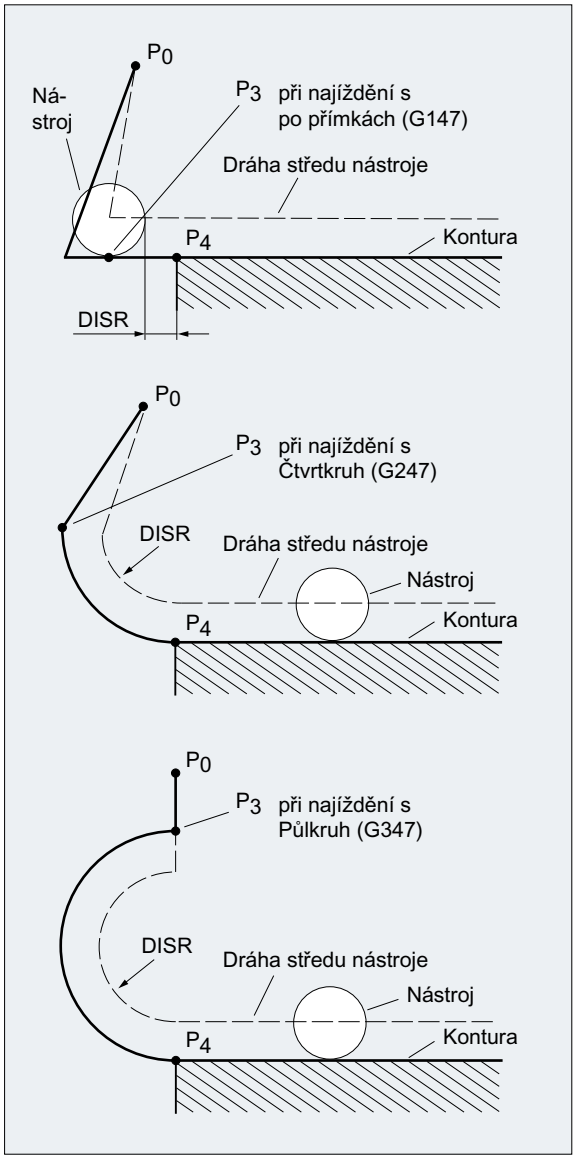
Programový kód	**Komentář**
`$TC_DP1[1,1]=120`	`; definice nástroje T1/D1`
`$TC_DP6[1,1]=10`	`; rádius`
`N10 G0 X0 Y0 Z20 G64 D1 T1 OFFN=5`	`; (najíždění na P0)`
`N20 G41 G247 G341 Z0 DISCL=AC(7) DISR=10 F1500 FAD=200`	`; najíždění (na P3)`
`N30 G1 X30 Y-10`	`; (najíždění na P4)`
`N40 X40 Z2`	
`N50 X50`	`; (odjíždění od P4)`
`N60 G248 G340 X70 Y0 Z20 DISCL=6 DISR=5 G40 F10000`	`; (odjíždění od P3)`
`N70 X80 Y0`	`; (odjíždění od P0)`
`N80 M30`	

Další informace

Volba kontury pro najíždění, příp. odjíždění

Volba kontury pro najíždění, příp. odjíždění se uskutečňuje pomocí odpovídajícího příkazu G-funkce ze skupiny G-funkcí č. 2.

G147:	Najíždění po přímce
G247:	Najíždění po čtvrtkruhu
G347:	Najíždění po půlkruhu
G148:	Odjíždění po přímce
G248:	Odjíždění po čtvrtkruhu
G348:	Odjíždění po půlkruhu



Obrázek 3-20 Nájížděcí pohyb při současném aktivování korekce rádiusu nástroje

Volba směru pro najíždění, příp. odjíždění

Stanovení směru najíždění, příp. odjíždění pomocí korekce rádiusu nástroje (G140, základní nastavení), když je rádius nástroje kladný:

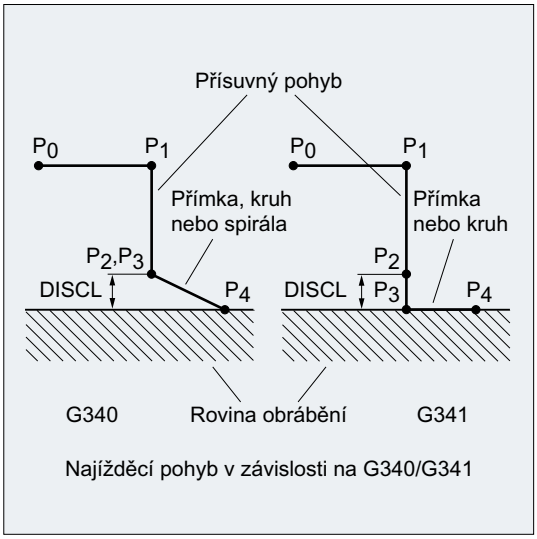
- G41 aktivní --> najíždění zleva
- G42 aktivní --> najíždění zprava

Další možnosti najíždění se zadávají pomocí příkazů G141, G142 a G143.

Tyto příkazy G-funkcí mají význam jen tehdy, pokud je nájížděcí konturou čtvrtkruh nebo půlkruh.

Rozdělení pohybů od počátečního do koncového bodu (G340 a G341)

Pohyby se v každé případě skládají z jedné nebo z několika přímek, stejně jako v závislosti na příkazu G-funkce za účelem stanovení najížděcí kontury z další přímky, příp. ze čtvrtkruhu nebo půlkruhu. Tyto 2 varianty rozdělení dráhy jsou uvedeny na následující obrázku.



Najížděcí pohyb v závislosti na G340/G341

G340:	Najíždění po přímce z bodu P_0 do bodu P_1. Pokud nebyl naprogramován parametr DISRP, je tato přímka rovnoběžná s rovinou obrábění (pracovní rovinou). Přísuv kolmo na rovinu obrábění z bodu P_1 do bodu P_3 na bezpečnostní vzdálenost od roviny obrábění, která je definována parametrem DISCL. Najíždění na koncový bod P4 pomocí křivky stanovené prostřednictvím příkazu G-funkce ze druhé skupiny (přímka, kruh, šroubovice). Pokud je aktivní příkaz G247 nebo G347 (čtvrtkruh nebo půlkruh) a pokud počáteční bod P_3 neleží v rovině obrábění definované prostřednictvím koncového bodu P_4, vloží se místo kruhové dráhy dráha opisující úsek šroubovice. Bod P_2 není definován, příp. spolu s bodem P_3 odpadá. Rovina kruhové dráhy, příp. osa šroubovice je přitom určena pomocí aktivní roviny (G17/G18/G19) stanovené v bloku WAB, tzn. z následujícího bloku se pro určení kruhové dráhy nepoužije samotná tečna v počátečním bodě, ale její průmět do aktivní roviny. Pohyb z bodu P_0 do bodu P_3 se uskuteční po dvou přímkách a s rychlostí, která byla v platnosti před blokem WAB.
G341:	Najíždění po přímce z bodu P_0 do bodu P_1. Pokud nebyl naprogramován parametr DISRP, je tato přímka rovnoběžná s rovinou obrábění (pracovní rovinou). Přísuv kolmo na rovinu obrábění z bodu P_1 až na bezpečnostní vzdálenost od roviny obrábění v bodě P_2, která je definována parametrem DISCL. Přísuv kolmo na rovinu obrábění z bodu P_2 do bodu P_3. Najíždění na koncový bod pomocí křivky stanovené prostřednictvím příkazu G-funkce ze druhé skupiny. P_3 a P_4 leží v rovině obrábění, takže v případě příkazů G247, příp. G347 se nikdy nevkládá šroubovice, ale vždy kruhová dráha.

Ve všech případech, ve kterých se vychází z polohy aktivní roviny G17/G18/G19 (rovina kruhu, osa spirály, přísuvný pohyb kolmo k aktivní rovině), se bere ohled na aktivní otočený frame.

Poloha najížděcí přímky, příp. najížděcího rádiusu při najíždění po kruhovém oblouku (DISR)

- Najíždění/odjíždění po přímce
  DISR udává vzdálenost hrany frézy od počátečního bodu kontury, tzn. délka přímky při aktivní korekci rádiusu nástroje je součtem rádiusu nástroje a naprogramované hodnoty DISR. Rádius nástroje se započítává jen tehdy, pokud je kladný.
  Výsledná délka přímky musí být kladná, tzn. jsou přípustné i záporné hodnoty DISR, pokud je však hodnota DISR menší než rádius nástroje.
- Najíždění/odjíždění po kruhových drahách
  Parametr DISR udává rádius dráhy středu nástroje. Pokud je aktivní korekce rádiusu nástroje, vznikne kruh s takovým rádiusem, aby i v tomto případě vznikla dráha středu nástroje s naprogramovaným rádiusem.

Vzdálenost bodu P2 od roviny obrábění (DISCL)

Jestliže má být poloha bodu P_2 zadána absolutně na ose kolmo k rovině kruhu, je třeba tuto hodnotu naprogramovat ve formě `DISCL=AC(...)`.

Při DISCL=0 platí:

- V případě G340: Celkový najížděcí pohyb se skládá pouze ze dvou bloků (body P_1, P_2 a P_3 jsou spojeny). Najížděcí kontura je tvořena body P_1 až P_4.
- V případě G341: Celkový najížděcí pohyb se skládá ze tří bloků (body P_2 a P_3 jsou spojeny). Jestliže body P_0 a P_4 leží ve stejné rovině, vznikají pouze dva bloky (přísuvný pohyb z P_1 do P_3 odpadá).
- Přitom se sleduje, jestli bod definovaný příkazem DISCL leží mezi body P_1 a P_3, tzn. u všech pohybů, které mají složku kolmou na rovinu obrábění, musí mít tato složka stejné znaménko.
- Při rozpoznání změny směru se připouští tolerance definovaná strojním parametrem MD20204 $MC_WAB_CLEARANCE_TOLERANCE.

Vzdálenost bodu P1 (návratová rovina) od roviny obrábění (DISRP)

Jestliže má být poloha bodu P_1 zadána absolutně na ose kolmo k rovině obrábění, je třeba tuto hodnotu naprogramovat ve formě `DISRP=AC(...)`.

Pokud není tento parametr naprogramován, má bod P_1 stejnou vzdálenost od roviny obrábění jako bod P_0, což znamená, že najížděcí přímka P_0 --> P_1 je rovnoběžná s rovinou obrábění.

Přitom se sleduje, jestli bod definovaný příkazem DISRP leží mezi body P_0 a P_2, tzn. u všech pohybů, které mají složku kolmou na rovinu obrábění (přísuvné pohyby, najížděcí pohyb z bodu P_3 do bodu P_4), musí mít tato složka stejné znaménko. Obrácení směru na opačný je nepřípustné. V takovém případě se aktivuje alarm.

Při rozpoznání změny směru se připouští tolerance definovaná strojním parametrem MD20204 $MC_WAB_CLEARANCE_TOLERANCE. Jestliže bod P_1 leží mimo oblast definovanou body P_0 a P_2 a pokud je tato odchylka menší nebo rovna dané toleranci, předpokládá se, že bod P_1 leží v rovině definované bodem P_0, příp. P_2.

Programování koncového bodu

Koncový bod se zpravidla programuje pomocí X... Y... Z...

Programování koncového bodu kontury při najíždění se výrazně liší od programování koncového bodu při odjíždění. O obou těchto případech je proto pojednáno samostatně.

Programování koncového bodu P4 při najíždění

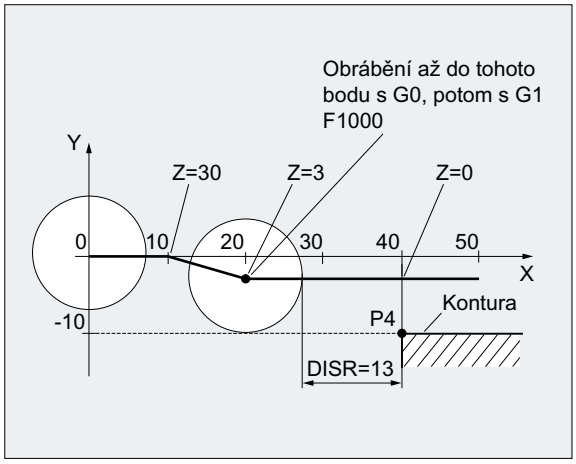
Koncový bod P_4 může být naprogramován v samotném bloku WAB. Kromě toho existuje možnost určit polohu bodu P_4 prostřednictvím koncového bodu následujícího pohybového bloku. Mezi blokem WAB a následujícím blokem pohybu mohou být vloženy další bloky bez pohybu geometrických os.

Příklad:

Programový kód	**Komentář**
`$TC_DP1[1,1]=120`	`; frézovací nástroj T1/D1`
`$TC_DP6[1,1]=7`	`; nástroj s rádiusem 7 mm`
`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`	
`...`	

Bloky `N30/N40` mohou být nahrazeny následujícím:
```
N30 G41 G147 DISCL=3 DISR=13 X40 Y-10 Z0 F1000
```

nebo
```
N30 G41 G147 DISCL=3 DISR=13 F1000
N40 G1 X40 Y-10 Z0
```



Programování koncového bodu P0 při odjíždění

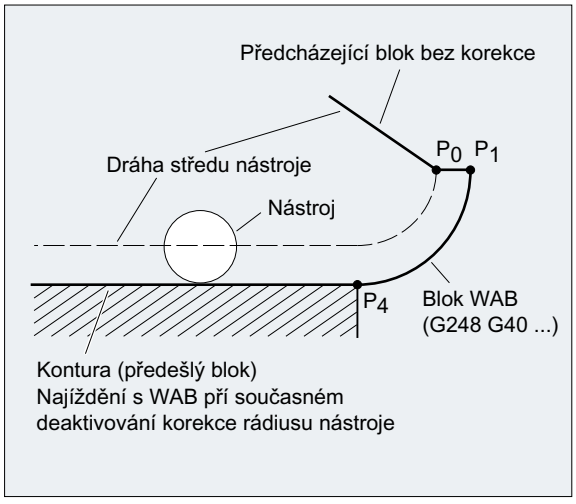
Při odjíždění není možné programování polohy koncového bodu kontury WAB v následujícím bloku předpovědět, což znamená, že se koncová poloha vždy přebírá ze samotného bloku WAB,

a to nezávisle na tom, kolik os bylo naprogramováno. Při určování koncového bodu je nutno rozlišovat mezi následujícími případy:

1. V bloku WAB není naprogramována žádná geometrická osa. Kontura v tomto případě končí v bodě P_1 (je-li naprogramován parametr DISRP), v bodě P_2 (je-li naprogramován parametr DISCL, ale ne DISRP) nebo v bodě P_3 (pokud není naprogramován ani parametr DISCL, ani DISRP).
   Pozice v osách tvořících rovinu obrábění vyplývá z kontury odjíždějící dráhy (koncový bod přímek, příp. kruhových oblouků). Složka v ose, která je kolmá, je definována parametrem DISCL, příp. DISRP. Pokud je v tomto případě jak DISCL=0, tak i DISRP=0, uskutečňuje se pohyb z tohoto důvodu celý v dané rovině, tzn. body P_0 až P_3 spadají v jeden.
2. V bloku WAB je naprogramována pouze osa kolmá na rovinu obrábění. V tomto případě kontura končí v bodě P_0. Jestliže je naprogramován parametr DISRP, (tzn. body P_0 a P_1 nejsou totožné), probíhá přímka P_1 --> P_0 kolmo na rovinu obrábění. Polohy zbývajících dvou os vyplývají stejně jako v případu 1.
3. Je naprogramována minimálně jedna osa z roviny obrábění. Eventuálně chybějící druhá osa z roviny obrábění je modálně doplněna na základě své poslední polohy v předcházejícím bloku.

Poloha osy kolmé na rovinu obrábění se stanoví stejně, jak bylo popsáno v bodu 1. a 2., a to nezávisle na tom, zda je či není tato osa naprogramována. Takto určení poloha definuje polohu koncového bodu P_0. Pokud je v bloku WAB pro odjíždění současně i blok pro deaktivování korekce rádiusu nástroje, vkládá se v prvních dvou případech doplňková dráhová složka v rovině obrábění z bodu P_1 do bodu P_0 tak, aby při deaktivování korekce rádiusu nástroje nevznikl na konci opisované kontury žádný pohyb, tzn. tento bod potom nedefinuje polohu kontury, na kterou má být uplatněna korekce, ale polohu středu nástroje. V tomto případě není nutné realizovat deaktivování korekce rádiusu nástroje zvlášť, protože naprogramovaný bod P_0 již přímo definuje polohu středu nástroje na konci celkové kontury.

Chování v obou případech 1. a 2., tzn. když koncový bod v rovině obrábění není při současném deaktivování korekce rádiusu nástroje explicitně naprogramován, ukazuje následující obrázek:

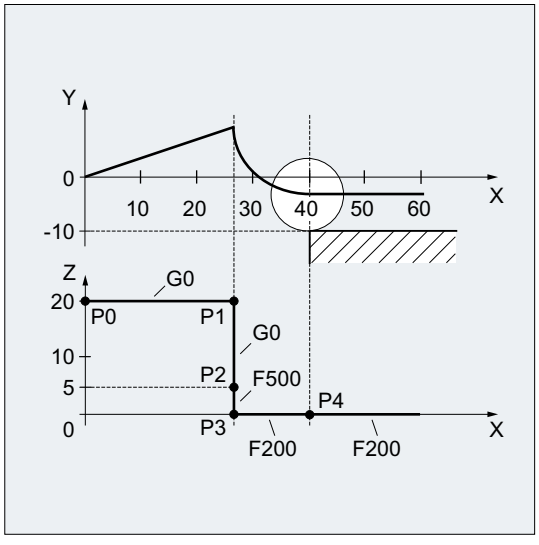


Rychlosti při najíždění, příp. odjíždění

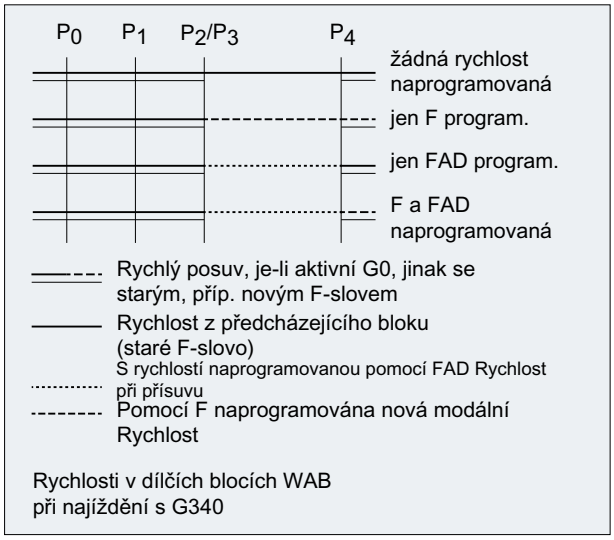
- Rychlost předcházejícího bloku (G0)
  S touto rychlostí jsou prováděny všechny pohyby od P_0 až do P_2, tzn. pohyby rovnoběžné s rovinou obrábění a část přísuvu na bezpečnostní vzdálenost.
- Programování s FAD
  Zadání rychlosti posuvu v těchto případech:
  - G341: Přísuvný pohyb kolmo na pracovní rovinu z bodu P_2 do bodu P_3
  - G340: Z bodu P_2 příp. P_3 do bodu P_4
    Pokud příkaz FAD není naprogramován, bude posuv i na této části kontury prováděn s modálně platnou rychlostí předešlého bloku, leda že by v bloku WAB bylo naprogramováno nové F-slovo.
- Naprogramovaný posuv F
  Tato hodnota posuvu je v platnosti od bodu P_3, příp. P_2, pokud však není naprogramováno FAD. Pokud v bloku WAB není žádné F-slovo naprogramováno, platí rychlost z předcházejícího bloku.

Příklad:

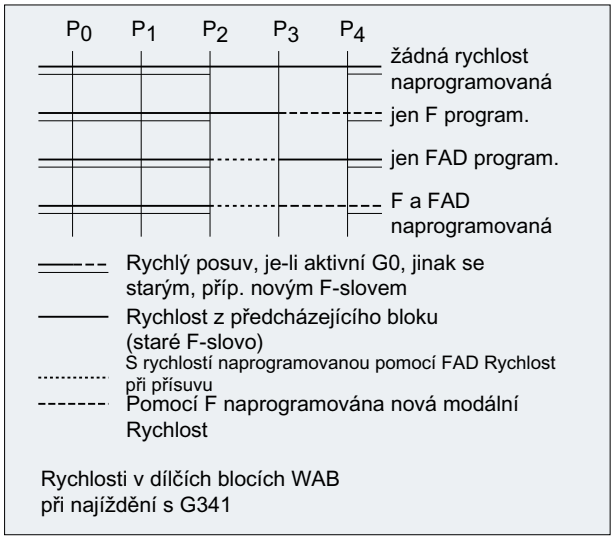
Programový kód	**Komentář**
`$TC_DP1[1,1]=120`	`; frézovací nástroj T1/D1`
`$TC_DP6[1,1]=7`	`; nástroj s rádiusem 7 mm`
`N10 G90 G0 X0 Y0 Z20 D1 T1`	
`N20 G41 G341 G247 DISCL=AC(5) DISR=13 FAD 500 X40 Y-10 Z=0 F200`	
`N30 X50`	
`N40 X60`	
`...`	



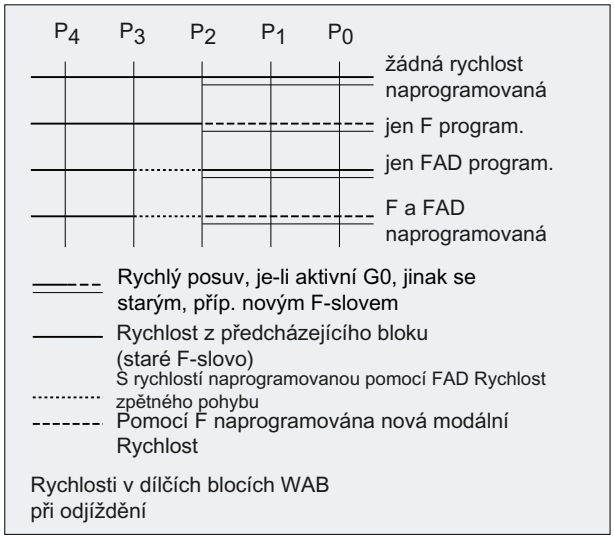
Při odjíždění jsou úlohy modálně platného posuvu z předcházejícího bloku a hodnoty posuvu naprogramované v bloku WAB vyměněny, tzn. pohyb po vlastní odjížděcí kontuře se bude provádět se starou hodnotou posuvu, nová pomocí F-slova naprogramovaná rychlost platí odpovídajícím způsobem od bodu P_2 do bodu P_0.



Rychlosti v dílčích blocích WAB
při najíždění s G340



Rychlosti v dílčích blocích WAB
při najíždění s G341



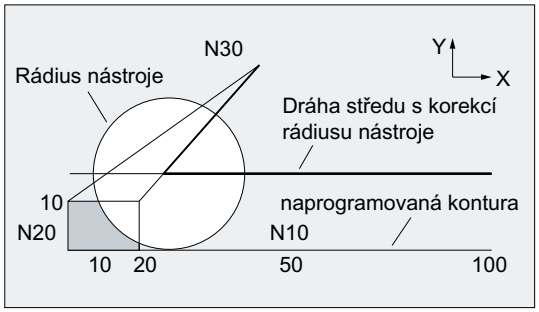
Načítání pozic

Body P_3 a P_4 mohou být při najíždění načítány jako systémové proměnné v souřadném systému obrobku.

- $P_APR: Načtení bodu P
- $_3$ (počáteční bod)
- $P_AEP: Načtení bodu P
- $_4$ (počáteční bod kontury)
- $P_APDV: čtení, zda $P_APR a $P_AEP obsahují platné hodnoty

3.10.4.2 Najíždění a odjíždění s rozšířenými strategiemi odjíždění (G460, G461, G462)

V určitých zvláštních geometrických případech jsou oproti dřívější realizaci se zapnutou protikolizní ochranou zapotřebí rozšířené strategie najíždění a odjíždění při aktivování nebo deaktivování korekce rádiusu nástroje. Protikolizní ochrana tak může mít kupříkladu za následek, že určitý úsek na kontuře nebude opracován až do konce, viz následující obrázek:



Obrázek 3-21 Chování při odjíždění při G460

Syntaxe

```
G460
G461
G462
```

Význam

G460:	Jako dříve (aktivování protikolizního monitorování pro blok najíždění a odjíždění)
G461:	Jestliže neexistuje žádný průsečík, do bloku korekce nástroje se vkládá kruh, jehož střed leží v koncovém bodě bloku bez korekce a jehož rádius je roven rádiusu nástroje. Obrábění se pak bude provádět až do průsečíku s **pomocnou kružnicí** okolo koncového bodu kontury (tedy do konce kontury).
G462:	Jestliže neexistuje žádný průsečík, do bloku korekce nástroje se vloží přímka; blok tak bude prodloužen svou tečnou (standardní nastavení). Obrábění bude tedy probíhat až do **prodloužení** posledního prvku kontury (tedy až do bodu krátce před koncem kontury).

Poznámka

Chování při najíždění je symetrické k chování při odjíždění.

Chování při najíždění, resp. odjíždění je dáno stavem G-příkazů v bloku pro najíždění nebo odjíždění. Chování při najíždění může proto být nastaveno nezávisle na chování při odjíždění.

Příklady

Příklad 1: Chování při odjíždění při G460

V následujících odstavcích je vždy zmiňována situace při deaktivování korekce rádiusu nástroje. Chování při najíždění je zcela analogické.

```
Programový kód          Komentář
G42 D1 T1               ; rádius nástroje 20 mm
...
G1 X110 Y0
N10 X0
N20 Y10
N30 G40 X50 Y50
```

Příklad 2: Najíždění s příkazem G461

```
Programový kód                      Komentář
N10 $TC_DP1[1,1]=120                ; typ nástroje fréza
N20 $TC_DP6[1,1]=10                 ; rádius nástroje
N30 X0 Y0 F10000 T1 D1
N40 Y20
```

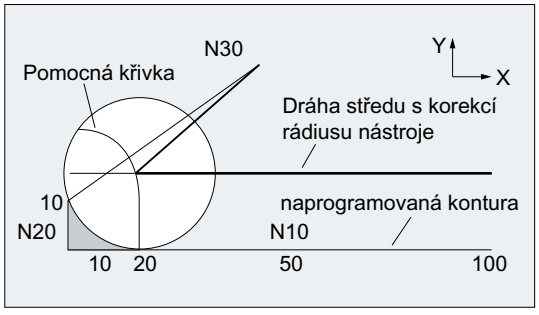
Programový kód	Komentář
N50 G42 X50 Y5 G461	
N60 Y0 F600	
N70 X30	
N80 X20 Y-5	
N90 X0 Y0 G40	
N100 M30	

Další informace

G461

Jestliže neexistuje žádný průsečík mezi posledním blokem s korekcí rádiusu nástroje a předcházejícím blokem, bude offsetová křivka tohoto bloku prodloužena kruhem, jehož střed leží v koncovém bodu bloku bez korekce a jehož rádius se rovná rádiusu nástroje.

Řídící systém se potom pokusí protnout tento kruh s některým z předešlých bloků.



Obrázek 3-22 Chování při odjíždění při G461

Monitorování kolize, CDON, CDOF

Pokud je aktivní CDOF (viz kapitola „Monitorování kolize (CDON, CDOF"), vyhledávání se ukončí, když je nalezen průsečík. Systém tedy nekontroluje, zda existují ještě nějaké další průsečíky s předešlými bloky.

Pokud je aktivní CDON, po nalezení prvního průsečíku pokračuje vyhledávání dalších průsečíků.

Takto nalezený průsečík je novým koncovým bodem předešlého bloku a počátečním bodem bloku deaktivace. Vkládaný kruh se používá výlučně pro výpočet průsečíku a neprodukuje žádný pohyb os.

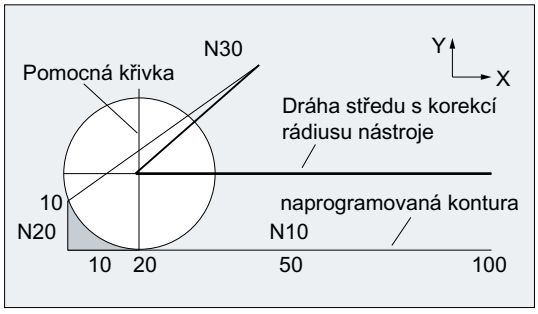
Poznámka

Jestliže není nalezen žádný průsečík, aktivuje se alarm 10751 (nebezpečí kolize).

G462

Jestliže neexistuje žádný průsečík mezi posledním blokem s korekcí rádiusu nástroje a předešlým blokem, při odjíždění pomocí G462 (základní nastavení) se v koncovém bodě posledního bloku s korekcí rádiusu nástroje vkládá přímka (blok je prodloužen svou tečnou).

Vyhledávání průsečíku potom probíhá stejně jako při G461.



Chování při odjíždění s G462 (viz příklad)

U G462 není roh generovaný bloky N10 a N20 v příkladu programu obroben tak, jak by to s použitým nástrojem mohlo být možné. Toto chování se však může ukázat jako nezbytné, pokud je nepřípustné, aby došlo k narušení části kontury (odlišně od naprogramované kontury) vlevo od N20 v příkladu, a to ani s hodnotami y většími než 10 mm.

Chování v rohu, když je aktivní příkaz KONT

Jestliže je aktivní KONT (objíždění kontury v počátečním nebo koncovém bodě), chování se liší podle toho, jestli se koncový bod nachází před nebo za konturou.

- **Koncový bod před konturou**
  Pokud koncový bod leží před konturou, je chování při odjíždění stejné jako při NORM. Tato charakteristika se nemění, ani když je poslední blok kontury při G451 prodloužen přímkou nebo obloukem. Další strategie objíždění zabraňující narušení kontury v blízkosti koncového bodu kontury nejsou proto zapotřebí.
- **Koncový bod za konturou**
  Pokud koncový bod leží za konturou, v závislosti na G450 /G451 se vždy vkládá kruh, příp. přímka. G460 – G462 nemá pak žádný význam. Pokud poslední blok posuvu nemá v této situaci žádný průsečík s předcházejícím blokem, může nyní vzniknout průsečík s vkládaným konturovým prvkem nebo s úsekem přímky od koncového bodu oblouku k naprogramovanému koncovému bodu.
  Pokud je vkládaným konturovým prvkem kruh (G450) a tento kruh má s předešlým bodem průsečík, je to stejný průsečík, který by vznikl při NORM a G461. Obecně však zůstává doplňkový kruhový úsek k dispozici. Pro lineární část odjížděcího bloku už není zapotřebí žádný výpočet průsečíku.
  Ve druhém případě, když není nalezen žádný průsečík vkládaného konturového prvku s předešlým blokem, se najíždí na průsečík odjížděcí přímky s předcházejícím blokem.
  Při aktivních příkazech G461, příp. G462 může tedy vzniknout odlišné chování oproti G460 jen tehdy, pokud je buď aktivní NORM nebo pokud je chování při KONT identické s NORM v důsledku geometrických podmínek.

3.10.5 Aktivování/deaktivování monitorování kolize („rozpoznání hrdla láhve“) (CDON, CDOF, CDOF2)

Monitorování kolize („rozpoznávání hrdla lahve“) s aktivní korekcí rádiusu nástroje je možno v NC programu aktivovat, příp. deaktivovat, pomocí příkazů z G-skupiny 23.

Syntaxe

```
G41/G42 CDON
...
CDOF/CDOF2
```

Význam

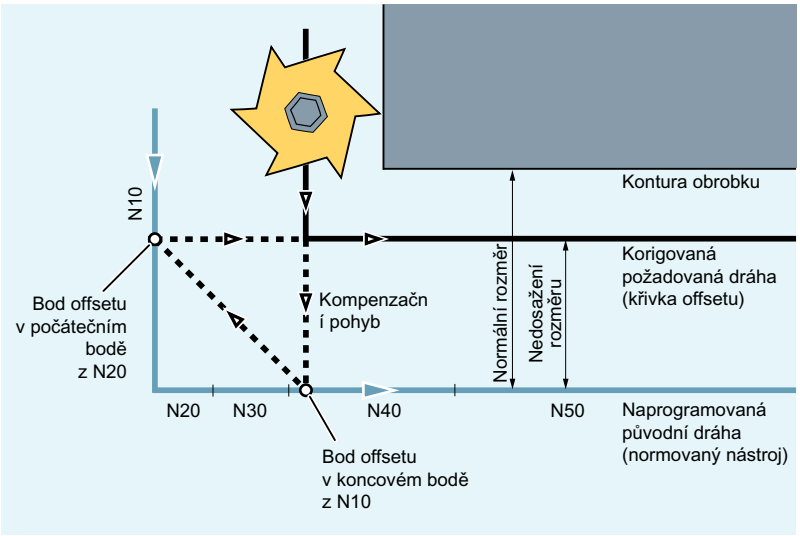
CDON:	Aktivování monitorování kolize („rozpoznávání hrdla lahve“) Pomocí příkazu CDON se v rámci určitého nastavitelného počtu (MD20240) následujících bloků kontroluje, zda se dráhy nástroje v nesousedících, příp. po sobě nenásledujících, blocích neprotínají. Díky tomu lze rozpoznat možné kolize a zavčasu jim předejít.
CDOF:	Deaktivování monitorování kolize („rozpoznávání hrdla lahve“) Pomocí příkazu CDOF se pro aktuální blok hledá společný průsečík u předcházejícího bloku posuvu (u vnitřních rohů) – v případě potřeby se hledá i na blocích nacházejících se dál v minulosti. Jestliže je průsečík nalezen, žádné další bloky se nevyhodnocují. Na vnějších rozích je mezi dvěma po sobě následujícími bloky průsečík vždy nalezen. **Poznámka:** Pomocí příkazu CDOF lze zabránit chybnému rozpoznávání úzkých míst, které je potřeba např. v důsledku chybějících informací v NC programu doplnit.
CDOF2:	Deaktivování monitorování kolize pro 3D obvodové frézování V rámci příkazu CDOF2 se zjišťuje směr korekce nástroje ze sousedících částí bloku. Příkaz CDOF2 se uplatňuje pouze při 3D obvodovém frézování (CUT3Dx) a ve všech ostatních druzích opracování (např. s příkazy CUT2D nebo CUT3DFD) má stejnou funkci jako příkaz CDOF.

Funkce monitorování kolize na příkladu

NC program popisuje dráhu středu normovaného nástroje. Kontura pro právě používaný nástroj má za následek nedosažení rozměru, které je v následujícím obrázku kvůli ilustraci geometrických poměrů zobrazeno nerealisticky velké.

Kromě toho má pro daný příklad platit, že řídící systém monitoruje pouze tři bloky:

MD20240 $MC_CUTCOM_MAXNUM_CHECK_BLOCKS = 3



Protože průsečík existuje pouze mezi offsetovými křivkami N10 a N40, musí být oba bloky N20 a N30 vypuštěny. V tomto příkladu, když řídící systém zpracovává blok N10, blok N40 mu není ještě znám. Z tohoto důvodu je možné vypustit jen jeden blok.

Když je aktivní CDOF2, bude se provádět vyrovnávací pohyb uvedený v obrázku a tento pohyb nebude možné zastavit. V této situaci by aktivní příkazy CDOF nebo CDON měly za následek alarm.

3.10.6 2 1/2 D-korekce nástroje (CUT2D, CUT2DD, CUT2DF, CUT2DFD)

2½ D korekci rádiusu nástroje je potřeba použít tehdy, jestliže se kvůli opracovávání šikmých ploch neprovádí nasměrování nástroje, ale otáčí se **obrobek**. Pro aktivování slouží příkazy `CUT2D`, `CUT2DD`, `CUT2DF` nebo `CUT2DFD`.

Korekce délky nástroje

Korekce délky nástroje se vždy vztahuje na pevnou neotočenou pracovní rovinu.

2½ D korekce rádiusu nástroje pro nástroje používané pro opracování kontur

2½ D korekce rádiusu nástroje pro obrábění kontury se aktivuje tehdy, pokud je spolu s příkazem `CUT2D`, `CUT2DD`, `CUT2DF` nebo `CUT2DFD` naprogramován jeden ze dvou příkazů `G41` (korekce rádiusu nástroje vlevo od kontury) nebo `G42` (korekce rádiusu nástroje vpravo od kontury). Slouží pro automatickou volbu břitu pro rotačně nesymetrické nástroje, s nimiž je možné kus po kuse obrábět jednotlivé úseky kontury.

Poznámka

Pokud není 2½ D korekce rádiusu nástroje aktivní, chová se nástroj pro obrábění kontury stejně jako normální nástroj, který je vybaven pouze prvním břitem.

2½ D korekce nástroje vztažená na diferenční nástroj

2½ D korekce rádiusu nástroje vztažená na diferenční nástroj se aktivuje prostřednictvím příkazů `CUT2DD` nebo `CUT2DFD`. Je potřeba ji použít tehdy, jestliže je naprogramovaná kontura vztažena

na dráhu středu diferenčního nástroje a pokud se pro opracování používá jeden z nástrojů, které se od něj odlišují. Při výpočtu 2½ D korekce rádiusu nástroje se potom započítává pouze hodnota opotřebení rádiusu aktivního nástroje ($TC_DP_15) a v případě potřeby naprogramovaný korekční offset nástroje OFFN (Strana 253) a TOFFR (Strana 84). Základní rádius ($TC_DP6) aktivního nástroje se **nezapočítává**.

Syntaxe

```
CUT2D
CUT2DD
CUT2DF
CUT2DFD
```

Význam

CUT2D:	Aktivování 2½ D korekce rádiusu
CUT2DD:	Aktivování 2½ D korekce rádiusu vztažená na diferenční nástroj
CUT2DF:	Aktivování 2½ D korekce rádiusu, korekce rádiusu nástroje vztažená na aktuální frame, příp. na šikmou rovinu.
CUT2DFD:	Aktivování 2½ D korekce rádiusu vztažené k diferenčnímu nástroji vzhledem k aktuálnímu framu, příp. na šikmé rovině.

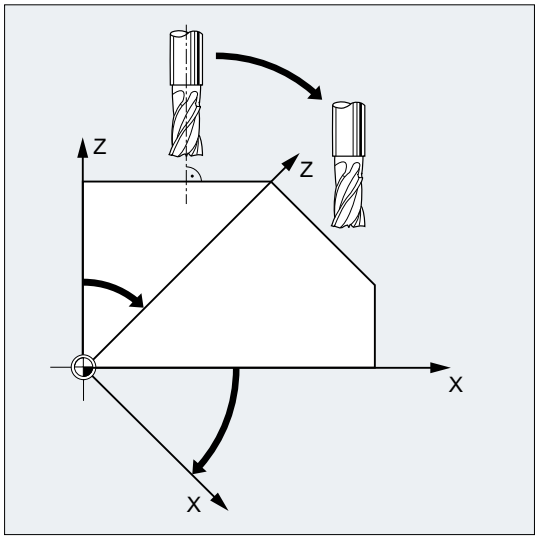
Další informace

Nástroje pro obrábění kontury

- Uvolnění
  Uvolnění korekce rádiusu nástroje pro obrábění kontury se uskutečňuje pro daný kanál prostřednictvím parametru:
  MD28290 $MC_MM_SHAPED_TOOLS_ENABLE
- Typ nástroje
  Typy nástrojů pro obrábění kontury jsou v daném kanálu definovány pomocí parametru:
  MD20370 $MC_SHAPED_TOOL_TYPE_NO
- Břity
  Každému nástroji pro obrábění kontury může být v libovolné posloupnosti přiřazen určitý počet břitů (D-čísel). Maximální počet břitů na jeden nástroj se nastavuje pomocí následujících parametrů:
  MD18106 $MN_MM_MAX_CUTTING_EDGE_PERTOOL
  První břit nástroje pro obrábění kontury je břit, který je zvolen při aktivování nástroje. Je-li např. v programu pomocí příkazu `D5 T3` aktivován pátý břit (D5) třetího nástroje (T3), potom definuje D5 a následující břity nástroj pro obrábění kontury buď částečně nebo jako celek. Břity nacházející se před D5 jsou ignorovány.

2½ D korekce rádiusu nástroje bez otočení roviny korekce (CUT2D, CUT2DD)

Pokud je naprogramován frame, který obsahuje otočení, a je-li aktivován příkaz `CUT2D`, příp. `CUT2DD`, potom se rovina, v níž se uskutečňuje korekce rádiusu nástroje (rovina korekce), **nebude otáčet**. Korekce rádiusu nástroje se započítává vzhledem k **neotočené** pracovní rovině (`G17`, `G18`, `G19`). Korekce délky nástroje se uplatňuje vzhledem k rovině korekce.

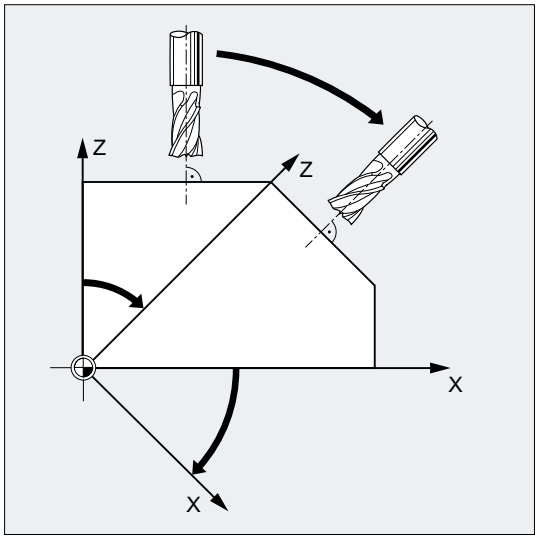


Pro obrábění šikmých ploch musí být hodnoty korekčních parametrů nástroje odpovídajícím způsobem definovány nebo musí být přepočítány pomocí funkcí „Korekce délky nástroje pro orientovatelné nástroje“.

2½ D korekce rádiusu nástroje s otočením roviny korekce (CUT2DF, CUT2DFD)

Pokud je naprogramován frame, který obsahuje otočení, a je-li aktivován příkaz `CUT2DF`, příp. `CUT2DFD`, potom se rovina, v níž se uskutečňuje korekce rádiusu nástroje (rovina korekce), **otáčí také**. Korekce rádiusu nástroje se započítává vzhledem k **otočené** pracovní rovině (`G17`, `G18`, `G19`). Korekce délky nástroje je ale i nadále vztažena k **neotočené** pracovní rovině.

Předpoklad: U stroje musí být možno nastavit orientaci nástroje tak, aby byl kolmo k otočené pracovní rovině a aby s ním bylo možno uskutečnit obrábění.



Poznámka

Korekce délky nástroje je i nadále vztažena k neotočené pracovní rovině.

Pokud budete potřebovat další informace, viz Příručka k funkcím, „Nástroje“.

3.10.7 Udržení konstantní korekce rádiusu nástroje (CUTCONON, CUTCONOF)

Funkce "Udržení konstantní korekce rádiusu nástroje" slouží k tomu, abyste mohli pro určitý počet bloků potlačit korekci rádiusu nástroje, přičemž však rozdíl mezi naprogramovanou a skutečně realizovanou dráhou středu nástroje, který vznikl v předcházejících blocích prostřednictvím korekce rádiusu nástroje, zůstává zachován jako posunutí. Tuto funkci je možné výhodně použít např. tehdy, jestliže je při frézování drážek zapotřebí uskutečnit v bodech obratu větší počet bloků posuvu, při kterých jsou ale nežádoucí kontury (strategie objíždění), které vznikají v důsledku korekce rádiusu nástroje. Tuto funkci lze používat nezávisle na druhu korekce rádiusu nástroje ($2^1/_2$D, 3D frézování na čelní ploše, 3D obvodové frézování).

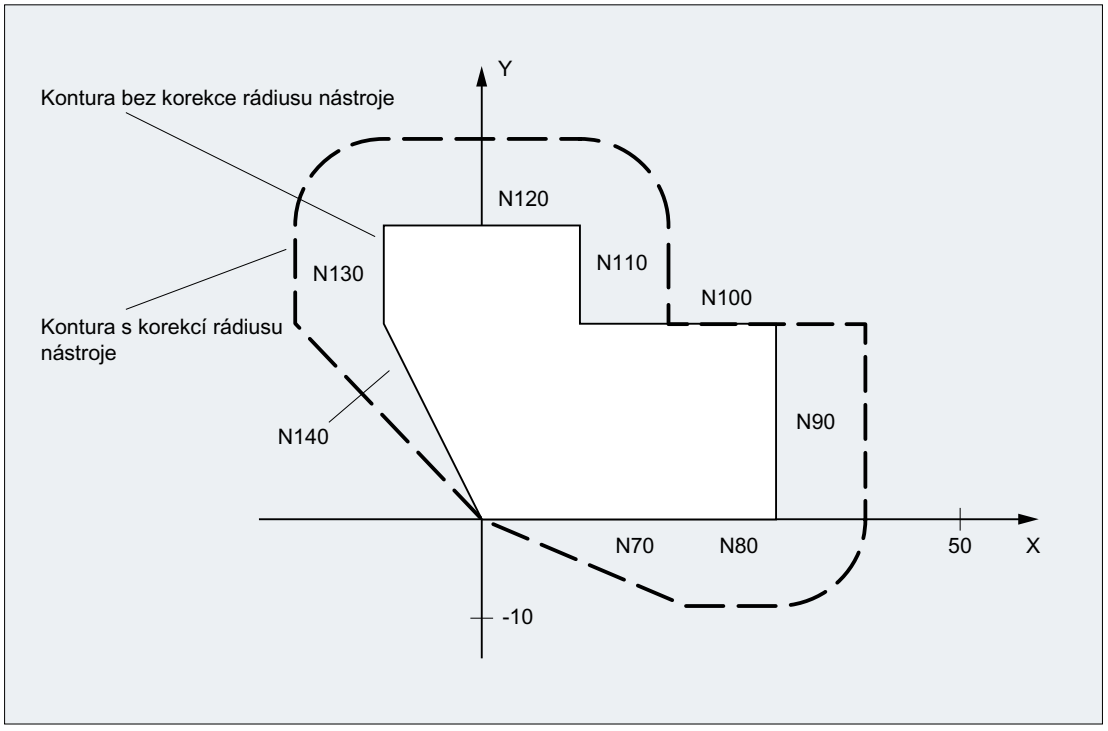
Syntaxe

```
CUTCONON
CUTCONOF
```

Význam

CUTCONON:	Příkaz pro aktivování funkce "Udržení konstantní korekce rádiusu nástroje"
CUTCONOF:	Příkaz pro deaktivování funkce "Udržení konstantní korekce rádiusu nástroje"

Příklad:



Programový kód	**Komentář**
N10	; definice nástroje D1
N20 $TC_DP1[1,1]= 110	; typ
N30 $TC_DP6[1,1]= 10.	; rádius
N40	
N50 X0 Y0 Z0 G1 G17 T1 D1 F10000	
N60	
N70 X20 G42 NORM	
N80 X30	
N90 Y20	
N100 X10 CUTCONON	; aktivování potlačení korekce
N110 Y30 KONT	; případné vložení obloukové dráhy při vypnuté korekci rádiusu nástroje
N120 X-10 CUTCONOF	
N130 Y20 NORM	; žádná oblouková dráha při vypnuté korekci rádiusu nástroje
N140 X0 Y0 G40	
N150 M30	

Další informace

Za normálních okolností je před aktivováním potlačení korekce rádiusu nástroje tato korekce již v platnosti, a jakmile je potlačování korekce rádiusu nástroje opět deaktivováno, vstupuje opět v platnost. V posledním pohybovém bloku před příkazem `CUTCONON` se v koncovém bodě bloku najíždí na posunutý bod. Ve všech následujících blocích, v nichž je potlačování korekce aktivní, se nástroj pohybuje bez korekce. Přitom je však prostorově posunutý o vektor z koncového bodu posledního bloku s korekcí k tomuto posunutému bodu. Typ interpolace v tomto bloku (lineární, kruhová, polynomická) je libovolný.

V bloku s deaktivováním potlačování korekce, tzn. v bloku, který obsahuje příkaz `CUTCONOF`, se korekce uplatňuje jako obvykle. Začíná v posunutém počátečním bodě. Mezi koncovým bodem předcházejícího bloku, tzn. posledním naprogramovaným blokem posuvu s aktivním příkazem `CUTCONON`, a tímto bodem se vkládá lineární blok.

Kruhové bloky, v nichž rovina kruhového oblouku leží kolmo na rovinu korekce (svislé kruhové oblouky), jsou realizovány tak, jako by v nich byl naprogramován příkaz `CUTCONON`. Toto implicitní aktivování potlačování korekce rádiusu nástroje se automaticky deaktivuje v prvním pohybovém bloku, který obsahuje příkaz pohybu v rovině korekce a který není kruhovým pohybem výše uvedeného druhu. Svislé kruhové pohyby se v tomto smyslu mohou vyskytnout pouze při obvodovém frézování.

3.10.8 Nástroje se specifickou polohou břitu

U nástrojů se specifickou polohou břitu (brusné a soustružnické nástroje – typy nástrojů 400 – 599; viz kapitola „Vyhodnocování znaménka opotřebení") se na přechod od G40 do G41/G42, příp. obráceně, pohlíží jako na výměnu nástroje. Při aktivní transformaci (např. TRANSMIT) to má za následek zastavení procesu předběžného zpracování (zastavení dekódování) a v důsledku toho případně i k odchylce od zamýšlené kontury obrobku.

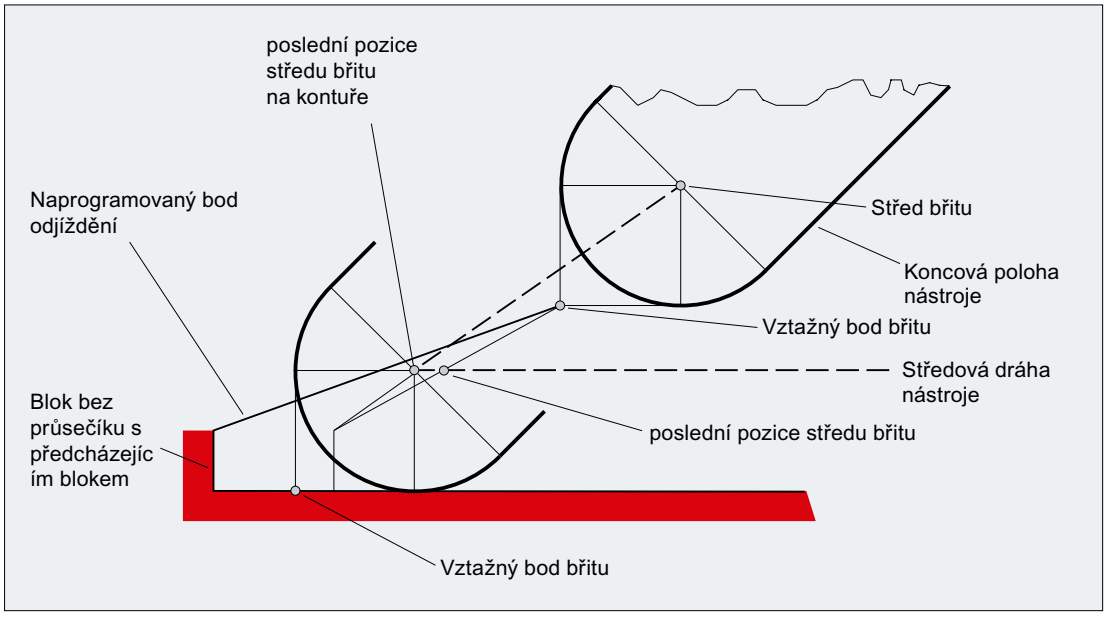
Předcházející chování této funkce se mění v následujících ohledech:

1. Když je aktivní TRANSMIT, dochází k zastavení předběžného zpracování
2. Výpočet průsečíků při najíždění nebo odjíždění v režimu KONT
3. Výměna nástroje při aktivní korekci rádiusu nástroje
4. Korekce rádiusu nástroje s proměnnou orientací nástroje při transformaci

Další informace

Předcházející chování této funkce bylo změněno následujícím způsobem:

- Na přechod od G40 do G41/G42, příp. obráceně, se už nepohlíží jako na výměnu nástroje. Při aktivování příkazu TRANSMIT proto nedochází k zastavení preprocesoru.
- Pro výpočet průsečíků s blokem najíždění, příp. odjíždění se použije přímka spojující středy břitu na počátku bloku a na konci bloku. Rozdíl mezi vztažným bodem břitu a středem břitu bude superponován na tento pohyb.
  Při najíždění, příp. odjíždění s KONT (nástroj objíždí bod kontury, viz předešlý odstavec „Najíždění a odjíždění od kontury“) se provádí superpozice lineárního dílčího bloku najížděcího, resp. odjížděcího pohybu. Geometrické chování je proto u nástrojů identické, ať už s nebo bez relevantních poloh břitu. Rozdíly oproti dřívějšímu chování vznikají pouze v relativně vzácných případech, kdy blok najíždění, resp. odjíždění tvoří průsečík s blokem posuvu, který není sousední, viz následující obrázek:



- Výměna nástroje při aktivní korekci rádiusu nástroje, při které se mění vzdálenost mezi středem břitu a vztažným bodem břitu, je v kruhových blocích a blocích posuvu s racionálními polynomy stupně > 4 zakázána. Při jiných druzích interpolace je výměna na rozdíl od dřívějšího stavu přípustná i při aktivní transformaci (např. TRANSMIT).
- Při korekci rádiusu nástroje s proměnnou orientací nástroje už není možné provádět transformaci od vztažného bodu břitu na střed břitu pomocí jednoduchého posunutí počátku. Nástroje, pro které je poloha břitu důležitá, jsou proto při 3D-obvodovém frézování zakázány (alarm).

Poznámka

Pro čelní frézy toto téma nemá význam, protože jsou stejně jediným přípustným definovaným typem nástroje bez relevantní polohy břitu pro tuto operaci. (S nástroji, jejichž typ není výslovně povolen, se zachází jako s frézami s kulovou hlavou se specifikovaným rádiusem. Údaj polohy břitu je pak ignorován.)

3.11 Chování při pohybu po dráze

3.11.1 Přesné najetí (G60, G9, G601, G602, G603)

Přesné najetí je režim pohybu, při kterém jsou na konci každého bloku s pracovním posuvem všechny na tomto pohybu se podílející dráhové osy a doplňkové osy, které se nemají pohybovat za hranice bloku, zabrzděny do úplného zastavení.

Přesné najetí se používá tehdy, když se mají vyrábět ostré vnější nebo vnitřní rohy při obrábění načisto na konečný rozměr.

Prostřednictvím kritéria přesného najetí je definováno, jak přesně se na rohový bod bude najíždět a kdy má dojít k přechodu na další blok.

- "Jemné přesné najetí"
  Přechod na další blok se uskuteční v okamžiku, kdy všechny osy podílející se na pohybu pracovním posuvem dosáhnou osových tolerančních hranic "Jemné přesné najetí".
  „Jemné přesné najetí“ se nastavuje pomocí parametru: MD36010 $MA_STOP_LIMIT_FINE[<osa>]
- "Hrubé přesné najetí"
  Přechod na další blok se uskuteční v okamžiku, kdy všechny osy podílející se na pohybu pracovním posuvem dosáhnou osových tolerančních hranic "Hrubé přesné najetí".
  „Hrubé přesné najetí“ se nastavuje pomocí parametru: MD36000 $MA_STOP_LIMIT_COARSE[<osa>]
- "Zastavení interpolátoru"
  Přechod na následující blok se uskuteční, jestliže řídící systém má vypočítánu požadovanou hodnotu rychlosti os, které se na pohybu pracovním posuvem podílejí, rovnu nule. Skutečná polohy, příp. vzdálenost daná vlečnou chybou, os podílejících se na pohybu se nebere v úvahu.

Syntaxe

```
G60 ...
G9 ...
G601/G602/G603 ...
```

Význam

G60:	Příkaz pro aktivování přesného najetí s **modální** platností
G9:	Příkaz pro aktivování přesného najetí s **blokovou** platností
G601:	Příkaz pro aktivování kritéria "**jemné přesné najetí**"
G602:	Příkaz pro aktivování kritéria "**hrubé přesné najetí**"
G603:	Příkaz pro aktivování kritéria "**zastavení interpolátoru**"

Poznámka

Příkazy pro aktivování kritérií přesného najetí (G601 / G602 / G603) se uplatňují jen tehdy, když je aktivován příkaz G60 nebo G9.

Příklad:

Programový kód	Komentář
N5 G602	; je aktivováno kritérium "Hrubé přesné najetí
N10 G0 G60 Z...	; modální přesné najetí aktivní
N20 X... Z...	; G60 je v platnosti i nadále
...	
N50 G1 G601	; je aktivováno kritérium "Jemné přesné najetí"
N80 G64 Z...	; přepnutí do režimu řízení pohybu po dráze
...	
N100 G0 G9	; přesné najetí je v platnosti jen v tomto bloku
N110 ...	; znovu je aktivní režim řízení pohybu po dráze

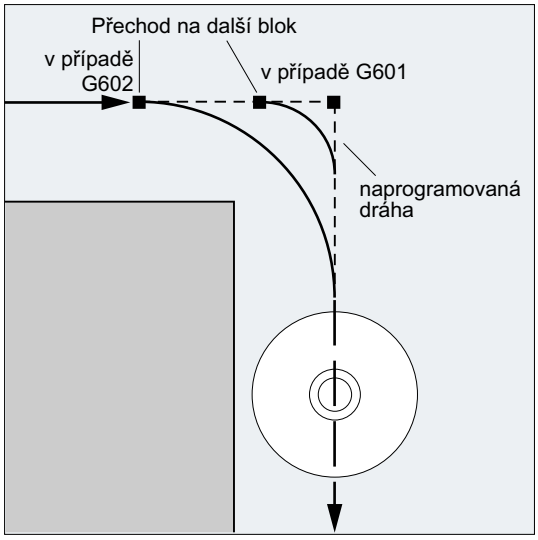
Další informace

G60, G9

G9 v aktuálním bloku aktivuje přesné najetí, G60 v aktuálním bloku a ve všech následujících blocích.

Pomocí příkazů pro řízení pohybu po dráze G64 nebo G641 ... G645 se příkaz G60 deaktivuje.

G601, G602



Pohyb bude přibržděn a v rohovém bodě krátce pozastaven.

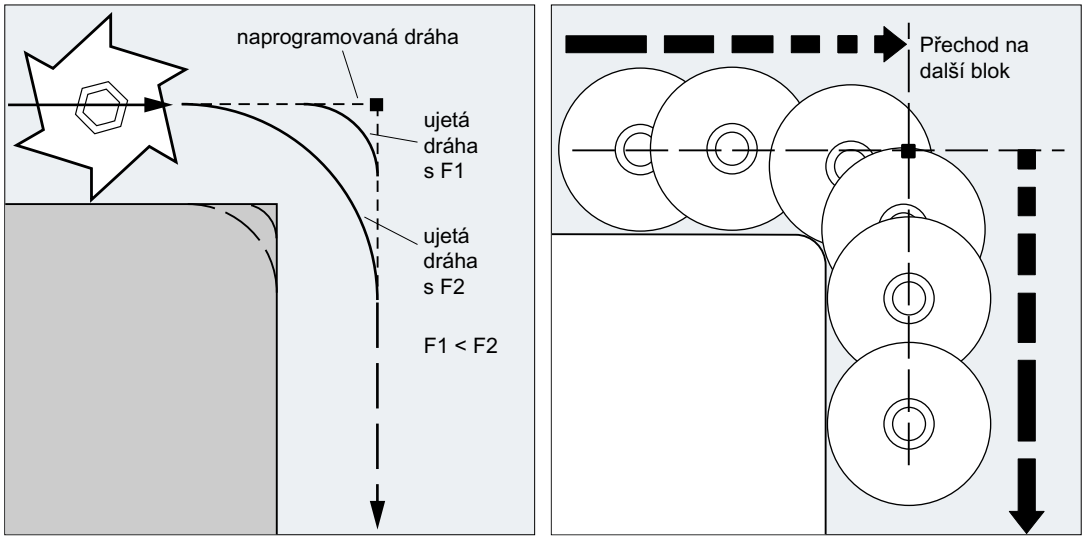
Poznámka

Hrance kritérií pro přesné najetí by měly být nastaveny jen tak úzké, jak je to nutné. Čím jsou tyto meze užší, tím déle trvá polohování a najíždění do cílové pozice.

G603

Přechod na následující blok se uskuteční, jestliže řídící systém má vypočítánu požadovanou hodnotu rychlosti os, které se na pohybu podílejí, rovnu nule. V tomto okamžiku se skutečná

poloha nachází – v závislosti na dynamice os a rychlosti pohybu po dráze – pozadu o doběhovou vzdálenost. Rohy obrobku nyní mohou být zaobleny.



Kritérium přeného najetí nastavené v konfiguraci

Pro G0 a pro všechny zbývající příkazy z 1. skupiny G-funkcí je možné v každém kanálu zvlášť nastavit, že se bude automaticky používat předem definované kritérium přesného najetí, které se liší od kritéria naprogramovaného (viz dokumentace od výrobce stroje!).

3.11.2 Režim řízení pohybu po dráze (G64, G641, G642, G643, G644, G645, ADIS, ADISPOS)

V režimu řízení pohybu po dráze není rychlost dráhových os na konci bloku před přechodem na další blok zabrzděna na takovou hodnotu, která umožňuje dosažení kritéria přesného najetí. Cílem oproti tomu je zabránit velkým brzděním dráhových os v bodech přechodu na další blok, aby se přecházelo do následujícího bloku pokud možno se stejnou rychlostí pohybu po dráze. Aby bylo možno tohoto cíle dosáhnout, při zahájení režimu řízení pohybu po dráze se navíc ještě aktivuje funkce "Předvídání průběhu rychlosti (funkce Look Ahead)".

Režim řízení pohybu po dráze s přechodovými zaobleními znamená, že skokové změny na hranicích bloků v důsledku lokálních změn naprogramovaného průběhu jsou realizovány tangenciálně, příp. jsou vyhlazeny.

Režim řízení pohybu po dráze způsobuje:

- zaoblení kontury
- kratší doby opracování díky odstranění operací brždění a zrychlování, které jsou nezbytné pro dosažení kritéria přesného najetí
- lepší řezné podmínky díky spojitému průběhu rychlosti

Režim řízení pohybu po dráze má smysl za těchto okolností:

- Kontura má být opracována do možno nejplynuleji (např. rychlým posuvem).
- Přesný průběh se smí odchylovat od naprogramovaného v rámci kritérií daných tolerancemi, aby bylo možné vyrábět plynulé spojité křivky.

Režim řízení pohybu po dráze nemá smysl za těchto okolností:

- Kontura má být opracována přesně.
- Je požadována naprosto konstantní rychlost.

Poznámka

Režim řízení pohybu po dráze je přerušen bloky, které implicitně vyvolávají zastavení předběžného zpracování, co je např.:

- Přístup k určitým stavovým údajům stroje ($A...).
- Výstupy pomocných funkcí

Syntaxe

```
G64 ...
G641 ADIS=…
G641 ADISPOS=…
G642 ...
G643 ...
G644 ...
G645 ...
```

Význam

G64:	Režim řízení pohybu po dráze se snižováním rychlosti podle faktoru přetížení
G641:	Režim řízení pohybu po dráze s přechodovými zaobleními podle kritéria dráhy
ADIS=	Kritérium dráhy v příkazu G641 pro dráhové funkce G1, G2, G3, ...
ADISPOS=...	Kritérium dráhy v příkazu G641 pro rychlý posuv s G0
	Kritérium dráhy (= vzdálenost zaoblení) ADIS příp. ADISPOS popisuje úsek dráhy, o kterou smí zaoblení před koncem bloku předčasně začínat, příp. úsek dráhy po skončení bloku, na kterém musí být zaoblení ukončeno. **Upozornění:** Pokud pro parametry ADIS/ADISPOS není naprogramována žádná hodnota, platí hodnota nulová, takže chování je stejné jako u příkazu G64. Při krátkých úsecích dráhy se vzdálenost zaoblení automaticky zmenšuje (až o max. 36%).

G642:	Režim řízení pohybu po dráze s přechodovými zaobleními při dodržení definovaných tolerancí V tomto režimu se zaoblení uskutečňuje v normálním případě tak, aby byla dodržena maximální přípustná odchylka dráhy. Místo této tolerance, která je vztažena na určitou osu, může být v konfiguraci nastavena také maximální odchylka kontury (tolerance kontury) nebo maximální odchylka úhlu orientace nástroje (tolerance orientace). **Upozornění:** Rozšíření o toleranci kontury a toleranci orientace existuje jedině v systémech, v nichž je k dispozici volitelný doplněk "Polynomická interpolace".
G643:	Režim řízení pohybu po dráze s přechodovými zaobleními při dodržení definovaných tolerancí (uvnitř bloku) U příkazu G643 nevzniká oproti příkazu G642 nevzniká žádný vlastní blok přechodového prvku, ale pro každou osu se vkládají interní blokové přechodové pohyby. Dráha zaoblení může být pro každou osu rozdílná.
G644:	Režim řízení pohybu po dráze s přechodovými zaobleními s maximální možnou dynamikou **Upozornění:** Pokud je aktivní kinematická transformace, použití příkazu G644 není možné. Interně dojde k přepnutí na G642.
G645:	Režim řízení pohybu po dráze s přechodovými zaobleními v rozích a s tangenciálními přechody mezi bloky při dodržení definovaných tolerancí Na rozích se příkaz G645 chová stejně jako příkaz G642. U příkazu G645 jsou bloky přechodových zaoblení vytvářeny pouze na tangenciálních přechodech mezi bloky, jestliže průběh zakřivení původní kontury vykazuje u minimálně jedné osy skokovou změnu.

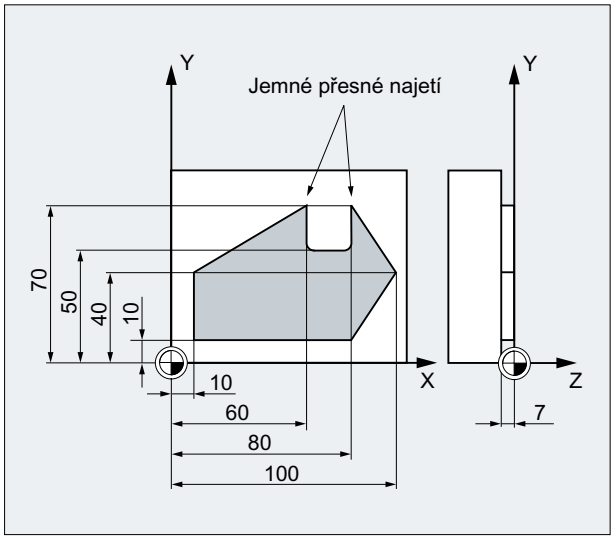
Poznámka

Tento typ zaoblení rozhodně není náhradou zaoblení v rohu (RND). Uživatel by neměl mít žádné představy o tom, jak bude kontura v místě zaoblení přechodu vypadat. Tento druh zaoblení přechodů může záviset zejména také na dynamických podmínkách, např. na rychlosti pohybu po dráze. Zaoblení přechodů na kontuře má proto smysl jen s malými hodnotami parametru ADIS. Pokud má být roh objížděn po definované kontuře bez výjimky, musí být použit příkaz RND.

Poznámka

Jestliže je pohyb při přechodovém zaoblení vytvářeném příkazy G641, G642, G643, G644 nebo G645 přerušen, při následném najíždění na původní polohu (REPOS) se nebude najíždět na místo, kde došlo k přerušení, ale na počáteční nebo koncový bod původního bloku posuvu (v závislosti na režimu funkce REPOS).

Příklad:



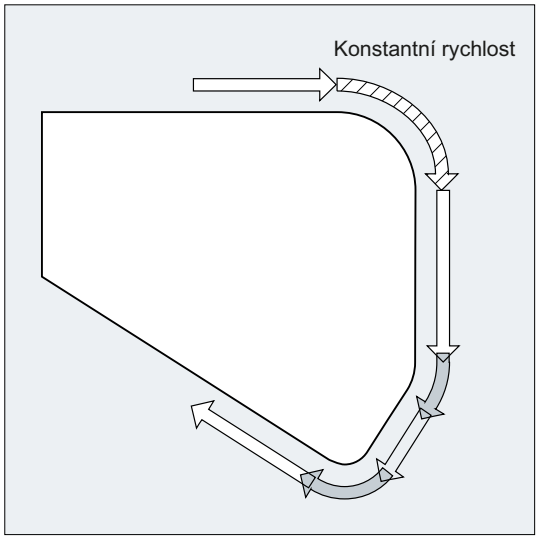
Oba vnější rohy drážky mají být opracovány přesně. Zbytek drážky má být opracován v režimu řízení pohybu po dráze.

Programový kód	**Komentář**
N05 DIAMOF	; zadávání rádiusů
N10 G17 T1 G41 G0 X10 Y10 Z2 S300 M3	; najíždění na počáteční pozici, zapnutí vřetena, korekce pohybu po dráze
N20 G1 Z-7 F8000	; přísuv nástroje
N30 G641 ADIS=0.5	; konturové přechody budou zaobleny
N40 Y40	
N50 X60 Y70 G60 G601	; najíždění přesně na pozici s jemným přesným najetím
N60 Y50	
N70 X80	
N80 Y70	
N90 G641 ADIS=0.5 X100 Y40	; konturové přechody budou zaobleny
N100 X80 Y10	
N110 X10	
N120 G40 G0 X-20	; vypnutí korekce posuvu po dráze
N130 Z10 M30	; odjíždění nástroje, konec programu

Další informace

Režim řízení pohybu po dráze G64

V režimu řízení pohybu po dráze se nástroj pohybuje při tangenciálních konturových přechodech s co možno nejvíce konstantní rychlostí (žádné brždění na hranicích bloků). Před rohy a bloky s přesným najetím se předem brzdí (funkce Look Ahead).



Také rohy jsou objížděny s konstantní rychlostí. Aby se zabránilo narušení kontury, rychlost se snižuje, neboť je nutno vzít v úvahu mezní hodnoty zrychlení a faktory přetížení.

Poznámka

To, nakolik jsou konturové přechody vyhlazeny, závisí na rychlosti posuvu a faktoru přetížení. Faktor přetížení může být nastaven ve strojním parametru MD32310 $MA_MAX_ACCEL_OVL_FACTOR.

Nastavení strojního parametru MD20490 $MC_IGNORE_OVL_FACTOR_FOR_ADIS má za následek, že přechody mezi bloky budou vždy zaoblovány bez ohledu na nastavení faktoru přetížení.

Aby se zabránilo nechtěnému zastavení pohybu po dráze (řezání naprázdno), je nutno dbát následujících zásad:

- Pomocné funkce, které se aktivují po skončení pohybu nebo před zahájením dalšího pohybu, způsobují přerušení režimu řízení pohybu po dráze (výjimka: rychlé pomocné funkce).
- Polohovací osy se vždy pohybují podle principu přesného najetí, okno přesného najetí jemné (jako u G601). Jestliže se v NC-bloku musí na polohovací osy čekat, bude režim řízení pohybu po dráze u dráhových os přerušen.

Vložené programové bloky, které obsahují pouze komentáře, výpočetní bloky nebo volání podprogramů, oproti tomu režim řízení pohybu po dráze nepřerušují.

Poznámka

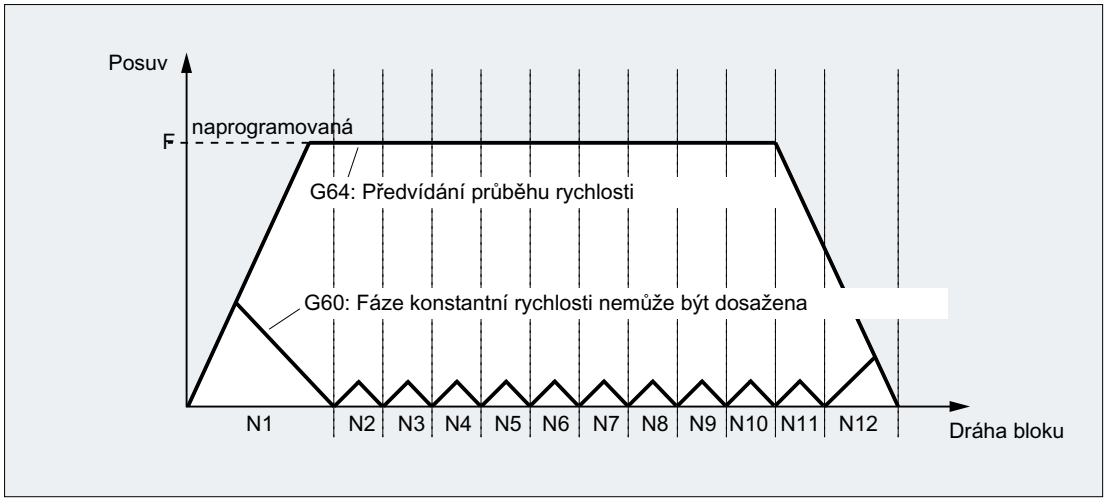
Jestliže v příkazu FGROUP nejsou obsaženy všechny dráhové osy, často se vyskytne skoková změna rychlosti na hranicích bloku u os, kterou jsou z FGROUP vyloučeny. Řídící systém omezuje tuto změnu rychlosti na přípustné hodnoty nastavené pomocí strojních parametrů MD32300 $MA_MAX_AX_ACCEL a MD32310 $MA_MAX_ACCEL_OVL_FACTOR. Této brzdící operaci je možné zabránit uplatnění funkce, která pomocí zaoblení "vyhlazuje" specifické poziční vzájemné vztahy mezi dráhovými osami.

Předvídání průběhu rychlosti (funkce Look Ahead)

V režimu řízení pohybu po dráze řídící systém automaticky zjišťuje na několik NC-bloků dopředu, jak bude vypadat průběh rychlosti. Jsou-li přechody aspoň přibližně tangenciální, umožňuje to zrychlování a zpomalování na více blocích.

Především pohybové řetězce, jež se skládají z krátkých úseků dráhy, se dají díky předvídání průběhu rychlosti obrábět s vyšším posuvem po dráze.

Maximální počet NC bloků, se kterými funkce Look Ahead pracuje, je možné nastavit ve strojním parametru.



Režim řízení pohybu po dráze s přechodovými zaobleními podle kritéria dráhy (G641)

Když je aktivní příkaz G641, vkládá řídící systém na konturových přechodech přechodové prvky. Prostřednictvím vzdálenosti zaoblení ADIS (příp. ADISPOS v případě G0) se zadává, jaké je maximální přípustné zaoblení v rozích. V rámci vzdálenosti (délky) přechodového zaoblení může řídící systém ignorovat dráhové vztahy a nahradit je dynamicky optimalizovanou dráhou.

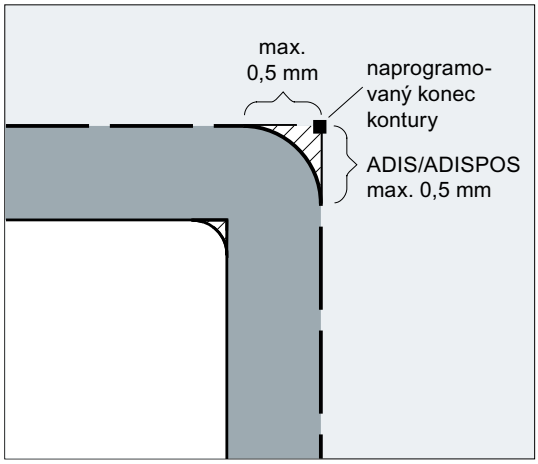
Nevýhoda: Pro všechny osy je k dispozici jen jedna hodnota parametru ADIS.

Příkaz G641 se chová podobně jako příkaz RNDM, není však omezen na osy pracovní roviny.

Stejně jako příkaz G64 pracuje i příkaz G641 s předvídáním průběhu rychlosti (funkce Look Ahead). Na bloky přechodového zaoblení s vyšším zakřivením se bude najíždět sníženou rychlostí.

Příklad:

Programový kód	**Komentář**
`N10 G641 ADIS=0.5 G1 X... Y...`	`; blok přibližného polohování může začínat nejdřív 0,5 mm před naprogramovaným koncem bloku a musí být ukončen 0,5 mm po konci bloku Toto nastavení má modální platnost.`



Poznámka

Přechodová zaoblení nemohou a nemají nahrazovat funkce pro definované vyhlazení (RND, RNDM, ASPLINE, BSPLINE, CSPLINE).

Přechodová zaoblení s axiálním přesným najetím s příkazem G642

Je-li aktivní příkaz G642, neuskutečňují se přechodová zaoblení v rámci oblasti definované příkazem ADIS, nýbrž zůstávají dodrženy osové tolerance definované strojním parametrem MD33100 $MA_COMPRESS_POS_TOL. Dráha přechodového zaoblení se vypočítá na základě nejkratšího zaoblení pro všechny osy. Z této hodnoty se vychází při vytváření bloku přechodového zaoblení.

Interní blokové přechodové zaoblení pomocí příkazu G643

Při definici přechodových zaoblení pomocí příkazu G643 jsou pro každou osu prostřednictvím strojního parametru MD33100 $MA_COMPRESS_POS_TOL definovány maximální odchylky od přesné kontury.

Když je aktivní příkaz G643, nevzniká žádný vlastní blok přechodového prvku, ale pro každou osu se vkládají interní blokové přechodové pohyby. Je-li aktivní příkaz G643, může být dráha přechodového zaoblení pro každou osu jiná.

Přechodová zaoblení s tolerancí kontury a orientace u příkazů G642/G643

Pomocí strojního parametru MD20480 $MC_SMOOTHING_MODE je možné v konfiguraci nastavit přechodová zaoblení s G642 a G643 tak, aby namísto tolerancí pro jednotlivé osy bylo možné zadat toleranci kontury a toleranci orientace.

Tolerance kontury a tolerance orientace se nastavují v kanálových nastavovaných parametrech:

SD42465 $SC_SMOOTH_CONTUR_TOL (maximální odchylka od kontury)

SD42466 $SC_SMOOTH_ORI_TOL (maximální úhlová odchylka orientace nástroje)

Nastavované parametry mohou být naprogramovány v NC programu a je tedy možné, aby byly pro každý přechod mezi bloky zadány odlišně. Velké rozdíly v nastavení pro toleranci kontury a toleranci orientace se mohou uplatňovat pouze ve spojení s příkazem G643.

Poznámka

Rozšíření o toleranci kontury a toleranci orientace existuje jedině v systémech, v nichž je k dispozici volitelný doplněk "Polynomická interpolace".

Poznámka

Aby bylo možné pracovat s přechodovými zaobleními při dodržení tolerance orientace, musí být aktivní transformace orientace.

Zaoblení přechodů s maximální možnou dynamikou s příkazem G644

Konfigurace přechodových zaoblení s maximální možnou dynamikou se nastavuje pomocí strojního parametru MD20480 $MC_SMOOTHING_MODE na místě tisíců.

Hodnota	Význam
0	Zadání maximální osové odchylky prostřednictvím: MD33100 $MA_COMPRESS_POS_TOL
1	Zadání maximální dráhy přechodového zaoblení naprogramováním parametru: ADIS=... příp. ADISPOS=...
2	Zadání maximálních možných frekvencí pro každou osu v oblasti přechodového zaoblení pomocí parametru: MD32440 $MA_LOOKAH_FREQUENCY Oblast přechodového zaoblení je definována tak, že dokud probíhá pohyb vytvářející zaoblení, nesmí se vyskytnout žádné frekvence překračující specifikované maximum.
3	U přechodových zaoblení s příkazem G644 nejsou monitorovány ani tolerance, ani vzdálenost přechodového zaoblení. Všechny osy se pohybují okolo rohu s maximální možnou dynamikou. Když je použit příkaz SOFT, jsou dodržovány jak maximální zrychlení, tak i maximální možný ryv pro každou osu Když je aktivní příkaz BRISK, není ryv nijak omezen; místo toho se každá osa pohybuje s maximálním možným zrychlením.

Tangenciální přechodová zaoblení na přechodech mezi bloky s příkazem G645

Pohyb na přechodových zaobleních, když je použit příkaz G645 je definován tak, aby u všech os, které se na pohybu podílejí, nedocházelo k žádným skokovým změnám zrychlení a aby nedošlo k překročení maximálních odchylek od původní kontury stanovených ve strojním parametru (MD33120 $MA_PATH_TRANS_POS_TOL).

V případě zlomových netangenciálních přechodů mezi bloky je chování přechodového zaoblení stejné jako v případě příkazu G642.

Žádné vkládané bloky přechodových zaoblení

V následujících případech se nevkládá žádný pomocný blok přechodového zaoblení:

- Mezi oběma bloky je zastavení.
  K tomu dojde za následujících okolností:
  - Následující blok obsahuje výstup pomocné funkce před pohybem.
  - Následující blok neobsahuje žádné pohyby po dráze.
  - Osa, která předtím byla polohovací osou, se v následujícím bloku poprvé pohybuje jako dráhová osa.
  - Osa, která předtím byla dráhovou osou, se v následujícím bloku poprvé pohybuje jako polohovací osa.
  - V přecházejícím bloku se prováděl pohyb geometrickými osami a v následujícím bloku nikoli.
  - V následujícím bloku se provádí pohyb geometrickými osami a v přecházejícím bloku nikoli.
  - Následující blok má jako podmínku dráhy řezání závitu s příkazem G33 a předešlý blok nikoli.
  - Došlo k přepnutí mezi režimy BRISK a SOFT.
  - Osy podílející se na transformaci nejsou úplně přiřazeny pohybu po dráze (např. při oscilačním pohybu, polohování os atd.).
- Blok přechodového zaoblení by způsobil zpomalení zpracování výrobního programu.
  K němu dojde za následujících okolností:
  - Bloky jsou velmi krátké.
    Protože každý blok vyžaduje aspoň jeden takt interpolátoru, vložení tohoto bloku by způsobilo zdvojnásobení doby zpracování.
  - Přechod mezi bloky s příkazem G64 (režim řízení pohybu po dráze bez přechodových zaoblení) může být uskutečněn bez snížení rychlosti.
    Zaoblení by zvýšilo dobu obrábění. To znamená, že hodnota přípustného faktoru přetížení (MD32310 $MA_MAX_ACCEL_OVL_FACTOR) rozhoduje, zda přechod mezi bloky bude či nebude zaoblený. Faktor přetížení se bere v úvahu jen u přechodových zaoblení ve spojení s příkazy G641 / G642. V případě přechodových zaoblení s příkazem G643 nemá faktor přetížení žádný vliv (toto chování může být nastaveno také pro příkazy G641 a G642 tak, že se nastaví MD20490 $MC_IGNORE_OVL_FACTOR_FOR_ADIS = TRUE).

- Přechodové zaoblení nemá nastaveny příslušné parametry.
  K tomu dojde za následujících okolností:
  - Když je aktivován příkaz G641 v blocích s G0 a ADISPOS=0 (předdefinované nastavení!).
  - Když je aktivován příkaz G641 v blocích bez G0 a ADIS=0 (předdefinované nastavení!).
  - Když je aktivován příkaz G641 na přechodech s G0 a bez G0 nebo bez G0 a s G0 platí menší hodnota z hodnot ADIS a ADISPOS.
  - U příkazů G642/G643, jestliže všechny specifické osové tolerance jsou rovny nule.
- Blok neobsahuje žádné příkazy pohybu (nulový blok).
  K tomu dojde za následujících okolností:
  - Jsou aktivní synchronní akce.
    Za normálních okolností jsou nulové bloky překladačem odstraňovány. Jestliže je ale aktivní synchronní akce, je tento nulový blok zřetězen a zpracován. Přitom se pracuje s přesným najetím, které je momentálně v programu aktivní. Tímto způsobem získává synchronní akce možnost se v případě potřeby deaktivovat.
  - Nulové bloky vznikají v důsledku programových skoků.

Řízení pohybu po dráze s rychlým posuvem G0

Také pro pohyby rychlým posuvem musí být specifikována jedna z funkcí G60/G9 nebo G64 příp. G641 - G645. Jinak se použije předdefinované nastavení podle strojního parametru.

3.12 Transformace souřadného systému (Frame)

3.12.1 Framy

Frame

Framem se rozumí uzavřený matematický předpis, který převádí jeden kartézský souřadný systém do jiného kartézského souřadného systému.

Základní frame (základní posunutí)

Základní frame popisuje transformaci souřadného systému ze základního souřadného systému (BCS) do základního počátečního systému (BZS) a chová se stejně jako nastavitelné framy.

Viz Základní souřadný systém (BCS) (Strana 40).

Nastavitené framy

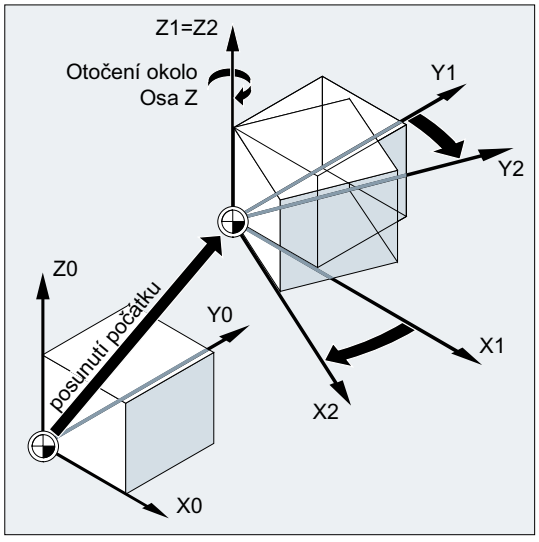
Nastavitelné framy jsou nastavitelná posunutí počátku (nulového bodu), která mohou být v libovolném NC-programu vyvolávána příkazy `G54` až `G57` a `G505` až `G599`. Hodnoty posunutí jsou obsluhujícím pracovníkem předem definovány a uloženy v paměti posunutí počátku řídícího systému. Jejich prostřednictvím je určen nastavitelný počátek (nulový bod) systému (ENS).

Viz:

- Nastavitelný souřadný systém (ENS) (Strana 41)
- Nastavitelná posunutí počátku (G54 ... G57, G505 ... G599, G53, G500, SUPA, G153) (Strana 144)

Programovatelné framy

Občas se jeví jako výhodné nebo dokonce nezbytné uvnitř jednoho NC-programu dříve zvolený počátek souřadného systému obrobku (příp. "Nastavitelný souřadný systém") posunout na jiné místo, případně systém pootočit, zrcadlově jej převrátit / změnit měřítko os. To se uskutečňuje prostřednictvím programovatelných framů.



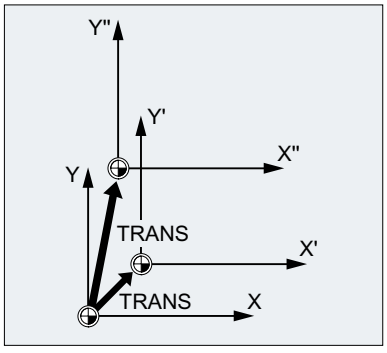
Viz Příkazy framů (Strana 309).

3.12.2 Příkazy framů

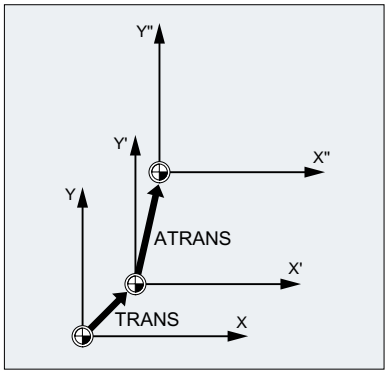
Funkce

Příkazy pro programovatelné framy platí v momentálně zpracovávaném NC programu. Uplatňují se buď jako aditivní nebo jako nahrazující příkazy:

- Nahrazující příkaz
  Všechny dříve naprogramované příkazy framu se vymažou. Jako vztažné se používá naposled zadané nastavitelné posunutí počátku (G54 ... G57, G505 ... G599).

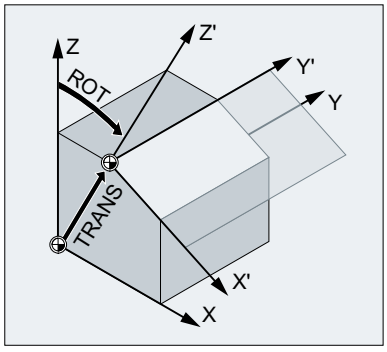


- Aditivní příkaz
  Superponuje se na již existující frame. Jako vztažný slouží právě nastavený nebo přes příkaz framu naposled naprogramovaný počátek souřadného systému obrobku.



Příklad použití

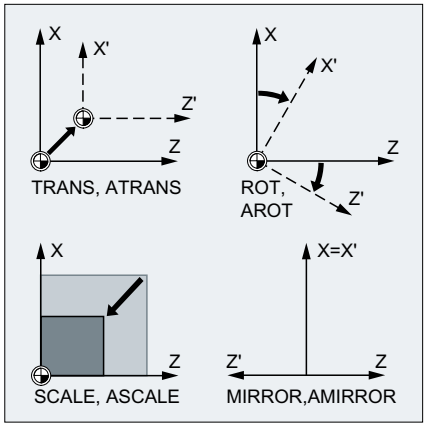
1. Posunutí počátku (nuly) souřadného systému obrobku (WCS)
2. Otočení souřadného systému obrobku (WCS) tak, aby jeho rovina byla nastavena rovnoběžně s požadovanou pracovní rovinou.

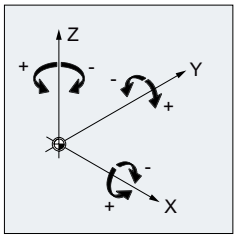


Syntaxe

Ersetzende Anweisungen	Aditivní příkazy
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

Význam



TRANS/ATRANS:	Posunutí WCS ve směru uvedených geometrických os		
ROT/AROT:	Otáčení WCS: • prostřednictvím zřetězení jednotlivých otočení okolo uvedených geometrických os nebo • o úhel RPL=... v aktuální pracovní rovině (G17/G18/G19)		
	Směr opisování:		
	Posloupnost při otáčení:	v případě zápisu typu RPY:	Z, Y', X"
		pomocí Eulerova úhlu:	Z, X', Z"
	Rozsah hodnot:	Úhly otočení jsou jednoznačně definovány pouze v následujících rozsazích:	
		v případě zápisu typu RPY:	-180 ≤ x ≤ 180
			-90 < y < 90
			-180 ≤ z ≤ 180
		pomocí Eulerova úhlu:	0 ≤ x < 180
			-180 ≤ y ≤ 180
			-180 ≤ z ≤ 180
ROTS/AROTS:	Otočení WCS pomocí udání prostorových úhlů Orientace roviny v prostoru je jednoznačně určena zadáním dvojice prostorových úhlů. Z tohoto důvodu smí být naprogramovány maximálně 2 prostorové úhly: ROTS/AROTS X... Y... / Z... X... / Y... Z...		
CROTS:	Příkaz CROTS se chová stejně jako příkaz ROTS, vztahuje se však na platný frame ve správě dat.		
SCALE/ASCALE:	Změna měřítka ve směru uvedených geometrických os za účelem zvětšení/zmenšení kontury		
MIRROR/AMIRROR:	Zrcadlové převrácení WCS prostřednictvím zrcadlového převrácení (změny směru) uvedené geometrické osy		
	Hodnota:	libovolně nastavitelná (zde: "0")	

Okrajové podmínky

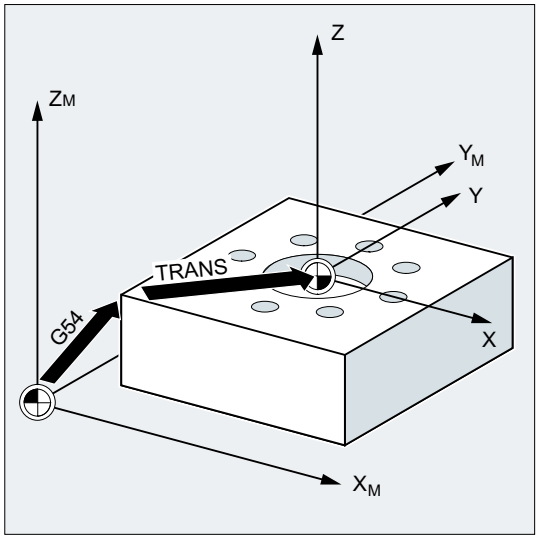
- Příkazy framu musí být naprogramovány v samostatném NC-bloku.
- Příkazy framů se mohou používat jednotlivě nebo mohou být libovolně kombinovány.
- Příkazy framů se uskutečňují v naprogramovaném pořadí.
- Aditivní příkazy se často používají v podprogramech. Základní příkazy definované v základním programu zůstávají po skončení podprogramu zachovány, jestliže byl podprogram sestaven s atributem SAVE.

3.12.3 Programovatelné posunutí počátku (TRANS, ATRANS)

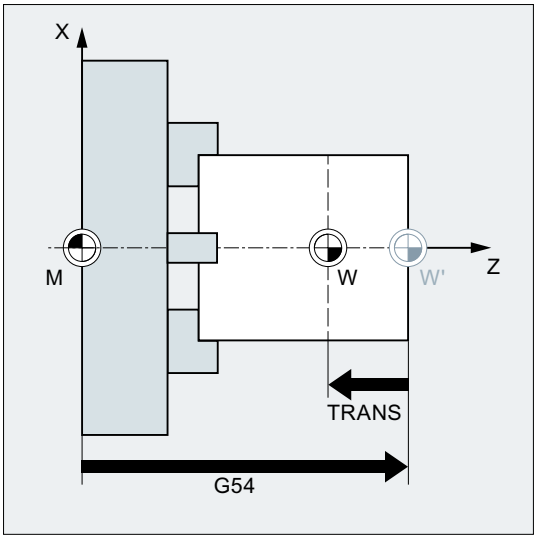
Prostřednictvím příkazu `TRANS` se uskutečňuje absolutní posunutí WCS, které je vztaženo na systém ENS vytvořený prostřednictvím nastavitelného posunutí počátku (G54 ... G57, G505 ... G599).

Pomocí příkazu `ATRANS` se systém WCS vytvořený příkazem `TRANS` aditivně posunuje.

Frézování:



Soustružení:



Syntaxe

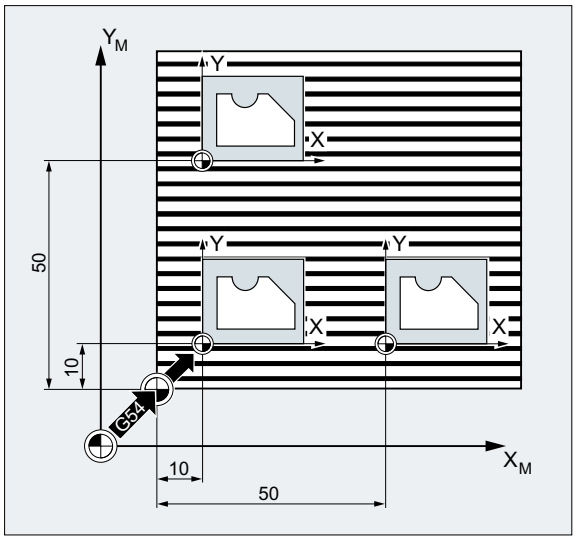
```
TRANS X... Y... Z...
ATRANS X... Y... Z...
```

Význam

`TRANS`:	Absolutní posunutí WCS vztažené na počátek souřadné soustavy obrobku (ENS) definované prostřednictvím nastavitelného posunutí počátku (nuly) (G54 ... G57, G505 ... G599)	
	Samostatný blok:	ano
`ATRANS`:	Aditivní posunutí počátku (nuly) WCS vztažené na počátek souřadného systému obrobku nastaveného pomocí příkazu `TRANS`	
	Samostatný blok:	ano
`X... Y... Z...`:	Hodnoty posunutí ve směru uvedených geometrických os	

Příklady

Příklad 1: Frézování



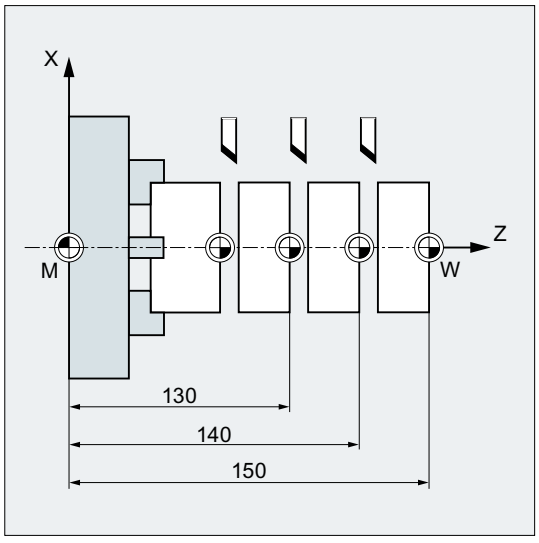
U tohoto obrobku se zobrazované tvary vyskytují v programu vícekrát.

Posloupnost obrábění je pro tento tvar uložena v podprogramu.

Pomocí posunutí počátku jsou definována pouze potřebná posunutí počátků a pak je vyvoláván podprogram.

Programový kód	**Komentář**
N10 G1 G54	; pracovní rovina X/Y, počátek souřadného systému obrobku
N20 G0 X0 Y0 Z2	; najetí na počáteční bod
N30 TRANS X10 Y10	; absolutní posunutí
N40 L10	; volání podprogramu
N50 TRANS X50 Y10	; absolutní posunutí
N60 L10	; volání podprogramu
N70 M30	; konec programu

Příklad 2: Soustružení



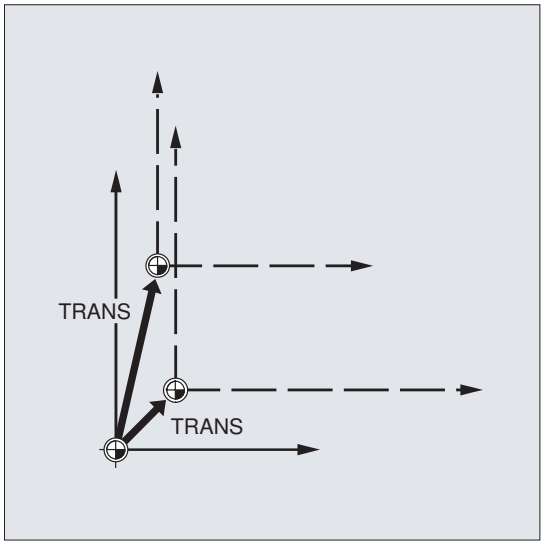
Programový kód	Komentář
...	
N10 TRANS X0 Z150	; absolutní posunutí
N15 L20	; volání podprogramu
N20 TRANS X0 Z140 (nebo ATRANS Z-10)	; absolutní posunutí
N25 L20	; volání podprogramu
N30 TRANS X0 Z130 (nebo ATRANS Z-10)	; absolutní posunutí
N35 L20	; volání podprogramu
...	

Další informace

TRANS X... Y... Z...

Posunutí počátku o naprogramované hodnoty posunutí ve směrech specifikovaných os (dráhové, synchronizované a polohovací osy). Jako vztažné se používá naposled zadané nastavitelné posunutí počátku (G54 ... G57, G505 ... G599).

UPOZORNĚNÍ
Žádný počáteční frame
Příkaz `TRANS` zruší veškeré komponenty dříve definovaného programovatelného framu.

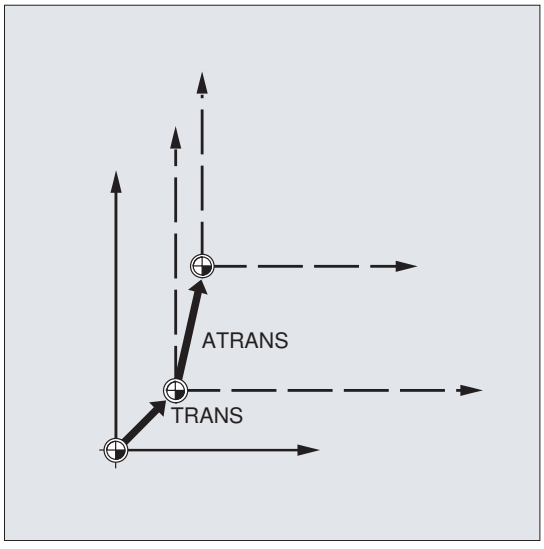


Poznámka

Pokud budete chtít naprogramovat posunutí, které se bude přičítat k už existujícím framům, můžete použít příkaz ATRANS.

ATRANS X... Y... Z...

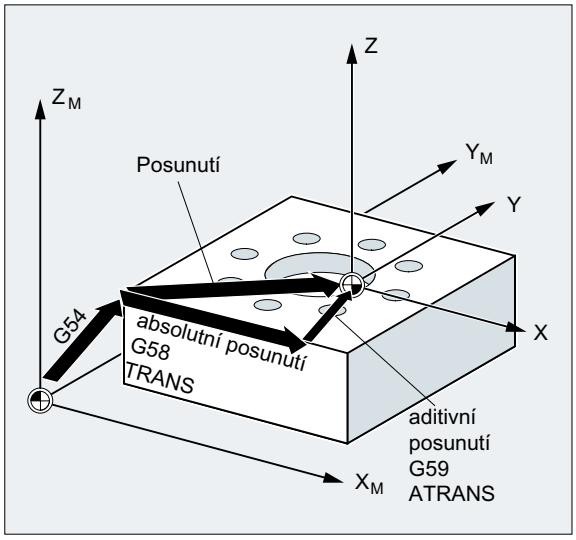
Posunutí počátku o naprogramované hodnoty ve směrech specifikovaných os. Jako vztažný bod se používá v daném okamžiku nastavený nebo naposled naprogramovaný počátek.



3.12.4 Programovatelné posunutí počátku (G58, G59)

Pomocí funkcí G58 a G59 mohou být pro specifickou osu nahrazovány translační složky programovatelného posunutí počátku (TRANS/ATRANS) (Strana 312):

- G58: absolutní složka posunutí (hrubé posunutí)
- G59: aditivní složka posunutí (jemné posunutí)



Předpoklady

Funkce G58 a G59 je možno používat jedině tehdy, pokud je v konfiguraci aktivováno jemné posunutí (MD24000 $MC_FRAME_ADD_COMPONENTS = 1).

Syntaxe

```
G58 <osa_1><hodnota_1> ... <osa_3><hodnota_3>
G59 <osa_1><hodnota_1> ... <osa_3><hodnota_3>
```

Význam

`G58`:	Příkazem `G58` se nahrazuje aditivní složka posunutí programovatelného posunutí počátku pro specifikovanou osu, ale naprogramované aditivní posunutí zůstává v zachováno. Jako vztažné se používá naposled zadané nastavitelné posunutí počátku (`G54` ... `G57`, `G505` ... `G599`).
	Samostatný blok: ano
`G59`:	Příkazem `G59` se nahrazuje aditivní složka posunutí programovatelného posunutí počátku pro uvedenou osu, ale naprogramované absolutní posunutí zůstává v zachováno.
	Samostatný blok: ano
`<osa_n>`:	Geometrická osa v kanálu
`<hodnota_n>`:	Hodnoty posunutí ve směru uvedené geometrické osy

Příklad:

```
Programový kód              Komentář
...
N50 TRANS X10 Y10 Z10       ; absolutní složka posunutí X10 Y10 Z10
N60 ATRANS X5 Y5            ; aditivní složka posunutí X5 Y5
                            ==> Celkové posunutí: X15 Y15 Z10
N70 G58 X20                 ; absolutní složka posunutí X20
                            ==> Celkové posunutí X25 Y15 Z10
N80 G59 X10 Y10             ; aditivní složka posunutí X10 Y10
                            ==> Celkové posunutí X30 Y20 Z10
...
```

Další informace

Absolutní složku posunutí (**hrubé posunutí**) je možno změnit pomocí následujících příkazů:

- TRANS
- G58
- CTRANS
- CFINE
- $P_PFRAME[X,TR]

Aditivní složku posunutí (**jemné posunutí**) je možno změnit pomocí následujících příkazů:

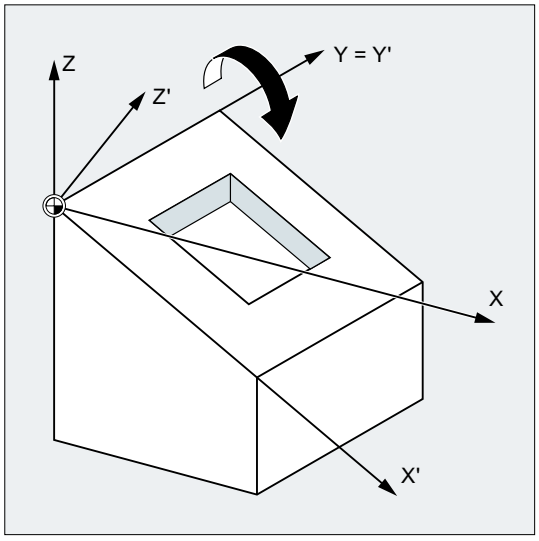
- ATRANS
- G59
- CTRANS
- CFINE
- $P_PFRAME[X,FI]

Příklady

Příkaz	Hrubé posunutí V_C	Jemné posunutí V_F
TRANS X10	$V_C = 10$	nezměněno
G58 X10	$V_C = 10$	nezměněno
$P_PFRAME[X,TR] = 10	$V_C = 10$	nezměněno
ATRANS X10	nezměněno	$V_F = V_F + 10$
G59 X10	nezměněno	$V_F = 10$
$P_PFRAME[X,FI] = 10	nezměněno	$V_F = 10$
CTRANS(X,10)	$V_C = 10$	$V_F = 0$
CTRANS()	$V_C = 0$	$V_F = 0$
CFINE(X,10)	$V_C = 0$	$V_F = 10$

3.12.5 Programovatelné otočení (ROT, AROT, RPL)

Pomocí příkazů `ROT` / `AROT` může být souřadný systém obrobku otáčen v prostoru. Příkazy se vztahují výlučně na programovatelný frame `$P_PFRAME`.



Syntaxe

```
ROT <1. geometrická osa><úhel> <2. geometrická osa><úhel> <3.
geometrická osa><úhel>
ROT RPL=<úhel>
AROT <1. geometrická osa><úhel> <2. geometrická osa><úhel> <3.
geometrická osa><úhel>
AROT RPL=<úhel>
```

Poznámka

Otáčení souřadného systému obrobku se uskutečňuje pomocí Eulerova úhlu.

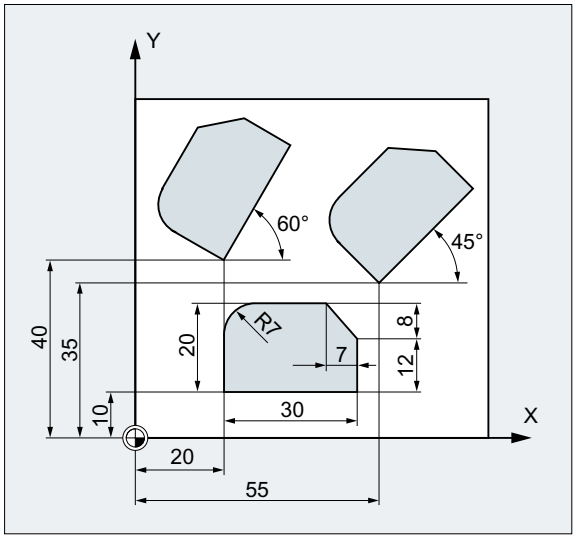
Význam

`ROT`:	Úhel otočení absolutně	
	Vztažný frame:	Programovatelný frame `$P_PFRAME`
	Vztažný bod:	Počátek aktuálního souřadného systému obrobku definovaný příkazy `G54` ... `G57`, `G505` ... `G599`
`AROT`:	Aditivní otočení	
	Vztažný frame:	Programovatelný frame `$P_PFRAME`
	Vztažný bod:	Počátek aktuálního souřadného systému obrobku definovaný příkazy `G54` ... `G57`, `G505` ... `G599`
`<n-tá GeoOsa>`:	Identifikátor n-té geometrické osy, okolo které se má otáčení o uvedený úhel uskutečnit. Pro nenaprogramované geometrické osy se jako úhel otočení implicitně dosazuje hodnota 0°.	

RPL:	Otočení o zadaný úhel okolo geometrické osy, která je kolmá k aktivní rovině (G17, G18, G19).	
	Vztažný frame:	Programovatelný frame $P_PFRAME
	Vztažný bod:	Počátek aktuálního souřadného systému obrobku definovaný příkazy G54 ... G57, G505 ... G599
<úhel>	Údaj úhlu ve stupních	
	Rozsah hodnot:	-360° ≤ úhel ≤ 360°

Příklady

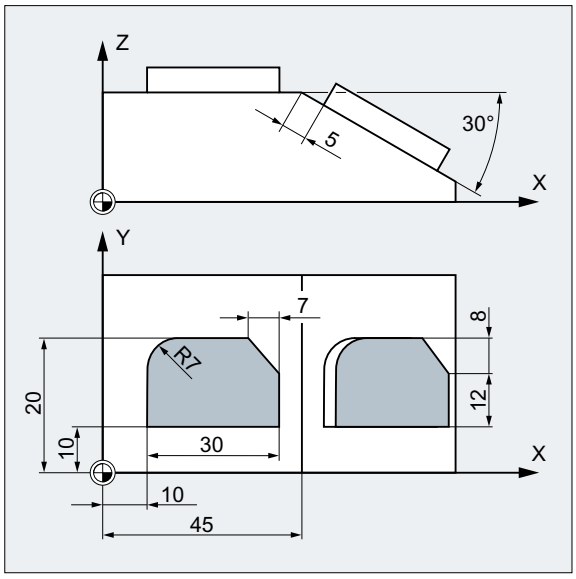
Příklad 1: Otočení v rovině G17



U tohoto obrobku se zobrazované tvary vyskytují v programu vícekrát. Kromě posunutí počátku musí být uskutečněno pootočení, protože tvary nejsou uspořádány rovnoběžně s osami.

Programový kód	Komentář
N10 G17 G54	; pracovní rovina X/Y, počátek souřadného systému obrobku
N20 TRANS X20 Y10	; absolutní posunutí
N30 L10	; volání podprogramu
N40 TRANS X55 Y35	; absolutní posunutí
N50 AROT RPL=45	; aditivní otočení okolo osy Z, která je
	; kolmá k rovině G17, o 45°
N60 L10	; volání podprogramu
N70 TRANS X20 Y40	; absolutní posunutí
	; (vynuluje všechna předešlá posunutí)
N80 AROT RPL=60	; aditivní otočení okolo osy Z, která je
	; kolmá k rovině G17, o 60°
N90 L10	; volání podprogramu
N100 G0 X100 Y100	; odjíždění nástroje
N110 M30	; konec programu

Příklad 2: Prostorové otočení okolo osy Y

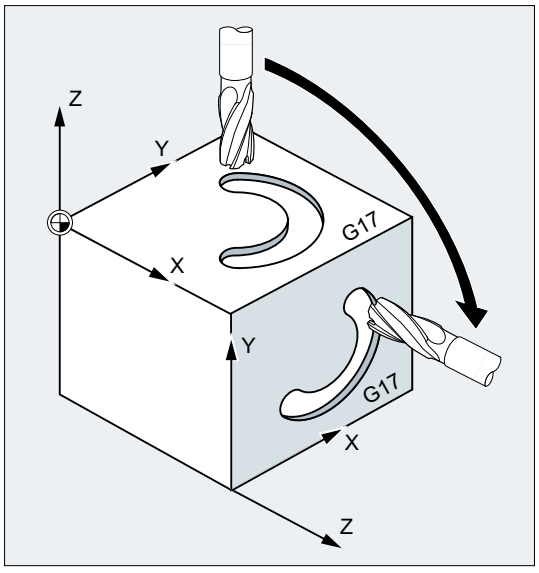


V tomto příkladu mají být při jednom upnutí obrobeny plochy obrobku ležící rovnoběžně s osou a ležící šikmo.

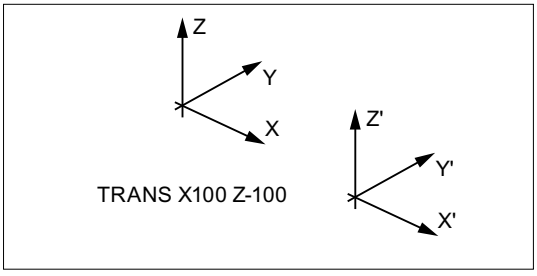
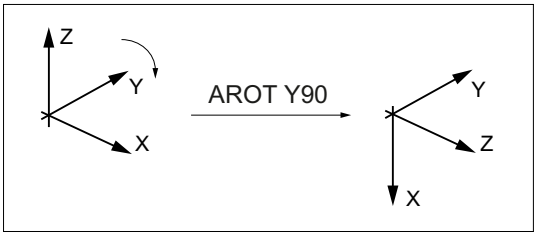
Předpoklad:
Nástroj musí být v pootočeném směru Z nastaven kolmo k šikmé ploše.

Programový kód	**Komentář**
N10 G17 G54	; pracovní rovina X/Y, počátek souřadného systému obrobku
N20 TRANS X10 Y10	; absolutní posunutí
N30 L10	; volání podprogramu
N40 ATRANS X35	; aditivní posunutí
N50 AROT Y30	; aditivní otočení okolo osy Y
N60 ATRANS X5	; aditivní posunutí
N70 L10	; volání podprogramu
N80 G0 X300 Y100 M30	; odjíždění nástroje, konec programu

Příklad 3: Obrábění na více stranách



V tomto příkladu jsou prostřednictvím podprogramu obráběny identické tvary nacházející se na dvou na sebe kolmých plochách obrobku. V novém souřadném systému na ploše obrobku na pravé straně jsou směr přísuvu, pracovní plocha a počátek uspořádány stejně jako na horní ploše. Díky tomu dále platí podmínky, které jsou potřebné pro zpracování podprogramu: Pracovní rovina `G17`, souřadná rovina X/Y, směr přísuvu Z.

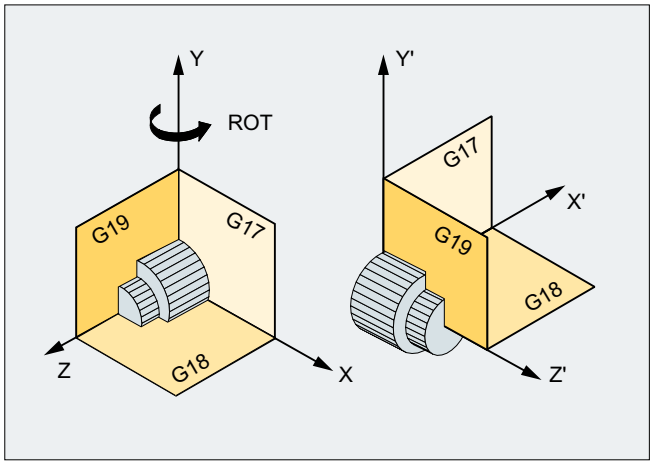
Programový kód	**Komentář**
`N10 G17 G54`	`; pracovní rovina X/Y, počátek souřadného systému obrobku`
`N20 L10`	`; volání podprogramu`
`N30 TRANS X100 Z-100`	`; absolutní posunutí WCS`
	
`N40 AROT Y90`	`; aditivní otočení WCS okolo osy Y o 90°`
	

Programový kód	Komentář
N50 AROT Z90	; aditivní otočení WCS okolo osy Z o 90°
N60 L10	; volání podprogramu
N70 G0 X300 Y100 M30	; odjíždění nástroje, konec programu

Další informace

Otáčení v aditivní rovině

Při programování prostřednictvím příkazu RPL=... se WCS otáčí okolo osy, která je kolmá k aktivní rovině.



Obrázek 3-23 Otáčení okolo osy Y, příp. v rovině G18

⚠ VÝSTRAHA

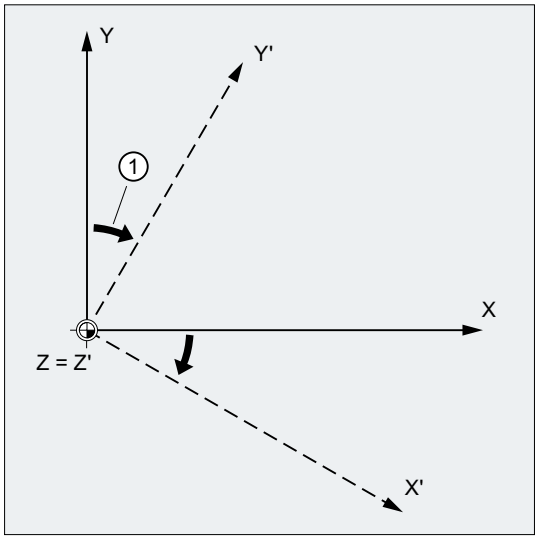
Změna roviny

Jestliže po otočení je naprogramováno přepnutí roviny (G17, G18, G19), zůstává momentální úhel otočení pro jednotlivé osy zachován a uplatňuje se i v nové rovině. Z tohoto důvodu Vám rozhodně doporučujeme před přepínáním roviny momentální úhly otočení vynulovat (dosadit hodnotu "0").

- N100 ROT X0 Y0 Z0 ; explicitní programování úhlu
- N100 ROT ; implicitní programování úhlu

Absolutní otáčení pomocí příkazu ROT X... Y... Z...

WCS bude pootočen okolo zvolené osy do naprogramovaného úhlu.

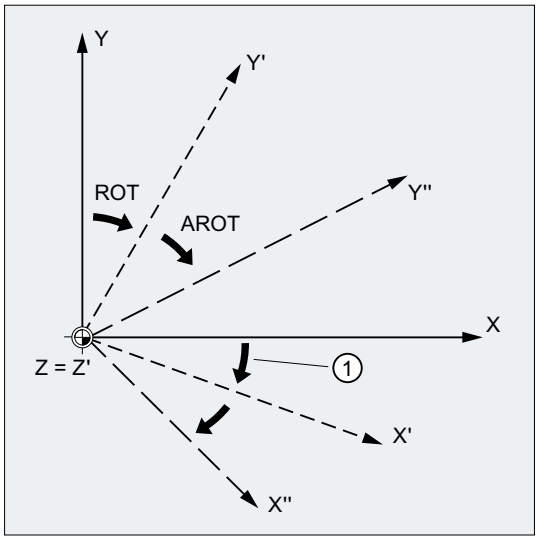


① Úhel otočení

Obrázek 3-24 Absolutní otočení okolo osy Z

Aditivní otáčení pomocí příkazu AROT X... Y... Z...

WCS bude pootočen dále okolo zvolené osy o naprogramovaný úhel.



① Úhel otočení

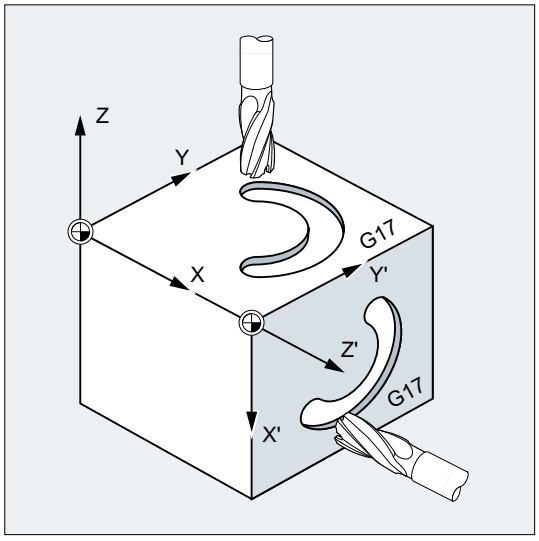
Obrázek 3-25 Absolutní a aditivní otočení okolo osy Z

Otáčení pracovní roviny

Při otáčení pomocí příkazů `ROT` / `AROT` se otáčí také pracovní rovina (`G17`, `G18`, `G19`).

Příklad: Pracovní rovina G17
Počátek WCS se nachází na horní ploše obrobku. Posunutím a otočením se souřadný systém přesouvá na jednu z bočních ploch. Pracovní rovina G17 se otáčí také. Díky tomu mohou být pracovní posuvy i nadále programovány v rovině G17 pomocí souřadnic X a Y a přísuvy pomocí souřadnice Z.

Předpoklad:
Nástroj se musí nacházet kolmo na pracovní rovinu a kladný směr přísuvné osy je orientován k držáku nástroje.
Zadáním příkazu `CUT2DF` bude korekce rádiusu nástroje aplikována i v otočené rovině.



3.12.6 Programové otočení framu o prostorový úhel (ROTS, AROTS, CROTS)

Pomocí příkazů `ROTS`, `AROTS` a `CROTS` můžete otáčení souřadného systému obrobku zadávat pomocí prostorových úhlů. Prostorové úhly jsou úhly, které tvoří řezové přímky požadované roviny otočené v prostoru s hlavními rovinami dosud neotočeného WCS.

Poznámka

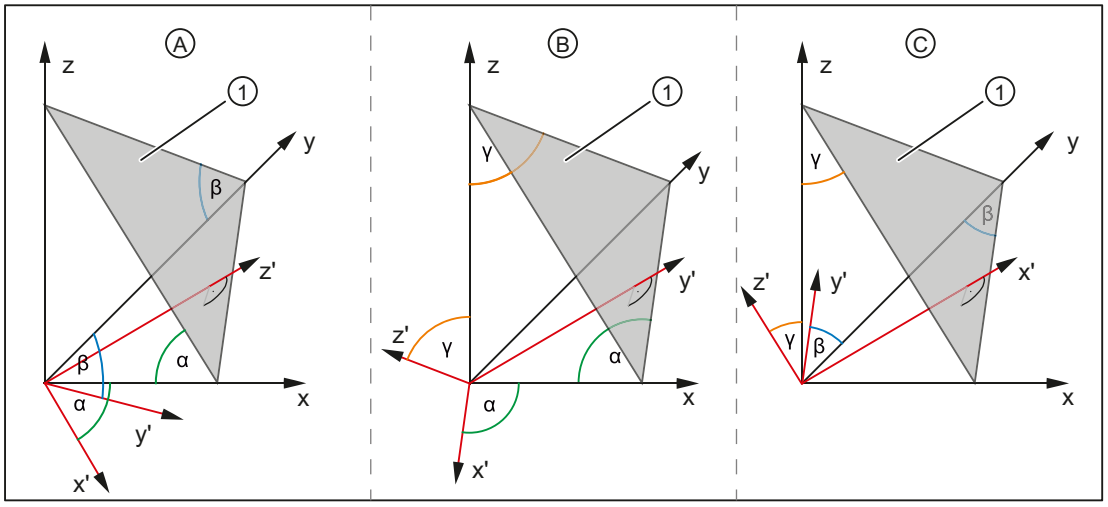
Identifikátory geometrických os

Pro účely dalšího popisu budeme používat například následující definice:

- 1. geometrická osa X
- 2. geometrická osa Y
- 3. geometrická osa Z

Například, programový příkaz `ROTS Xα Yβ` způsobí nasměrování roviny G17 WCS tak, aby byla rovnoběžně s uvedenou šikmou rovinou, jak ukazuje následující obrázek. Poloha počátku souřadného systému zůstává přitom nezměněna.

Orientace otočeného WCS je definována takovým způsobem, že první otočená osa leží v rovině, která je dána touto osou a 3. osou původního souřadného systému. V příkladu: Osa X leží v původní rovině X/Z.



①	Šikmá rovina
α, β, γ	Prostorový úhel
A	Nasměrování roviny G17 rovnoběžně s šikmou rovinou:

- 1. otočení
  Otočení osy X okolo osy Y o úhel α ⇒
  Osa X' rovnoběžně s šikmou rovinou
- 2. otočení
  Otočení osy Y' okolo X' o úhel β ⇒
  Osa Y' rovnoběžně s šikmou rovinou
  ⇒ Osa Z' kolmo na šikmou rovinu
  ⇒ G17 rovnoběžně s šikmou rovinou

B Nasměrování roviny G18 rovnoběžně s šikmou rovinou:

- 1. otočení
  Otočení osy Z okolo osy X o úhel γ ⇒
  Osa Z' rovnoběžně s šikmou rovinou
- 2. otočení
  Otočení osy X' okolo osy Z' o úhel α ⇒
  Osa X' rovnoběžně s šikmou rovinou
  ⇒ Osa Y' kolmo na šikmou rovinu
  ⇒ G18 rovnoběžně s šikmou rovinou

C Nasměrování roviny G19 rovnoběžně s šikmou rovinou:

- 1. otočení
  Otočení osy Y okolo osy Z o úhel β ⇒
  Osa Y' rovnoběžně s šikmou rovinou
- 2. otočení
  Otočení osy Z' okolo osy Y' o úhel γ ⇒
  Osa Z' rovnoběžně s šikmou rovinou
  ⇒ Osa X' kolmo na šikmou rovinu
  ⇒ G19 rovnoběžně s šikmou rovinou

Syntaxe

definice

Poloha roviny v prostoru je jednoznačně určena pomocí dvojice prostorových úhlů. Zadáním třetího úhlu by bylo určení roviny přílišné. Z tohoto důvodu je to nepřípustné.

Při naprogramování jen jednoho prostorového úhlu probíhá otáčení WCS zcela stejně, jako u příkazů `ROT`, `AROT` (viz kapitola "Programovatelné otočení (ROT, AROT, RPL) (Strana 318)").

Prostřednictvím obou naprogramovaných os je definována rovina v souladu s definicemi pro roviny `G17`, `G18`, `G19`. Tímto způsobem je definována posloupnost os souřadného systému (1. osa / 2. osa v rovině), příp. posloupnost otáčení o prostorový úhel:

Rovina	1. osa	2. osa
G17	X	Y
G18	Z	X
G19	Y	Z

Nastavení orientace roviny G17 => prostorové úhly pro osy X a Y

- 1. otočení: osa X okolo osy Y o úhel α
- 2. otočení: Osa Y okolo osy X' o úhel β
- Orientace: Osa X' leží v původní rovině Z/X.

```
ROTS  X<α> Y<β>
AROTS X<α> Y<β>
CROTS X<α> Y<β>
```

Nastavení orientace roviny G18 => prostorové úhly pro osy Z a X

- 1. otočení: Osa Z okolo osy X o úhel γ
- 2. otočení: osa X okolo osy Z' o úhel α
- Orientace: Osa Z' leží v původní rovině Y/Z.

```
ROTS  Z<γ> X<α>
AROTS Z<γ> X<α>
CROTS Z<γ> X<α>
```

Nastavení orientace roviny G19 => prostorové úhly pro osy Y a Z

- 1. otočení: Osa Y okolo osy Z o úhel β
- 2. otočení: Osa Z okolo osy Y' o úhel γ
- Orientace: Osa Y' leží v původní rovině X/Y.

```
ROTS  Y<β> Z<γ>
AROTS Y<β> Z<γ>
CROTS Y<β> Z<γ>
```

Význam

ROTS:	Absolutní otáčení framu pomocí prostorových úhlů, vztažný frame: Programovatelný frame $P_PFRAME
AROTS:	Aditivní otáčení framu pomocí prostorových úhlů, vztažný frame: Programovatelný frame $P_PFRAME
CROTS:	Absolutní otáčení framu pomocí prostorových úhlů, vztažný frame: programovatelný frame $P_...
X, Y, Z:	Identifikátor geometrické osy (viz upozornění uvedené výše: Identifikátory geometrických os)
α, β, γ:	Prostorový úhel vztažený na odpovídající geometrickou osu: • α --> X • β --> Y • γ --> Z

3.12.7 Programovatelná změna měřítka (SCALE, ASCALE)

Pomocí příkazů SCALE/ASCALE můžete naprogramovat pro všechny dráhové, synchronizované a polohovací osy faktory pro změnu měřítka za účelem zvětšení nebo zmenšení ve směru jednotlivých uváděných os. Tímto způsobem můžete při programování zohlednit např. geometricky podobné tvary nebo odlišné velikosti úbytků.

Syntaxe

```
SCALE X... Y... Z...
ASCALE X... Y... Z...
```

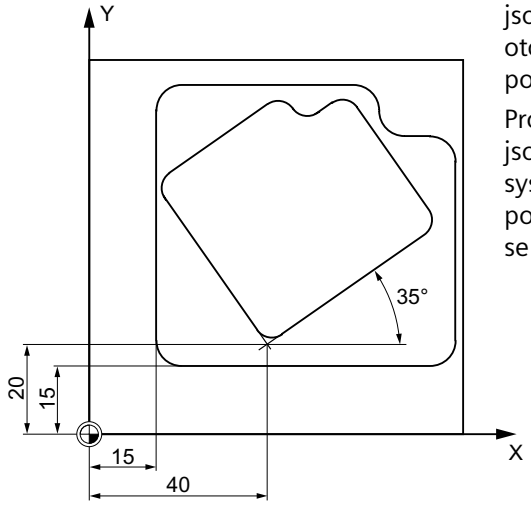
Poznámka

Příkazy framu musí být vždy naprogramovány v samostatném NC-bloku.

Význam

SCALE:	Absolutní zvětšení/zmenšení vztažené na právě platný souřadný systém nastavený pomocí příkazů G54 ... G57, G505 ... G599
ASCALE:	Aditivní zvětšení/zmenšení vztažené na právě platný nastavený nebo naprogramovaný souřadný systém
X... Y... Z...:	Faktory změny měřítka ve směru uvedených geometrických os

Příklad:



U tohoto obrobku se vyskytují dvě dutiny, které jsou však různé velikosti a jsou vůči sobě po-otočené. Postup obrábění je naprogramován v podprogramu.

Prostřednictvím posunutí počátku a otočení jsou definovány příslušné počátky souřadného systému obrobku, jak je zapotřebí. Přitom se pomocí změny měřítka kontura zmenší a pak se vyvolává podprogram znovu.

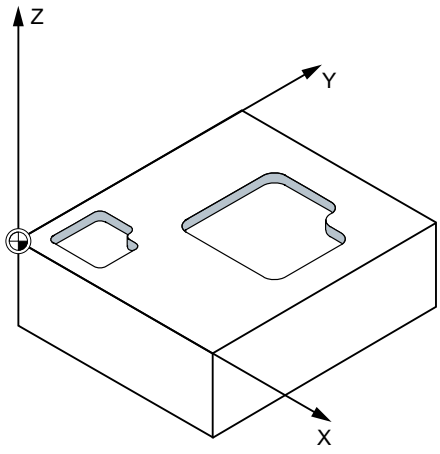
Programový kód	**Komentář**
N10 G17 G54	; pracovní rovina X/Y, počátek souřadného systému obrobku
N20 TRANS X15 Y15	; absolutní posunutí
N30 L10	; obrobení velké kapsy
N40 TRANS X40 Y20	; absolutní posunutí
N50 AROT RPL=35	; otočení v rovině o 35°
N60 ASCALE X0.7 Y0.7	; faktor změny měřítka pro malou kapsu
N70 L10	; obrobení malé kapsy
N80G0 X300 Y100 M30	; odjíždění nástroje, konec programu

Další informace

SCALE X... Y... Z...

Pro každou osu může být zadán její vlastní faktor změny měřítka, o který má být rozměr zvětšen nebo zmenšen. Změna měřítka se vztahuje na souřadný systém obrobku nastavený pomocí příkazů G54 ... G57, G505 ... G599.

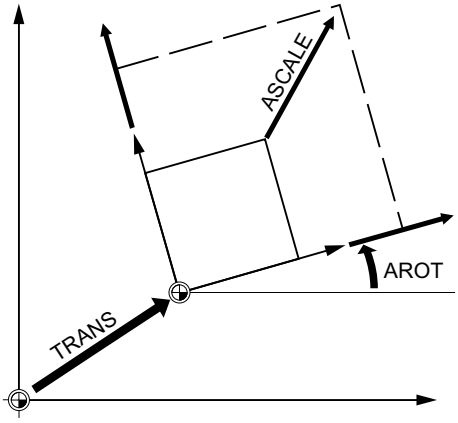
UPOZORNĚNÍ
Žádný počáteční frame
Příkaz SCALE zruší veškeré komponenty dříve definovaného programovatelného framu.



ASCALE X... Y... Z...

Pokud budete chtít naprogramovat faktor změny měřítka, který se bude přičítat k už existujícím framům, můžete použít příkaz `ASCALE`. V tomto případě bude naposled platná změna měřítka vynásobena novou změnou měřítka.

Jako vztažný souřadný systém pro změnu měřítka se bere právě nastavený nebo naposled naprogramovaný souřadný systém.



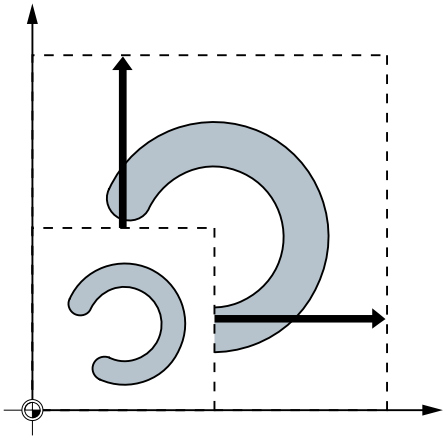
Změna měřítka a posunutí

Poznámka

Jestliže je po příkazu `SCALE` naprogramováno posunutí pomocí příkazu `ATRANS`, pak budou hodnoty posunutí rovněž podléhat změně měřítka.

Odlišné faktory změny měřítka

UPOZORNĚNÍ
Nebezpečí kolize Pozor při různých faktorech změny měřítka! Např. pro kruhovou interpolaci je možná změna měřítka pouze se stejnými faktory.



Poznámka

Pro programování deformovaných kruhů však mohou být odlišné faktory změny měřítka použity cíleně.

3.12.8 Programovatelné zrcadlové převrácení (MIRROR, AMIRROR)

Pomocí funkcí `MIRROR`/`AMIRROR` je možné vyrábět zrcadlově převrácené tvary obrobků na souřadných osách. Všechny pohyby pracovním posuvem, které jsou potom naprogramovány, např. v podprogramu, se budou provádět zrcadlové převrácené.

Syntaxe

```
MIRROR X... Y... Z...
AMIRROR X... Y... Z...
```

Poznámka

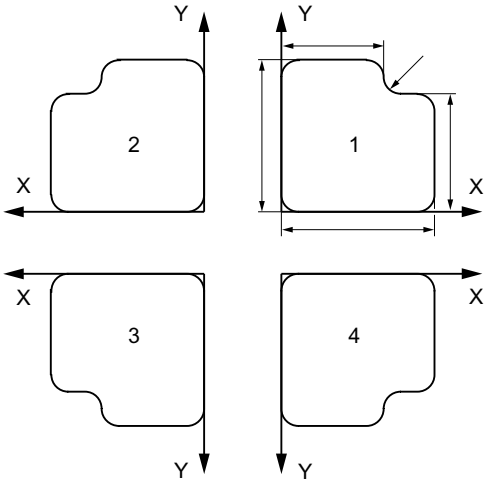
Příkazy framu musí být vždy naprogramovány v samostatném NC-bloku.

Význam

MIRROR:	Absolutní zrcadlové převrácení vztažené na právě platný souřadný systém nastavený pomocí příkazů G54 ... G57, G505 ... G599
AMIRROR:	Aditivní zrcadlové převrácení vztažené na právě platný nastavený nebo naprogramovaný souřadný systém
X... Y... Z...:	Geometrická osa, jejíž směry mají být přehozeny. Zde uváděná hodnota je libovolná, např. X0, Y0, Z0.

Příklady

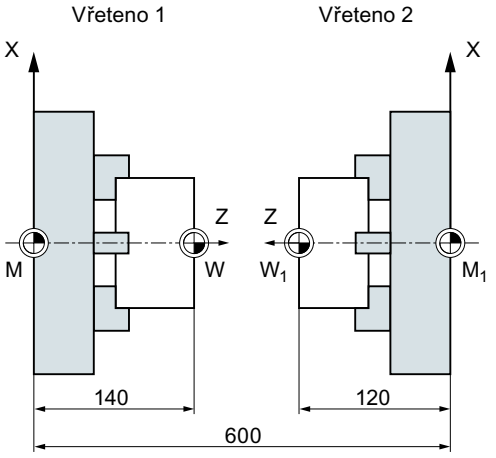
Příklad 1: Frézování



Zde zobrazovaná kontura je naprogramována jen jednou jako podprogram. Další tři kontury jsou vyrobeny pomocí zrcadlového převracení. Počátek souřadného systému obrobku je umístěn uprostřed kontur.

Programový kód	**Komentář**
N10 G17 G54	; pracovní rovina X/Y, počátek souřadného systému obrobku
N20 L10	; obrobení první kontury vpravo nahoře
N30 MIRROR X0	; zrcadlové převrácení osy X (směr osy X bude opačný)
N40 L10	; obrobení druhé kontury vlevo nahoře
N50 AMIRROR Y0	; zrcadlové převrácení osy Y (směr osy Y bude opačný)
N60 L10	; obrobení třetí kontury vlevo dole
N70 MIRROR Y0	; příkaz MIRROR vynuluje předcházející frame Zrcadlové převrácení osy Y (směr osy Y bude opačný)
N80 L10	; obrobení čtvrté kontury vpravo dole
N90 MIRROR	; zrušení zrcadlového převrácení
N100 G0 X300 Y100 M30	; odjíždění nástroje, konec programu

Příklad 2: Soustružení



Vlastní obrábění je uloženo jako podprogram, opracování v příslušných vřetenech je realizováno prostřednictvím zrcadlového převrácení a posunutí.

Programový kód	**Komentář**
N10 TRANS X0 Z140	; posunutí nuly do bodu W
...	; opracování 1. strany pomocí vřetena 1
N30 TRANS X0 Z600	; posunutí nuly ke vřetenu 2
N40 AMIRROR Z0	; zrcadlové převrácení osy Z
N50 ATRANS Z120	; posunutí nuly do bodu W1
...	; opracování 2. strany pomocí vřetena 2

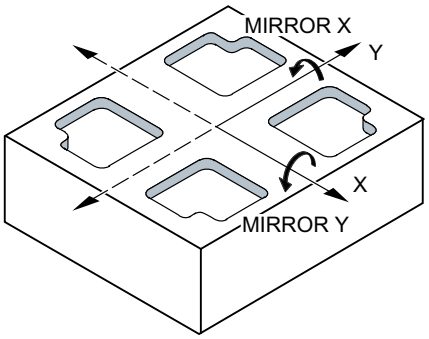
Další informace

MIRROR X... Y... Z...

Zrcadlové převrácení je programováno pomocí přehození směrů os ve zvolené pracovní rovině.

Příklad: Pracovní rovina G17 X/Y

Zrcadlové převrácení (podle osy Y) vyžaduje přehození směrů osy X a proto bude naprogramováno příkazem `MIRROR X0`. Kontura potom bude vyrobena zrcadlově převrácená na protilehlé straně osy zrcadlového převrácení Y.

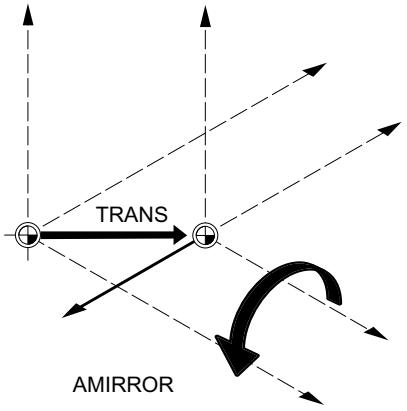


Zrcadlové převrácení je vztaženo na právě platný souřadný systém nastavený pomocí příkazů G54 ... G57, G505 ... G599.

UPOZORNĚNÍ
Žádný počáteční frame Příkaz `MIRROR` zruší veškeré komponenty dříve definovaného programovatelného framu.

AMIRROR X... Y... Z...

Zrcadlové převrácení, které má být superponováno k už existující transformaci, se naprogramuje pomocí příkazu `AMIRROR`. Jako vztažný bod se používá v daném okamžiku nastavený nebo naposled naprogramovaný souřadný systém.



Deaktivování zrcadlového převrácení

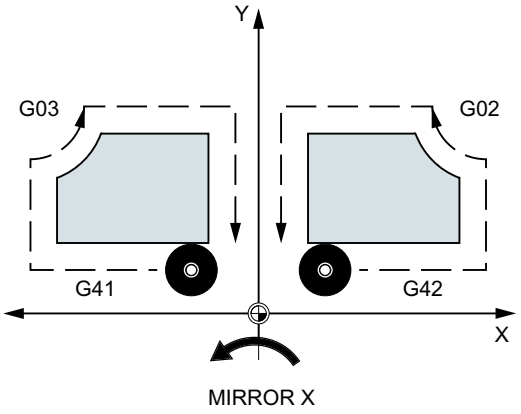
Pro všechny osy: `MIRROR` (bez udání osy)

Všechny komponenty předtím naprogramovaného framu jsou tím vynulovány.

Korekce rádiusu nástroje

Poznámka

Řídící systém spolu s příkazem zrcadlového převrácení automaticky přepíná příkaz korekce posuvu po dráze (`G41`/`G42` příp. `G42`/`G41`), v závislosti na změněném směru obrábění.



Totéž platí také pro směry opisování kruhu (G2/G3, příp. G3/G2).

Poznámka

Jestliže je po příkazu `MIRROR` naprogramováno aditivní otočení `AROT`, možná budete muset pracovat s převráceným směrem otáčení (kladný/záporný, příp. záporný/kladný). Zrcadlová převrácení v geometrických osách jsou řídícím systémem automaticky převáděna do rotací, a pokud je to žádoucí, jsou pro zrcadlová převrácení použity zrcadlové osy specifikované ve strojních parametrech. To se vztahuje také na nastavitelná posunutí počátku.

Zrcadlově převrácená osa

Pomocí strojního parametru je možné nastavit, podle které osy se má zrcadlové převracení provádět:

MD10610 $MN_MIRROR_REF_AX = <hodnota>

Hodnota	Význam
0	Zrcadlové převrácení se bude provádět podle naprogramované osy (negování hodnoty).
1	Osa X je vztažnou osou.
2	Osa Y je vztažnou osou.
3	Osa Z je vztažnou osou.

Interpretace naprogramovaných hodnot

Prostřednictvím strojního parametru může být definováno, jak mají být naprogramované hodnoty interpretovány:

MD10612 $MN_MIRROR_TOGGLE = <hodnota>

Hodnota	Význam
0	Naprogramovaná hodnota osy se nebude vyhodnocovat.
1	Naprogramovaná hodnota osy se bude vyhodnocovat: • Když je naprogramovaná hodnoty osy jiná než 0 (nula), bude se zrcadlově převracet osa, která ještě není zrcadlově převrácena. • Pokud je naprogramovaná hodnota = 0, bude zrcadlové převrácení zrušeno.

3.12.9 Generování framu v závislosti na orientaci nástroje (TOFRAME, TOROT, PAROT)

Příkaz `TOFRAME` vytváří pravoúhlý souřadný systém, jehož osa Z se kryje s právě nastaveným směrem nástroje. Díky tomu má uživatel možnost nástrojem vyjíždět ve směru osy Z, aby nedošlo ke kolizi (např. po zlomení nástroje během zpracování programu, který pracuje s 5 osami).

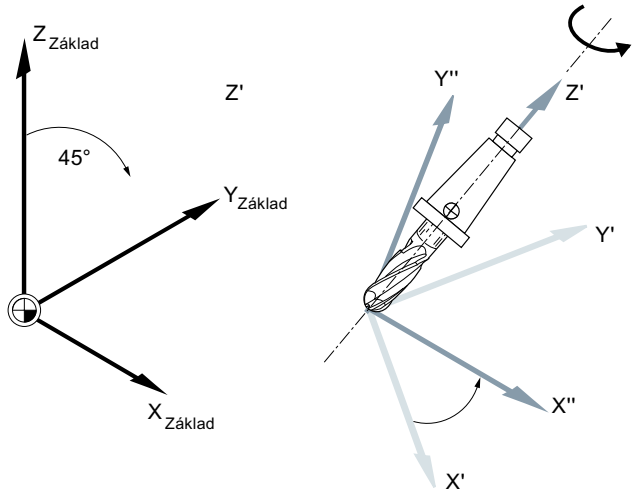
Poloha obou os X a Y je přitom závislá na nastavení strojního parametru MD21110 $MC_X_AXES_IN_OLD_X_Z_PLANE (souřadný systém při automatické definici framu). Nový souřadný systém je buď ponechán ve stavu, který vyplývá z kinematiky stroje, nebo se navíc ještě uskutečňuje otočení okolo nové osy Z, aby se nová osa X kryla se starou osou X (viz informace od výrobce stroje).

Výsledný frame, který tuto orientaci popisuje, se zapisuje do systémových proměnných pro programovatelný frame ($P_PFRAME).

Pomocí příkazu `TOROT` se v programovatelném framu přepisuje pouze rotační složka. Všechny zbývající komponenty zůstávají nezměněny.

Příkazy `TOFRAME` a `TOROT` jsou ušity na míru pro obrábění frézováním, při kterém je za obvyklých okolností aktivní příkaz `G17` (pracovní rovina X/Y). Při soustružení nebo všeobecně při aktivních rovinách `G18` nebo `G19` jsou oproti tomu zapotřebí framy, u nichž se směr nástroje kryje s osou X nebo s osou Y. Pro programování těchto framů se používají příkazy `TOFRAMEX`/`TOROTX` nebo `TOFRAMEY`/`TOROTY`.

Pomocí příkazu `PAROT` se provádí srovnání polohy obrobku a souřadného systému obrobku (WCS).



Syntaxe

```
TOFRAME/TOFRAMEZ/TOFRAMEY/TOFRAMEX
...
TOROTOF

TOROT/TOROTZ/TOROTY/TOROTX
...
TOROTOF

PAROT
...
PAROTOF
```

Význam

TOFRAME:	Nastavení osy Z systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje
TOFRAMEZ:	Stejné jako příkaz TOFRAME
TOFRAMEY:	Nastavení osy Y systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje
TOFRAMEX:	Nastavení osy X systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje
TOROT:	Nastavení osy Z systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje Otočení definované příkazem TOROT je stejné jako při použití příkazu TOFRAME.
TOROTZ:	Stejné jako příkaz TOROT

TOROTY:	Nastavení osy Y systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje
TOROTX:	Nastavení osy X systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje
TOROTOF:	Deaktivování nasměrování rovnoběžně s orientací nástroje
PAROT:	Nasměrování polohy WCS podle obrobku prostřednictvím otočení framu Posunutí, změny měřítka a zrcadlová převrácení v aktivním framu zůstanou zachována.
PAROTOF:	Otočení framu vztažené na obrobek a aktivované příkazem PAROT se příkazem PAROTOF zruší.

Poznámka

Pomocí příkaz TOROT se dosahuje konzistentního programování s aktivními orientovatelnými držáky nástrojů pro každý typ kinematiky.

Analogicky k situaci s otočnými držáky nástrojů může být příkaz PAROT použit pro aktivování otáčení pracovního stolu. Tím je definován frame, který mění polohu souřadného systému obrobku takovým způsobem, že na stroji se neprovádí žádný kompenzační pohyb. Pokud žádný orientovatelný držák nástroje není aktivní, není příkaz PAROT odmítnut.

Příklad:

Programový kód	Komentář
N100 G0 G53 X100 Z100 D0	
N120 TOFRAME	
N140 G91 Z20	; příkaz TOFRANE se započítá, všechny naprogramované pohyby geometrických os se vztahují na nový souřadný systém.
N160 X50	
...	

Další informace

Přiřazení směrů os

Jestliže je na místě příkazu TOFRAME / TOFRAMEZ nebo TOROT / TOROTZ naprogramován příkaz TOFRAMEX, TOFRAMEY, TOROTX, TOROTY, potom platí přiřazení směrů os podle následující tabulky:

Příkaz	směr nástroje (aplikáta)	Vedlejší osa (abscisa)	Vedlejší osa (ordináta)
TOFRAME / TOFRAMEZ / TOROT / TOROTZ	Z	X	Y
TOFRAMEY / TOROTY	Y	Z	X
TOFRAMEX / TOROTX	X	Y	Z

Samostatný systémový frame pro TOFRAME nebo TOROT

Framy vytvořené prostřednictvím příkazů TOFRAME nebo TOROT mohou být zapsány do samostatného systémového framu $P_TOOLFRAME. Za tím účelem musí být nastaven bit 3 ve strojním parametru MD28082 $MC_MM_SYSTEM_FRAME_MASK. Naprogramovaný frame zůstane přitom zachován v nezměněném stavu. Rozdíly se vyskytnou tehdy, pokud je naprogramovaný frame dále zpracováván.

3.12.10 Deaktivování framu (G53, G153, SUPA, G500)

Při zpracovávání určitých operací, jako je např. najíždění na bod pro výměnu nástroje, musí být definovány odlišné složky framů a je nutné je po určitou dobu potlačit.

Nastavené framy mohou být buď modálně zrušeny nebo mohou být blokově potlačeny.

Programovatelné framy mohou být blokově potlačeny nebo mohou být vymazány.

Syntaxe

```
G53
G153
SUPA
G500
TRANS
ROT
SCALE
MIRROR
```

Význam

`G53`:	Potlačení všech programovatelných a nastavitelných framů s blokovou platností
`G153`:	Příkaz `G153` se chová jako příkaz `G53` a potlačuje také celkový základní frame ($P_ACTBFRAME)
`SUPA`:	Příkaz `SUPA` se chová stejně jako příkaz `G153` a potlačuje kromě toho ještě i následující: • Posunutí ručním kolečkem (DRF) • Superponované pohyby • Externí posunutí počátku • Posunutí PRESET
`G500`:	Zrušení všech nastavitelných framů (G54 ... G57, G505 ... G599) s modální platností, pokud se v příkazu G500 nevyskytuje žádná hodnota.
`TRANS ROT SCALE MIRROR`:	bez udání osy způsobí vymazání programovatelného framu.

3.12.11 Programování: Deaktivování superponovaných pohybů pro specifickou osu (CORROF)

Pomocí procedury CORROF jsou deaktivovány následující superponované pohyby pro specifickou osu:

- Aditivní posunutí počátku nastavená pomocí ručního kolečka (DRF-posunutí)
- Offsety pozice naprogramované pomocí systémové proměnné $AA_OFF

Vymazáním hodnot superponovaných pohybů se spustí se zastavení předběžného zpracování a poziční složka deaktivovaného superponovaného pohybu se přenese do pozice v základním souřadném systému. Žádná osa se přitom nebude pohybovat.

Hodnota pozice v souřadném systému **stroje**, kterou je možno načíst pomocí systémové proměnné $AA_IM (aktuální požadovaná hodnota osy v **MCS**), **se nezmění**.

Hodnota pozice v souřadném systému **obrobku**, kterou je možno načíst pomocí systémové proměnné $AA_IW (aktuální požadovaná hodnota osy ve **WCS**), **se změní**, protože nyní obsahuje deaktivovanou složku superponovaného pohybu.

Poznámka

V programu **NC systému** smí být příkaz `CORROF` naprogramován.

V **synchronní akci** příkaz `CORROF` **nesmí** být naprogramován.

Syntaxe

```
CORROF(<osa>,"<řetězec>"[,<osa>,"<řetězec>"])
```

Význam

`CORROF`:	Procedura pro deaktivování následujících posunutí, příp. superponovaných pohybů, osy: • Posunutí DRF • Offset pozice ($AA_OFF)	
	Platnost:	modální
`<osa>`:	Identifikátor osy (identifikátor kanálové, geometrické nebo strojní osy)	
	Datový typ:	AXIS
`<řetězec>`:	Řetězec znaků pro definici druhu superponovaného pohybu	
	Datový typ:	BOOL
	Hodnota	**Význam**
	`DRF`	Posunutí DRF
	`AA_OFF`	Offset pozice ($AA_OFF)

Příklady

Příklad 1: Deaktivování posunutí DRF pro specifickou osu (1)

Posuvem ručním kolečkem DRF se vytváří DRF-posunutí v ose X. Pro ostatní osy tohoto kanálu nejsou v platnosti žádná DRF-posunutí.

Programový kód	Komentář
N10 CORROF(X,"DRF")	; CORROF se zde chová stejně jako příkaz DRFOF
...	

Příklad 2: Deaktivování posunutí DRF pro specifickou osu (2)

Posuvem ručním kolečkem DRF se vytváří DRF-posunutí v ose X a v ose Y. Pro ostatní osy tohoto kanálu nejsou v platnosti žádná DRF-posunutí.

Programový kód	Komentář
; Deaktivuje se pouze posunutí DRF v ose X, posunutí DRF v ose Y zůstává zachováno.	
; Příkazem DRFOF by byla deaktivována obě posunutí.	
N10 CORROF(X,"DRF")	
...	

Příklad 3: Deaktivování offsetu polohy $AA_OFF pro určitou osu

Programový kód	Komentář
; pro osu X se provádí interpolace offsetu pozice == 10	
N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5	
...	
; offset pozice osy X je deaktivován příkazem: $AA_OFF[X]=0	
; Osa X se přitom nepohybuje.	
; Aktuální poloha osy X se započítá do offsetu pozice.	
N80 CORROF(X,"AA_OFF")	
...	

Příklad 4: Deaktivování posunutí DRF a offsetu pozice $AA_OFF pro určitou osu (1)

Posuvem ručním kolečkem DRF se vytváří DRF-posunutí v ose X. Pro ostatní osy tohoto kanálu nejsou v platnosti žádná DRF-posunutí.

Programový kód	Komentář
; pro osu X se provádí interpolace offsetu pozice 10	
N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5	
...	
; Deaktivuje se pouze posunutí DRF a offset polohy v ose X.	
; Posunutí DRF v ose Y zůstává zachováno.	
N70 CORROF(X,"DRF",X,"AA_OFF")	
...	

Příklad 5: Deaktivování posunutí DRF a offsetu pozice $AA_OFF pro určitou osu (2)

Posuvem ručním kolečkem DRF se vytváří posunutí DRF v ose X a v ose Y. Pro ostatní osy tohoto kanálu nejsou v platnosti žádná DRF-posunutí.

Programový kód	Komentář
; pro osu X se provádí interpolace offsetu pozice == 10	
N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5	
...	
; Deaktivuje se posunutí DRF v ose Y a offset polohy v ose X.	
; Posunutí DRF v ose X zůstává zachováno.	
N70 CORROF(Y,"DRF",X,"AA_OFF")	
...	

Další informace

$AA_OFF_VAL

Po deaktivování offsetu polohy prostřednictvím $AA_OFF je systémová proměnná $AA_OFF_VAL (integrovaná dráha superponovaného pohybu osy) pro odpovídající osu rovna nule.

$AA_OFF v provozním režimu JOG

Při změně parametru $AA_OFF se také v provozním režimu JOG uskutečňuje interpolace offsetu polohy ve formě superponovaného pohybu, pokud je však prostřednictvím strojního parametru MD36750 $MA_AA_OFF_MODE tato funkce odblokována.

$AA_OFF v synchronní akci

Jestliže je v okamžiku deaktivování offsetu polohy pomocí příkazu `CORROF(<osa>,"AA_OFF")` aktivní nějaká synchronní akce, která proměnnou $AA_OFF okamžitě znovu nastavuje (`DO $AA_OFF[<osa>]=<hodnota>`), potom se proměnná $AA_OFF deaktivuje a znovu se neaktivuje a aktivuje se alarm 21660. Pokud se však synchronní akce znovu aktivuje později, např. v bloku za příkazem `CORROF`, potom se parametr $AA_OFF nastavuje a offset pozice je interpolován.

Automatická výměna kanálu

Jestliže je osa, pro kterou byl naprogramován příkaz `CORROF`, aktivní v nějakém jiném kanálu, potom je při výměně osy přenesena do daného kanálu (předpoklad: MD30552 $MA_AUTO_GET_TYPE > 0) a potom se offset polohy a/nebo posunutí DRF deaktivují.

3.12.12 Deaktivování aditivních posunutí počátku (DRFOF)

Aditivní posunutí počátku nastavená pomocí ručního kolečka (DRF-posunutí) se deaktivují prostřednictvím procedury `DRFOF`.

Tímto deaktivováním se spustí zastavení předběžného zpracování a poziční složka deaktivovaného posunutí DRF se přenese do pozice v základním souřadném systému a žádné osy se přitom nepohybují. Hodnota v systémové proměnné $AA_IM (aktuální požadovaná hodnota osy v MCS) se nemění, hodnota v systémové proměnné $AA_IW (aktuální požadovaná

hodnota osy ve WCS) se změní, protože nyní obsahuje deaktivovanou složku ze superponovaného pohybu.

Syntaxe

```
DRFOF
```

Význam

DRFOF:	Procedura pro deaktivování všech posunutí DRF pro všechny aktivní osy kanálu	
	Platnost:	modální

Příklady

Příklad 1: Deaktivování posunutí DRF pro specifickou osu (1)

Posuvem ručním kolečkem DRF se vytváří DRF-posunutí v ose X. Pro ostatní osy tohoto kanálu nejsou v platnosti žádná DRF-posunutí.

Programový kód	Komentář
N10 CORROF(X,"DRF")	; CORROF se zde chová stejně jako příkaz DRFOF
...	

Příklad 2: Deaktivování posunutí DRF pro specifickou osu (2)

Posuvem ručním kolečkem DRF se vytváří DRF-posunutí v ose X a v ose Y. Pro ostatní osy tohoto kanálu nejsou v platnosti žádná DRF-posunutí.

Programový kód	Komentář
N10 CORROF(X,"DRF")	; deaktivuje se pouze posunutí DRF v ose X, posunutí DRF v ose Y zůstává zachováno (pomocí příkazu DRFOF by byla deaktivována obě posunutí)
...	

Příklad 3: Deaktivování posunutí DRF a offsetu pozice $AA_OFF pro určitou osu (1)

Posuvem ručním kolečkem DRF se vytváří DRF-posunutí v ose X. Pro ostatní osy tohoto kanálu nejsou v platnosti žádná DRF-posunutí.

Programový kód	Komentář
N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5	; pro osu X se provádí interpolace offsetu pozice == 10
...	
N70 CORROF(X,"DRF",X,"AA_OFF")	; deaktivuje se pouze posunutí DRF a offset polohy v ose X, posunutí DRF v ose Y zůstává zachováno
...	

Příklad 4: Deaktivování posunutí DRF a offsetu pozice $AA_OFF pro určitou osu (2)

Posuvem ručním kolečkem DRF se vytváří posunutí DRF v ose X a v ose Y. Pro ostatní osy tohoto kanálu nejsou v platnosti žádná DRF-posunutí.

Programový kód	**Komentář**
`N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5`	`; pro osu X se provádí interpolace offsetu pozice == 10`
`...`	
`N70 CORROF(Y,"DRF",X,"AA_OFF")`	`; deaktivuje se posunutí DRF v ose Y a offset polohy v ose X, posunutí DRF v ose X zůstává zachováno`
`...`	

3.12.13 Posunutí počátku specifická pro technologii broušení (GFRAME0, GFRAME1 ... GFRAME100)

Příkaz pro aktivování framu pro technologii broušení v kanálu

Naprogramováním příkazu `GFRAME<n>` se v kanálu aktivuje odpovídající frame pro technologii broušení ze správy dat $P_GFR[<n>]. Kromě toho se aktivnímu framu technologie broušení $P_GFRAME současně dosadí frame broušení ze správy dat $P_GFR[<n>]:

`GFRAME<n>` ⇒ $P_GFRAME = $P_GFR[<n>]

Příkaz	**Frame technologie broušení aktivovaný v kanálu**
`GFRAME0`	$P_GFR[0] (**Nulový frame**)
`GFRAME1`	$P_GFR[1]
...	...
`GFRAME100`	$P_GFR[100]

Syntaxe

`GFRAME<n>`

Význam

`GFRAME<n>`:	Aktivování framu pro broušení <n> ve správě dat	
	G-skupina:	64
	Základní nastavení:	MD20150 $MC_GCODE_RESET_VALUES[**63**]
	Platnost:	Modální
<n>:	Číslo framu technologie broušení	
	Rozsah hodnot:	0, 1, 2, ... 100

3.13 Výstupy pomocných funkcí

Funkce

Pomocí pomocných funkcí se PLC v pravý okamžik sděluje, kdy vyšetřovací program potřebuje, aby PLC uskutečnilo na obráběcím stroji specifické spínací operace. Pomocné funkce jsou spolu se svými parametry přenášeny na rozhraní PLC. Předávané hodnoty a signály musí být zpracovávány uživatelským programem PLC.

Pomocné funkce

Do PLC se mohou přenášet následující pomocné funkce:

Pomocná funkce	Adresa
Volba nástroje	T
Korekční parametry nástroje	D, DL
Posuv	F / FA
Otáčky vřetena	S
M-funkce	M
H-funkce	H

Pro každou skupinu funkcí nebo pro jednotlivé funkce je pomocí strojních parametrů definováno, zda se uskutečňuje **před**, **v průběhu** nebo **po** pohybu pracovním posuvem.

PLC je možné naprogramovat, aby přenášené pomocné funkce různým způsobem potvrzovalo.

Vlastnosti

Důležité vlastnosti pomocných funkcí jsou shrnuty v následující přehledové tabulce:

Funkce	Rozšíření adresy		Hodnota			Vysvětlení	Maximální počet na blok
	Význam	Rozsah	Rozsah	Typ	Význam		
M	-	0 (implicitní)	0 ... 99	INT	Funkce	Pro rozsah hodnot mezi 0 a 99 je rozšíření adresy 0. Příkazy, které musí být bez rozšíření adresy: M0, M1, M2, M17, M30	5
	Č. vřetena	1 - 12	1 ... 99	INT	Funkce	M3, M4, M5, M19, M70 s rozšířením adresy o číslo vřetena (např. `M2=5` ; příkaz zastavení pro vřeteno 2). Je-li funkce bez čísla vřetena, platí pro řídící vřeteno.	
	libovolný	0 - 99	100 ... 2147483647	INT	Funkce	Uživatelská M-funkce*	
S	Č. vřetena	1 - 12	0 ... ± 1,8*10^{308}	REAL	Otáčky	Je-li funkce bez čísla vřetena, platí pro řídící vřeteno.	3

Funkce	Rozšíření adresy		Hodnota			Vysvětlení	Maximální počet na blok
	Význam	Rozsah	Rozsah	Typ	Význam		
H	libovolný	0 - 99	0 ... ± 2147483647 ± 1,8*10^{308}	 INT REAL	libovolný	V NC systému nemají tyto funkce žádný efekt, jsou realizovány výlučně prostřednictvím PLC.*	3
T	Č. vřetena (při aktivní správě nástrojů - SN)	1 - 12	0-32000 (také názvy nástrojů, když je aktivní správa nástrojů)	INT	Volba nástroje	Názvy nástrojů se nepředávají na rozhraní PLC.	1
D	-	-	0 - 12	INT	volba korekce nástroje	D0: Deaktivování Předdefinovaná hodnota: D1	1
DL	Lokálně závislá korekce	1 - 6	0 ... ± 1,8*10^{308}	REAL	Aktivování jemné korekce nástroje	Vztahuje se na dříve zvolené D-číslo.	1
F	-	-	0.001 - 999 999,999	REAL	Posuv po dráze		6
FA	Č. osy	1 - 31	0.001 - 999 999,999	REAL	Posuv osy		
* Význam funkcí je definován výrobcem stroje (**viz informace od výrobce stroje!**).							

Další informace

Počet volání funkce na jeden NC blok

V jednom NC bloku smí být naprogramováno maximálně 10 volání funkcí. Pomocné funkce mohou být volány také z akční části **synchronních akcí**.

Seskupení

Zmiňované funkce mohou být soustředěny do skupin. Pro některé M-příkazy je rozdělení do skupin už provedeno. Pomocí seskupení je možné definovat způsob potvrzování.

Rychlé výstupy funkcí (QU)

Funkce, které nebyly konfigurovány jako rychlý výstup, mohou být definovány pro jednotlivá volání jako rychlý výstup pomocí klíčového slova `QU`. Zpracování programu bude pokračovat, aniž by se čekalo na potvrzení provedení této doplňkové funkce (program čeká na potvrzení transportu). Tímto lze zabránit zbytečnému zastavování a přerušování posuvů.

Poznámka

Pro funkci "Rychlé výstupy funkcí" musí být nastaveny odpovídající strojní parametry (--> **výrobce stroje!**).

Výstup funkcí při posuvech

Předávání informací, jakož i čekání na odpovídající reakci stojí čas a v důsledku toho dochází k ovlivňování posuvů.

Rychlé potvrzení bez zpoždění přechodu na další blok

Chování při přechodu na další blok může být ovlivňováno pomocí strojních parametrů. S nastavením „bez zpoždění přechodu na další blok" vyplývá pro rychlé pomocné funkce následující chování:

Výstup pomocné funkce	Chování
před pohybem	Přechod na další blok mezi bloky s rychlými pomocnými funkcemi se uskutečňuje **bez** přerušení a **bez** snížení rychlosti. Výstup pomocných funkcí se uskutečňuje v prvním taktu interpolátoru bloku. Následující blok se provádí bez zpoždění pro potvrzení.
během pohybu	Přechod na další blok mezi bloky s rychlými pomocnými funkcemi se uskutečňuje **bez** přerušení a **bez** snížení rychlosti. Výstup pomocných funkcí se uskutečňuje v průběhu zpracování bloku. Následující blok se provádí bez zpoždění pro potvrzení.
po pohybu	Pohyb se na konci bloku zastaví. Výstup pomocných funkcí se provádí na konci bloku. Následující blok se provádí bez zpoždění pro potvrzení.

⚠ POZOR

Výstup funkcí v režimu řízení pohybu po dráze

Výstup funkcí **před** interpolačním pohybem přeruší režim řízení pohybu po dráze (G64 / G641) a pro předcházející blok vygeneruje přesné najetí.

Výstup funkcí **po** interpolačním pohybu přeruší režim řízení pohybu po dráze (G64 / G641) a pro aktuální blok vygeneruje přesné najetí.

Důležité: Čekání na chybějící potvrzovací signál z PLC může rovněž vést k přerušení režimu řízení pohybu po dráze, např. při posloupnostech M-příkazů v blocích s mimořádně krátkou délkou dráhy.

3.13.1 M-funkce

Pomocí M-funkcí je možné ovládat např. spínací funkce, jako např. "Chladicí kapalina ZAP/VYP" a jiné funkce stroje.

Syntaxe

```
M<hodnota>
M[<rozšíření adresy>]=<hodnota>
```

Význam

`M`:	Adresa pro programování M-funkcí
`<rozšíření adresy>`:	Pro některé M-funkce platí rozšířený způsob zápisu adresy (např. s udáním čísla vřetena u funkcí pro ovládání vřeten).

<hodnota>:	Prostřednictvím zadání hodnoty (číslo M-funkce) se uskutečňuje přiřazení k určité funkci stroje.	
	Typ:	INT
	Rozsah hodnot:	0 ... 2147483647 (max. INT hodnota)

Předem definované M-funkce

Některé M-funkce důležité pro zpracování programu jsou již předem připraveny v rámci standardního vybavení řídícího systému:

M-funkce	Význam
M0*	Programovatelné zastavení
M1*	Volitelné zastavení
M2*	Konec hlavního programu (jako v případě M30)
M3	vřeteno se otáčí doprava
M4	vřeteno se otáčí doleva
M5	zastavení vřetena
M6	Výměna nástroje (standardní nastavení)
M17*	Konec podprogramu
M19	Polohování vřetena
M30*	Konec hlavního programu (jako v případě M2)
M40	Automatické přepínání stupňů převodovky
M41	1. stupeň převodovky
M42	2. stupeň převodovky
M43	3. stupeň převodovky
M44	4. stupeň převodovky
M45	5. stupeň převodovky
M70	přepnutí vřetena do osového režimu

Poznámka

Pro funkce označené * je rozšířený způsob zápisu adresy nepřípustný.

Funkce M0, M1, M2, M17 a M30 se vždy spouštějí až **po** interpolačním posuvu.

M-funkce definované výrobcem stroje

Všechna volná čísla M-funkcí mohou být obsazena výrobcem stroje, např. pro spínací funkce pro řízení upínacích zařízení nebo pro zapínání a vypínání dalších funkcí stroje.

Poznámka

Funkce, které jsou přiřazeny volným číslům M-funkcí, závisí na daném typu stroje. Určité M-funkce proto mohou na různých strojích ovládat zcela odlišné funkce.

M-funkce, které jsou na určitém stroji k dispozici, a funkce, které ovládají, je potřeba nastudovat v informacích od výrobce stroje.

Příklady

Příklad 1: Maximální počet M-funkcí v jednom bloku

Programový kód	Komentář
N10 S...	
N20 X... M3	; M-funkce v bloku s pohybem os,
	; vřeteno se rozbíhá před pohybem osy X.
N180 M789 M1767 M100 M102 M376	; Maximálně 5 M-funkcí v bloku.

Příklad 2: M-funkce jako rychlý výstup

Programový kód	Komentář
N10 H=QU(735)	; rychlý výstup pro H735.
N10 G1 F300 X10 Y20 G64	
N20 X8 Y90 M=QU(7)	; Rychlý výstup pro M7.

M7 bylo naprogramováno jako rychlý výstup, takže režim řízení pohybu po dráze (G64) nebude přerušen.

Poznámka

Tuto funkci používejte pouze pro jednotlivé případy, protože např. v důsledku interakce s jinými funkcemi by mohlo dojít k narušení časové synchronizace.

Další informace vztahující se k předem definovaným M-funkcím

Programovatelné zastavení: M0

V NC bloku s příkazem `M0` bude zpracování pozastaveno. Nyní můžete např. odstranit třísky, provést měření atd..

Programovatelné zastavení 1 – volitelné zastavení: M1

M1 je možné nastavit pomocí:

- HMI/dialogové okno „Ovlivňování programu“
  nebo
- Rozhraní NC/PLC

Zpracovávání programu NC systémem se bude zastavovat na jednotlivých naprogramovaných blocích.

Programové zastavení 2 – Pomocné funkce spojené s M1 se zastavením zpracování programu

Programovatelné zastavení 2 může být nastavováno pomocí HMI/dialogového okna „Ovlivňování programu“ a umožňuje kdykoli přerušit technologickou operaci na konci obráběné části. Díky tomu obsluha může zasahovat do probíhající výroby, např. kvůli odstraňování třísek.

Konec programu: M2, M17, M30

Program je příkazem `M2`, `M17` nebo `M30` ukončen. Pokud je hlavní program vyvoláván z jiného programu (jako podprogram), chovají se příkazy `M2` / `M30` stejně jako příkaz `M17` a naopak, tzn. `M17` se v hlavním programu chová jako příkazy `M2` / `M30`.

Funkce pro ovládání vřetena: M3, M4, M5, M19, M70

Pro všechny funkce pro ovládání vřetena platí rozšířený způsob zápisu adresy s udáním čísla vřetena.

Příklad:

```
Programový kód    Komentář
M2=3              ; Druhé vřeteno se bude otáčet vpravo
```

Pokud není rozšíření adresy naprogramováno, funkce platí pro řídící vřeteno.

3.14 Doplňkové příkazy

3.14.1 Výstup hlášení (MSG)

Pomocí příkazu `MSG()` je možno z výrobního programu odeslat libovolný řetězec znaků jako hlášení pro obsluhujícího pracovníka.

Syntaxe

```
MSG("<text hlášení>"[,<provedení>])
...
MSG( )
```

Význam

`MSG`:	Volání předem definovaného programu pro výstup hlášení	
`<text hlášení>`:	Libovolný řetězec znaků, který se bude vypisovat jako hlášení	
	Typ:	STRING
	Maximální délka:	124 znaků; výpis se uskutečňuje na dvou řádcích (2*62 znaků)
	V rámci textových hlášení se mohou prostřednictvím operátoru zřetězení "<<" vypisovat také proměnné.	

<Provedení>:	Parametr pro stanovení časového okamžiku, kdy se má zápis hlášení uskutečnit (volitelné).		
	Typ:	INT	
	Hodnota:	0 (základní nastavení)	Pro zápis hlášení se nevytváří žádný vlastní blok v hlavní větvi programu. Uskuteční se v následujícím NC-bloku, který lze zpracovat. Nedochází k žádnému přerušení aktivního režimu řízení pohybu po dráze.
		1	Pro zápis hlášení se vytváří vlastní blok v hlavní větvi programu. Aktivní režim řízení pohybu po dráze se přeruší.
MSG():	Naprogramováním příkazu MSG() bez textu hlášení se momentálně vypisované hlášení opět vymaže. Pokud není hlášení vymazáno, vypisuje se tak dlouho, dokud se nevyskytne další hlášení.		

Poznámka

Jestliže se má hlášení vypisovat v aktivním jazyce uživatelského rozhraní, potřebuje uživatel informace o tom, který jazyk je momentálně na HMI nastaven. Tento údaj je možné ve výrobním programu a v synchronních akcích zjistit pomocí systémové proměnné $AN_LANGUAGE_ON_HMI (Strana 1320).

Příklady

Příklad 1: Výstup / vymazání hlášení

```
Programový kód                                  Komentář
N10 G91 G64 F100                                ; režim řízení pohybu po dráze
N20 X1 Y1
N... X... Y...
N20 MSG ("Obrábění, díl 1")                     ; hlášení se vypíše až v bloku N30
                                                ; režim řízení pohybu po dráze zůstává zachován
N30 X... Y...
N... X... Y...
N400 X1 Y1
N410 MSG ("Obrábění, díl 2",1)                  ; hlášení se vypíše v bloku N410
                                                ; režim řízení pohybu po dráze se přeruší
N420 X1 Y1
N... X... Y...
N900 MSG ()                                     ; vymazání hlášení
```

Příklad 2: Textové hlášení s proměnnou

```
Programový kód                                  Komentář
N10 R12=$AA_IW [X]                              ; aktuální pozice osy X v R12
```

```
Programový kód                                          Komentář
N20 MSG ("Polohu osy X"<<R12<<"zkontrolujte")  ; hlášení s proměnnou R12 se vypíše
...
N90 MSG ()                                      ;vymazání hlášení z bloku N20
```

3.14.2 Zápis řetězce znaků do proměnné BTSS (WRTPR)

Pomocí funkce `WRTPR()` je možno z výrobního programu zapsat libovolný řetězec znaků do proměnné BTSS programu progProtText.

Syntaxe

```
WRTPR(<řetězec>[,<doba zpracování>])
```

Význam

WRTPR:	Volání funkce pro výstup řetězce znaků.		
<řetězec>:	Libovolná posloupnost znaků, která se zapíše do proměnné BTSS programu progProtText.		
	Typ:	STRING	
	Maximální délka:	128 znaků	
<doba zpracování>:	Volitelný parametr pro stanovení časového okamžiku, kdy se má zápis řetězce znaků uskutečnit.		
	Typ:	INT	
	Rozsah hodnot:	0, 1	
		0 (předdefinované)	Pro zápis řetězce znaků se nevytváří žádný vlastní blok v hlavní větvi programu. Uskuteční se v následujícím NC-bloku, který lze zpracovat. Nedochází k žádnému přerušení aktivního režimu řízení pohybu po dráze.
		1	Pro zápis řetězce znaků se vytváří vlastní blok v hlavní větvi programu. Aktivní režim řízení pohybu po dráze se přeruší.

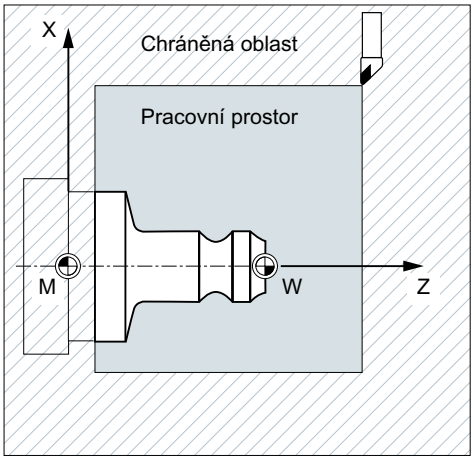
Příklady

```
Programový kód              Komentář
N10 G91 G64 F100            ; režim řízení pohybu po dráze
N20 X1 Y1
N30 WRTPR("N30")            ; řetězec znaků "N30" se zapíše až v bloku N40
                            ; režim řízení pohybu po dráze zůstává zachován
N40 X1 Y1
N50 WRTPR("N50",1)          ; řetězec znaků "N50" se zapíše až v bloku N50
                            ; režim řízení pohybu po dráze se přeruší
N60 X1 Y1
```

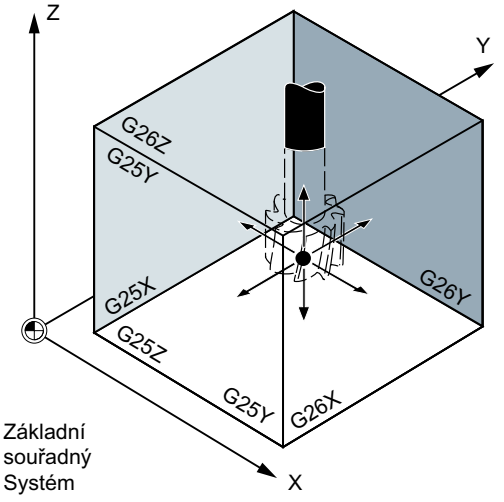
3.14.3 Ohraničení pracovního pole

3.14.3.1 Ohraničení pracovního pole v BCS (G25/G26, WALIMON, WALIMOF)

Pomocí funkcí `G25`/`G26` je možné ve všech kanálových osách ohraničit pracovní oblast (pracovní pole, pracovní prostor), ve které se má nástroj pohybovat. Oblasti mimo hranice pracovního pole definované příkazy `G25`/`G26` jsou pro pohyby nástroje zablokovány.



Údaje souřadnic pro jednotlivé osy platí v základním souřadném systému:



Ohraničení pracovního pole pro všechny osy, pro které byla tato funkce aktivována, musí být naprogramováno příkazem `WALIMON`. Pomocí příkazu `WALIMOF` se ohraničení pracovního prostoru zruší. Příkaz `WALIMON` je standardní systémové nastavení a tento příkaz musí být naprogramován jen tehdy, pokud bylo předtím ohraničení pracovního pole zrušeno.

Syntaxe

```
G25 X…Y…Z…
G26 X…Y…Z…
WALIMON
...
WALIMOF
```

Význam

G25:	**Spodní** hranice pracovního pole Přiřazení hodnot kanálovým osám v základním souřadném systému
G26:	**Horní** hranice pracovního pole Přiřazení hodnot kanálovým osám v základním souřadném systému
X… Y… Z…:	Spodní, příp. horní hranice pracovního pole pro jednotlivé kanálové osy Údaje jsou vztaženy na základní souřadný systém.
WALIMON:	**Aktivování** ohraničení pracovního pole pro všechny osy
WALIMOF:	**Deaktivování** ohraničení pracovního pole pro všechny osy

Vedle programovatelného zadání hodnoty pomocí příkazů G25/G26 je možné také zadání pomocí osových nastavovaných parametrů.

SD43420 $SA_WORKAREA_LIMIT_PLUS (ohraničení pracovního pole plus)

SD43430 $SA_WORKAREA_LIMIT_MINUS (ohraničení pracovního pole mínus)

Aktivování a deaktivování ohraničení pracovního pole s parametry nastavenými pomocí SD43420 a SD43430 se uskutečňuje specificky pro určitý směr pomocí osových nastavovaných parametrů s okamžitou platností:

SD43400 $SA_WORKAREA_PLUS_ENABLE (ohraničení pracovního pole v kladném směru aktivní)

SD43410 $SA_WORKAREA_MINUS_ENABLE (ohraničení pracovního pole v záporném směru aktivní)

Aktivováním/deaktivováním pro určitý směr je možné zajistit, aby byla pracovní oblast pro danou osu ohraničena jen v jednom směru.

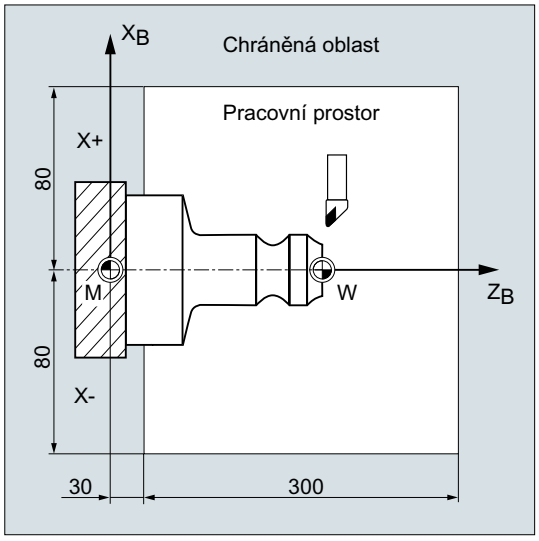
Poznámka

Ohraničení pracovního pole naprogramované pomocí příkazů G25/G26 má přednost a přepisuje hodnoty uložené v parametrech SD43420 a SD43430.

Poznámka

V příkazech G25/G26 můžete také pomocí adresy S naprogramovat mezní hodnotu pro otáčky vřetena. Pokud budete potřebovat další související informace, viz kapitola " Programovatelné omezení otáček vřetena (G25, G26) (Strana 105) ".

Příklad:



Prostřednictvím ohraničení pracovního pole pomocí příkazů G25/26 se pracovní prostor soustruhu omezí tak, aby byla níže položená zařízení, jako jsou revolverový zásobník, měřicí stanice atd. chráněna před poškozením.

Základní nastavení: WALIMON

Programový kód	**Komentář**
N10 G0 G90 F0.5 T1	
N20 G25 X-80 Z30	; definice dolního ohraničení pro jednotlivé souřadné osy
N30 G26 X80 Z330	; definice horního ohraničení
N40 L22	; program pro oddělování třísky
N50 G0 G90 Z102 T2	; do bodu pro výměnu nástroje
N60 X0	
N70 WALIMOF	; deaktivování ohraničení pracovního pole
N80 G1 Z-2 F0.5	; vrtání
N90 G0 Z200	; zpět
N100 WALIMON	; aktivování ohraničení pracovního pole
N110 X70 M30	; konec programu

Další informace

Vztažný bod na nástroji

Když je aktivní délková korekce nástroje, jako vztažný bod je monitorována špička nástroje, jinak se použije vztažný bod držáku nástroje.

Pokud jde o rádius nástroje, jeho zohlednění musí být řešeno samostatně. K tomuto účelu slouží kanálový strojní parametr:

MD21020 $MC_WORKAREA_WITH_TOOL_RADIUS

Jestliže se vztažný bod nástroje nachází mimo oblast definovanou ohraničením pracovního pole nebo pokud tuto oblast opustí, zpracování programu se zastaví.

Poznámka

Jestliže jsou aktivní transformace, může se vyhodnocování a započítávání parametrů nástroje (délka a rádius nástroje) od zde popisovaného chování lišit.

Programovatelné ohraničení pracovního pole, G25/G26

Pro každou osu může být definováno ohraničení pracovního pole shora (G26) a zdola (G25). Tyto hodnoty jsou v platnosti okamžitě a zůstávají v případě patřičného nastavení strojního parametru (--> MD10710 $MN_PROG_SD_RESET_SAVE_TAB) zachovány i po resetu a opětovném zapnutí.

Poznámka

Pomocí předem definované funkce CALCPOSI (Strana 587) je možno před zahájením pohybů pracovním posuvem zkontrolovat, zda bude možné předpokládanou dráhu uskutečnit, aniž by došlo k narušení ohraničení pracovního pole a/nebo chráněných oblastí.

3.14.3.2 Ohraničení pracovního pole ve WCS/ENS (WALCS0 ... WALCS10)

Funkce „Ohraničení pracovního pole ve WCS/ENS" umožňuje flexibilní ohraničení oblasti, ve které se může určitý konkrétní nástroj pohybovat, omezením rozsahu pohybu kanálových os v souřadném systému obrobku (WCS) nebo v nastavitelném souřadném systému počátku (ENS). Tato funkce je určena především pro použití u běžných soustruhů.

Předpoklady

Je nutno, aby bylo kanálovými osami najeto na referenční bod.

Skupina ohraničení pracovního pole

Aby se při změně uspořádání os, např. při aktivování/deaktivování transformací nebo aktivního framu, nemusely znovu zapisovat specifická ohraničení pracovního pole pro všechny kanálové osy, jsou k dispozici skupiny ohraničení pracovního pole.

Ve skupině ohraničení pracovního pole jsou zahrnuta následující data:

- Ohraničení pracovního pole pro všechny kanálové osy
- Vztažný systém pro ohraničení pracovního pole

Syntaxe

```
...
$P_WORKAREA_CS_COORD_SYSTEM[<WALimNo>]=<hodnota>
$P_WORKAREA_CS_PLUS_ENABLE[<WALimNo>,<osa>]=<hodnota>
$P_WORKAREA_CS_LIMIT_PLUS[<WALimNo>,<osa>]=<hodnota>
```

```
$P_WORKAREA_CS_MINUS_ENABLE[<WALimNo>,<osa>]=<hodnota>
$P_WORKAREA_CS_LIMIT_MINUS[<WALimNo>,<osa>]=<hodnota>
...
WALCS<n>
...
WALCS0
```

Význam

$P_WORKAREA_CS_COORD_SYSTEM[<WALimNo>]=<hodnota>		
Souřadný systém, na který se ohraničení pracovního pole z dané skupiny vztahuje		
<WALimNo>:	Skupina ohraničení pracovního pole	
	Typ:	INT
	Rozsah hodnot:	0 (skupina 1) ... 9 (skupina 10)
<hodnota>:	Hodnota typu INT.	
	1	Souřadný systém obrobku (WCS)
	3	Nastavitelný souřadný systém (ENS)

$$P_WORKAREA_CS_PLUS_ENABLE[<WALimNo>,<osa>]=<hodnota>		
Aktivování ohraničení pracovního pole v **kladném** směru uvedené kanálové osy.		
<WALimNo>:	Skupina ohraničení pracovního pole	
	Typ:	INT
	Rozsah hodnot:	0 (skupina 1) ... 9 (skupina 10)
<osa>:	Název kanálové osy	
<hodnota>:	Hodnota typu BOOL	
	0 (FALSE)	deaktivováno (blokováno)
	1 (TRUE)	Uvolnění

$P_WORKAREA_CS_MINUS_ENABLE[<WALimNo>,<osa>]=<hodnota>		
Aktivování ohraničení pracovního pole v **záporném** směru uvedené kanálové osy.		
<WALimNo>:	Skupina ohraničení pracovního pole	
	Typ:	INT
	Rozsah hodnot:	0 (skupina 1) ... 9 (skupina 10)
<osa>:	Název kanálové osy	
<hodnota>:	Hodnota typu BOOL.	
	0 (FALSE)	deaktivováno (blokováno)
	1 (TRUE)	aktivováno

$P_WORKAREA_CS_LIMIT_PLUS[<WALimNo>,<osa>]=<hodnota>
Ohraničení pracovního pole v **kladném** směru uvedené kanálové osy

<WALimNo>:	Skupina ohraničení pracovního pole	
	Typ:	INT
	Rozsah hodnot:	0 (skupina 1) ... 9 (skupina 10)
<osa>:	Název kanálové osy	
<hodnota>:	Hodnota typu REAL	

$P_WORKAREA_CS_LIMIT_MINUS[<WALimNo>,<osa>]=<hodnota>		
Ohraničení pracovního pole v **záporném** směru uvedené kanálové osy		
<WALimNo>:	Skupina ohraničení pracovního pole	
	Typ:	INT
	Rozsah hodnot:	0 (skupina 1) ... 9 (skupina 10)
<osa>:	Název kanálové osy	
<hodnota>:	Hodnota typu REAL.	

WALCS<n>:	Aktivování ohraničení pracovního pole ze skupiny ohraničení pracovního pole	
<n>:	Číslo skupiny ohraničení pracovního pole	
	Rozsah hodnot:	1 ... 10

WALCS0:	Deaktivování aktivních ohraničení pracovního pole

Poznámka

Počet skupin ohraničení pracovního pole, který je skutečně k dispozici, závisí na nastavení konfiguračních parametrů (--> viz informace od výrobce stroje).

Příklad:

V kanálu jsou definovány 3 osy: X, Y a Z

Má být definována skupina ohraničení pracovního pole č. 2 a ta má být potom aktivována tak, aby jednotlivé osy ve WCS byly ohraničeny podle následujících údajů:

- Osa X ve směru plus: 10 mm
- Osa X ve směru mínus: žádné ohraničení
- Osa Y ve směru plus: 34 mm
- Osa Y ve směru mínus: -25 mm
- Osa Z ve směru plus: žádné ohraničení
- Osa Z ve směru mínus: -600 mm

Programový kód	Komentář
...	
N51 $P_WORKAREA_CS_COORD_SYSTEM[1]=1	; ohraničení pracovního pole ve skupině ohraničení pracovního pole č. 2 platí ve WCS
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	; aktivování skupiny ohraničení pracovního pole č. 2
...	

Další informace

Platnost

Ohraničení pracovního pole aktivované příkazy `WALCS1` - `WALCS10` se uplatňují nezávisle na ohraničení pracovního pole, které bylo aktivováno příkazem `WALIMON`. Jestliže jsou aktivní obě funkce, bude se uplatňovat to ohraničení, na které se při pohybu osy narazí jako první.

Vztažný bod na nástroji

Vyhodnocování a započítávání dat nástroje (délka nástroje a rádius nástroje) a tím pádem také vztažný bod na nástroji při monitorování ohraničení pracovního pole odpovídá chování při ohraničení pracovního pole pomocí příkazu `WALIMON`.

3.14.4 Najíždění na referenční bod (G74)

Po zapnutí stroje je nutné (jestliže je použit inkrementální systém pro měření dráhy), aby saně všech os najely na své referenční značky. Teprve poté je možné programovat pohyby posuvu.

Pomocí příkazu `G74` je možné najíždění na referenční bod uskutečnit v rámci NC programu.

Syntaxe

`G74 X1=0 Y1=0 Z1=0 A1=0 ...` ; programování vyžaduje samostatný blok

Význam

`G74`:	Příkaz G-funkce pro najíždění na referenční bod
`X1=0 Y1=0 Z1=0 …`:	Uvedené adresy os stroje `X1`, `Y1`, `Z1`... pro **lineární osy**, které budou najíždět na referenční bod
`A1=0 B1=0 C1=0 …`:	Uvedené adresy os stroje `A1`, `B1`, `C1`... pro **kruhové osy**, které budou najíždět na referenční bod

Poznámka

Pro osu, která má pomocí příkazu `G74` najíždět na referenční bod, by neměla být před tímto najížděním naprogramována žádná transformace.

Transformaci deaktivujte příkazem `TRAFOOF`.

Příklad:

Při změně měřicího systému se najíždí na referenční bod a provádí se seřízení polohy počátku souřadné soustavy obrobku.

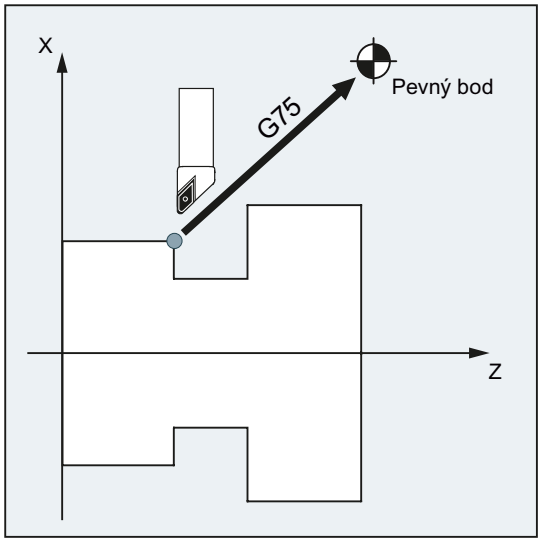
Programový kód	**Komentář**
`N10 SPOS=0`	`; vřeteno v režimu regulace polohy`
`N20 G74 X1=0 Y1=0 Z1=0 C1=0`	`; najíždění na referenční bod lineárními osami a kruhovými osami`
`N30 G54`	`; posunutí počátku`
`N40 L47`	`; program pro oddělování třísky`
`N50 M30`	`; konec programu`

3.14.5 Najíždění na pevný bod (G75)

Prostřednictvím příkazu `G75` s blokovou platností je možno osami jednotlivě a nezávisle na sobě navzájem najíždět na pevné body v pracovním prostoru stroje, např. na bod pro výměnu nástroje, bod pro upnutí, bod pro výměnu palety atd.

Pevné body jsou polohy v souřadném systému stroje, které jsou uloženy ve strojních parametrech (MD30600 $MA_FIX_POINT_POS[n]). Pro každou osu mohou být definovány maximálně 4 pevné body.

Na pevné body je možno najíždět ze kteréhokoli NC programu nezávisle na aktuální pozici nástroje nebo obrobku. Před pohybem os se uskutečňuje interní zastavení předběžného zpracování.



Předpoklady

Pro najíždění na pevné body pomocí příkazu `G75` musí být splněny následující předpoklady:

- Souřadnice pevného bodu musí být přesně zjištěny a uloženy ve strojních parametrech.
- Pevné body musí ležet v rámci platného rozsahu pohybu (--> pozor na hranice dané softwarovými koncovými spínači!).
- Osami, které se mají pohybovat, musí být najeto na referenční bod.
- Nesmí být aktivní žádné korekce rádiusu nástroje.
- Nesmí být aktivní žádné kinematické transformace.
- Osy, které se mají pohybovat, se nesmí podílet na žádné aktivní transformaci.
- Žádná z os, které se mají pohybovat, nesmí být vlečnou osou v aktivní vazbě.
- Žádná z os, které se mají pohybovat, nesmí být osou v seskupení os gantry.
- Cykly překladače nesmí způsobovat deaktivování žádné složky pohybu.

Syntaxe

```
G75 <název osy><poloha osy> ... FP=<n>
```

Význam

`G75`:	Najíždění na pevný bod
`<název osy>`:	Název osy stroje, která má najet na pevný bod. Jsou přípustné všechny identifikátory os.
`<poloha osy>`:	Hodnota polohy nemá žádný význam. Proto se zpravidla zadává hodnota "0".

FP=:	Pevný bod, na který se má najíždět.		
	<n>:	Číslo pevného bodu	
		Rozsah hodnot:	1, 2, 3, 4
	Upozornění: Jestliže FP=<n> není naprogramováno nebo pokud není uvedeno číslo pevného bodu nebo pokud je zadáno FP=0, bude příkaz interpretován, jako by bylo zadáno FP=1 a bude se tedy najíždět na pevný bod č. 1.		

Poznámka

V bloku s příkazem G75 může být naprogramován i větší počet os. Osy potom budou na uvedený pevný bod najíždět současně.

Poznámka

Hodnota v adrese FP nesmí být větší, než je počet definovaných pevných bodů pro kteroukoli z naprogramovaných os (MD30610 $MA_NUM_FIX_POINT_POS).

Příklad:

Za účelem výměny nástroje se má osami X (= AX1) a Z (= AX3) najet na pevnou pozici os stroje č. 1 se souřadnicemi X = 151,6 a Z = -17,3.

Strojní parametry:

- MD30600 $MA_FIX_POINT_POS[AX1,0] = 151.6
- MD30600 $MA_FIX_POINT[AX3,0] = 17.3

NC program:

```
Programový kód              Komentář
…
N100 G55                    ; Aktivování nastavitelného posunutí počátku.
N110 X10 Y30 Z40            ; Najíždění na pozici ve WCS.
N120 G75 X0 Z0 FP=1 M0      ; Osa X najíždí na 151,6
                            ; a osa Z najíždí na pozici 17,3 (v MCS).
                            ; Každá osa se pohybuje svou maximální rychlostí.
                            ; V tomto bloku nesmí být aktivní žádné jiné pohyby.
                            ; Aby po dosažení koncové polohy
                            ; nebyly prováděny žádné další doplňkové pohyby,
                            ; je zde vloženo zastavení.
N130 X10 Y30 Z40            ; Znovu se bude najíždět na pozici z bloku N110.
                            ; Opět se aktivuje posunutí počátku.
…
```

Poznámka

Jestliže je aktivní funkce "Správa nástrojů se zásobníkem", pomocná funkce `T`... příp. `M...` (zpravidla funkce `M6`) pro uvolnění blokování přechodu na další blok na konci pohybu s příkazem `G75` nepostačuje.

Důvod: Jestliže je v platnosti nastavení "Správa nástrojů se zásobníkem je aktivní", nejsou pomocné funkce pro výměnu nástroje předávány do PLC.

Další informace

G75

Osy se pohybují jako osy stroje rychlým posuvem. Při vytváření příslušného pohybu se interně uplatňují funkce "SUPA" (potlačení všech framů) a "G0 RTLIOF" (pohyb rychlým posuvem s interpolací jednotlivých os).

Jestliže podmínky pro funkci "RTLIOF" (interpolace jednotlivých os) nejsou splněny, bude se na pevný bod najíždět v dráhovém režimu.

Při dosažení pevného bodu se osy zastavují v rámci tolerančního okna "Jemné přesné najetí".

Dynamika nastavitelná pomocí parametrů pro příkazy G75

Pro pohyby, jejichž cílem je najíždění na pevný bod (G75), je možné prostřednictvím následujícího strojního parametru nastavit požadovaný režim dynamického chování:

MD18960 $MN_POS_DYN_MODE (druh dynamicky polohovací osy)

Doplňkové pohyby os

V okamžiku interpretace bloku s příkazem G75 jsou zohledňovány následující doplňkové axiální pohyby os:

- Externí posunutí počátku
- DRF
- Synchronizační offset ($AA_OFF)

Doplňkové pohyby os se potom nesmí změnit, a to až do okamžiku, kdy je dosaženo konce pohybu, který byl vyvolá příkazem G75.

Doplňkové pohyby po interpretaci bloku s příkazem G75 mají za následek odpovídající posunutí pevného bodu, na který se najíždí.

Následující doplňkové pohyby nejsou zohledňovány nezávisle na okamžik interpretace a mají za následek odpovídající posunutí cílové pozice:

- On-line korekce nástroje
- Doplňkové pohyby z cyklů překladače ve BCS, viz MCS.

Aktivní frame

Všechny aktivní framy jsou ignorovány. Najíždění se uskutečňuje v souřadném systému stroje.

Ohraničení pracovního pole ve WCS/ENS

Ohraničení pracovního pole pro specifický souřadný systém (WALCS0 ... WALCS10) se v bloku s příkazem G75 neuplatňují. Cílový bod je monitorován jako počáteční bod následujícího bloku.

Pohyby os/vřeten s příkazy POSA/SPOSA

Jestliže se naprogramované osy/vřetena pohybovaly pomocí příkazů POSA, příp. SPOSA, budou tyto pohyby dovedeny do konce ještě před najížděním na pevný bod.

Funkce vřetena v bloku s příkazem G75

Jestliže je vřeteno z funkce "Najíždění na pevný bod" vyjmuto, potom mohou být v bloku s příkazy G75 naprogramovány ještě i funkce vřetena (např. nastavování polohy pomocí příkazů SPOS/SPOSA).

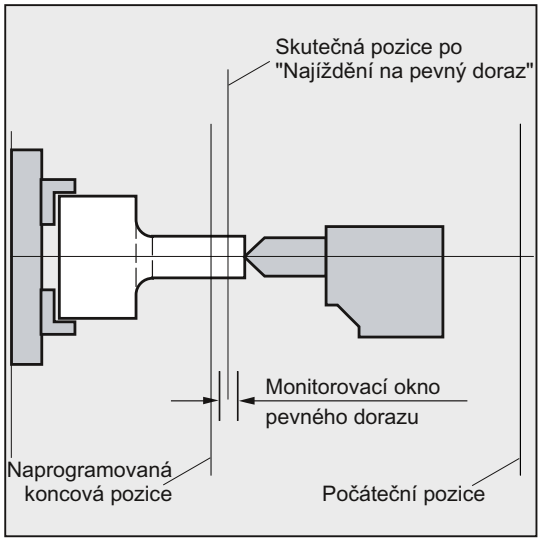
Osy typu modulo

U os typu modulo se na pevný bod najíždí po nejkratší dráze.

3.14.6 Najíždění na pevný doraz (FXS, FXST, FXSW)

Funkce

Pomocí funkce „Najíždění na pevný doraz" je možné vytvářet definované síly pro upnutí obrobku, jaké jsou např. zapotřebí pro koníky soustruhu, hrotové objímky a držáky. Kromě toho můžete pomocí této funkce najíždět na mechanické referenční body.



V případě dostatečně sníženého kroutícího momentu lze uskutečňovat také jednoduché měřicí operace, aniž by bylo nutné mít připojenu měřicí sondu. Funkce „Najíždění na pevný doraz" se může používat pro osy a vřetena, která jsou schopna se pohybovat jako osy.

Syntaxe

```
FXS[<osa>]=...
```

```
FXST[<osa>]=...
FXSW[<osa>]=...
FXS[<osa>]=... FXST[<osa>]=...
FXS[<osa>]=... FXST[<osa>]=... FXSW[<osa>]=...
```

Význam

`FXS`:	Příkaz pro aktivování a deaktivování funkce "Najíždění na pevný doraz"
	`FXS[<osa>]=1`: Aktivování funkce
	`FXS=[<osa>]=0`: Deaktivování funkce
`FXST`:	Volitelný příkaz pro nastavení upínacího momentu
	Zadává se v % maximálního momentu pohonu.
`FXSW`:	Volitelný příkaz pro nastavení šířky okna pro monitorování pevného dorazu
	Údaj v mm, palcích nebo ve stupních
`<osa>`:	Názvy os stroje
	Programují se osy stroje (X1, Y1, Z1 atd.)

Poznámka

Příkazy `FXS`, `FXST` a `FXSW` mají modální platnost.

Naprogramování příkazů `FXST` a `FXSW` je nepovinné: Pokud není specifikován žádný parametr, platí vždy naposled naprogramovaná hodnota, příp. hodnota nastavená v odpovídajícím strojním parametru.

Aktivování najíždění na pevný doraz: FXS[<osa>] = 1

Pohyby k cílovému bodu mohou být popsány jako pohyby po dráze nebo polohovací pohyby. V případě polohovacích os může být funkce prováděna i přes hranice bloku.

Najíždění na pevný doraz může být prováděno i více osami najednou a souběžně s pohybem jiných os. Pevný doraz musí ležet mezi počáteční a cílovou pozicí.

UPOZORNĚNÍ

Nebezpečí kolize

Jestliže byla pro nějakou osu/vřeteno aktivována funkce „Najíždění na pevný doraz“, nesmí být pro tuto osu naprogramována žádná nová pozice.

Vřetena je nutné před aktivováním této funkce přepnout do režimu polohové regulace.

Příklad:

Programový kód	**Komentář**
`X250 Y100 F100 FXS[X1]=1 FXST[X1]=12.3 FXSW[X1]=2`	`; Osa X1 bude najíždět s posuvem F100 (údaj volitelný) na cílovou pozici X=250 mm.` `Blokovací moment činí 12.3% maximálního momentu pohonu, monitorování se uskutečňuje v okně, které má šířku 2 mm.`
`...`	

Deaktivování najíždění na pevný doraz: FXS[<osa>] = 0

Při deaktivování funkce se spouští zastavení předběžného zpracování.

V bloku s příkazem `FXS[<osa>]=0` se smí a mají nacházet příkazy posuvu.

UPOZORNĚNÍ

Nebezpečí kolize

Pohyby posuvu na návratovou pozici musí vést od okna pevného dotazu, jinak by mohlo dojít k poškození pevného dorazu nebo stroje.

Přechod na následující blok se provede po dosažení návratové pozice. Jestliže žádná návratová pozice není zadána, uskuteční se přechod na následující blok přímo po vypnutí omezení momentu.

Příklad:

Programový kód	**Komentář**
`X200 Y400 G01 G94 F2000 FXS[X1] = 0`	`; Osa X1 se bude stahovat z okna pevného dorazu na pozici X=200 mm. Všechny ostatní údaje jsou volitelné.`
`...`	

Blokovací moment (FXST) a monitorovací okno (FXSW)

Naprogramované omezení momentu FXST je funkční od začátku bloku, tzn. také najíždění na pevný doraz se provádí se sníženým momentem. Příkazy FXST a FXSW mohou být naprogramovány, příp. změněny, na libovolném místě výrobního programu. Změny vstupují v platnost před pohyby posuvu, které se nacházejí ve stejném bloku.

UPOZORNĚNÍ
Nebezpečí kolize Jestliže je naprogramováno nové monitorovací okno pevného dorazu, změní se nejen šířka okna, ale i vztažný bod pro střed okna, jestliže se osa předtím pohybovala. Skutečná poloha osy stroje je při změně okna novým středem tohoto okna. Okno musí být zvoleno tak, že pouze odpadnutí od dorazu povede k aktivování monitorování okna.

Další informace

Náběžná hrana charakteristiky

Pomocí strojního parametru může být definována náběžná hrana pro nový mezní moment, aby se zabránilo jeho skokovému nastavení (např. aby se zabránilo otisku hrotové objímky).

Potlačení alarmu

Při použití tohoto příkazu může být v případě nutnosti ve výrobním programu potlačen alarm mezního zastavení. Za tím účelem je třeba ve strojním parametru alarm maskovat a nové nastavení strojního parametru aktivovat pomocí příkazu NEW_CONF.

Aktivování

Příkazy pro najíždění na pevný doraz mohou být vyvolávány ze synchronních akcí / technologických cyklů. Aktivování se může uskutečňovat i bez pohybů, moment bude okamžitě omezen. Pokud se má osa pohybovat přes nastavený bod, monitorování dorazu se aktivuje.

Aktivování ze synchronních akcí

Příklad:

Jestliže se vyskytne očekávaná událost ($R1) a najíždění na pevný doraz ještě neproběhlo, má se aktivovat FXS pro osu Y. Moment má činit 10% momentu jmenovitého. Pro šířku monitorovaného okna platí předdefinovaná hodnota.

Programový kód

```
N10 IDS=1 WHENEVER (($R1=1) AND ($AA_FXS[Y]==0)) DO $R1=0 FXS[Y]=1
FXST[Y]=10
```

Normální výrobní program se musí postarat o to, aby parametr $R1 byl v požadovaném okamžiku nastaven.

Deaktivování ze synchronních akcí

Příklad:

Jestliže se uskuteční očekávaná akce ($R3) a existuje-li stav „Najeto na pevný doraz" (systémová proměnná $AA_FXS), měl by se příkaz FXS deaktivovat.

Programový kód

```
IDS=4 WHENEVER (($R3==1) AND ($AA_FXS[Y]==1)) DO FXS[Y]=0
FA[Y]=1000 POS[Y]=0
```

Pevného dorazu bylo dosaženo

Poté, co je dosaženo pevného dorazu:

- Zbytková dráha je vymazána a je dosazena nová požadovaná hodnota polohy.
- Moment pohonu naroste až na naprogramovanou mezní hodnotu `FXSW` a zůstane konstantní.
- Aktivuje se monitorování pevného dorazu v rámci zadané šířky okna.

Okrajové podmínky

- Měření s vymazáním zbytkové dráhy
  Příkazy "Měření s vymazáním zbytkové dráhy" (příkaz `MEAS`) a "Najíždění na pevný doraz" nemohou být naprogramovány současně v jednom bloku.
  Výjimka: Jedna funkce působí na jednu dráhovou osu a druhá funkce na polohovací osu nebo obě působí na polohovací osy.
- Kontrola kontury
  Jestliže je aktivní funkce "Najíždění na pevný doraz", monitorování kontury se neprovádí.
- Polohovací osy
  Při „Najíždění na pevný doraz" s polohovacími osami se přechod na další blok uskutečňuje nezávisle na vyhodnocení pevného dorazu.
- Spřažené osy a osový zásobník
  Najíždění na pevný doraz je přípustné také pro spřažené osy a osový zásobník:
  Stav přiřazených os stroje zůstává v průběhu otáčení zásobníku nezměněn. To platí také pro modální omezení momentu příkazem FOCON.
- Najíždění na pevný doraz **není možné** naprogramovat za těchto okolností:
  - Jedná se o osu gantry.
  - V případě konkurenčních polohovacích os, které jsou výlučně řízeny z PLC (aktivování `FXS` musí být uskutečněno v NC programu).
- Jestliže hranice momentu poklesla natolik, že osa není schopna dosáhnout zadané požadované hodnoty, regulátor polohy se zablokuje na své mezní hodnotě a odchylka od kontury začne narůstat. V tomto provozním stavu může mít zvýšení mezního momentu za následek náhlé trhavé pohyby. Aby bylo zajištěno, že osa je stále ještě schopna sledovat požadovaný bod, je nutno kontrolovat, že odchylka kontury není větší než při neomezeném momentu.

3.14.7 Doba prodlevy (G4)

Prostřednictvím příkazu `G4` se v daném bloku naprogramuje čas (doba prodlevy), která bude plynout, když se bude daný blok zpracovávat v hlavní větvi programu. Přechod na následující blok se uskuteční, až když všechen tento čas uplyne.

Poznámka

Příkaz `G4` způsobuje přerušení režimu řízení pohybu po dráze.

Syntaxe

```
G4 F<čas>
G4 S<NumSpi>
G4 S<n> = <NumSpi>
```

Význam

`G4`:	Aktivování doby prodlevy	
	samostatný blok:	ano
`F<čas>`:	Pomocí adresy `F` se zadává doba prodlevy `<čas>` v sekundách.	
`S<NumSpi>`:	Pomocí adresy S se zadává doba prodlevy v otáčkách aktuálního hlavního vřetena `<NumSpi>`.	
`S<n>=NumSpi>`:	Pomocí adresy S se zadává doba prodlevy v otáčkách vřetena `<NumSpi>`, přičemž se jedná o vřeteno definované pomocí rozšíření adresy `<n>`.	

Poznámka

Adresy `F` a `S` použité pro udání čas v bloku pro definici doby prodlevy `G4` nijak neovlivňují hodnoty posuvu `F...` a otáček vřetena `S...` v programu.

Okrajové podmínky

Synchronní akce

V programu jsou naprogramovány dvě synchronní akce tak, že z následujícího bloku s dobou prodlevy se stává blok akce, ve kterém má být synchronní akce zpracována. Tato synchronní akce je modální synchronní akcí. Druhá synchronní akce je blokovou synchronní akcí. Pokud má bloková synchronní akce ovlivňovat modální synchronní akci, např. pokud má být zpracování uvolněno příkazem UNLOCK, musí být jako platná doba prodlevy k dispozici **minimálně dva takty interpolátoru**, např. `G4 F<interpolační takt * 2>`.

Platná doba prodlevy závisí na nastavení strojního parametru MD10280 $MN_PROG_FUNCTION_MASK, Bit 4 = <hodnota>

Hodnota	Význam
0	Platná doba prodlevy se rovná naprogramované době prodlevy
1	Platná doba prodlevy se rovná naprogramované době prodlevy zaokrouhlené na nejbližší vyšší násobek taktu interpolátoru (MD10071 $MN_IPO_CYCLE_TIME)

Příklad programování:

- MD10071 $MN_IPO_CYCLE_TIME == 8 ms
- MD10280 $MN_PROG_FUNCTION_MASK, Bit 4 = 1

```
Programový kód                          Komentář
N10 WHEN TRUE DO LOCK(1)                ; bloková synchronní akce: LOCK
                                        ; modální synchronní akce ID=1
N20 G4 F2                               ; blok akce pro synchronní akci z N10
N30 WHEN TRUE DO UNLOCK(1)              ; bloková synchronní akce: UNLOCK
                                        ; modální synchronní akce ID=1
N40 ID=1 WHENEVER TRUE DO $R0=1 RDISABLE ; modální synchronní akce ID=1
                                        ; R-parametr R0=1
                                        ; aktivování blokování operace načtení
N50 G4 F0.012                           ; blok akce pro synchronní akci z N40 a
                                        N50
                                        ; viz níže, odstavec „Popis“
N60 G4 F10
```

Popis

Požadované chování je následující: Bloková synchronní akce z bloku N30 odstraňuje aktivní blokování (LOCK) modální synchronní akce s nastavením ID=1 z bloku N40 a tím se v bloku N50 zapisuje do R-parametru a aktivuje se blokování operace načtení. Tohoto chování je ale dosaženo jen tehdy, pokud je platná doba prodlevy dlouhá minimálně dva takty interpolátoru.

Platná doba prodlevy vyplývá z naprogramované doby prodlevy, taktu interpolátoru a nastavení parametru MD10280 $MN_PROG_FUNCTION_MASK, bit 4. Aby byla platná doba prodlevy dlouhá minimálně dva takty interpolátoru, musí být naprogramována následující doba prodlevy.

- Bit 4 == 0: Naprogramovaná doba prodlevy ≥ 2 * takt interpolátoru
- Bit 4 == 1: Naprogramovaná doba prodlevy ≥ 1,5 * takt interpolátoru

Pokud je platná doba prodlevy kratší než dva takty interpolátoru, zápis R-parametru a blokování načítání se uskuteční až v bloku N60.

Příklad:

```
Programový kód                  Komentář
N10 G1 F200 Z-5 S300 M3         ; posuv F, otáčky vřetena S
N20 G4 F3                       ; doba prodlevy: 3s
```

Programový kód	Komentář
N30 X40 Y10	
N40 G4 S30	;čekání po dobu 30 otáček vřetena, což při S=300 ot/min a korekci otáček 100% odpovídá času: t = 0,1 min).
N50 X...	; posuv a otáčky vřetena naprogramované v bloku N10 jsou opět v platnosti

3.14.8 Zastavení interního předběžného zpracování

Funkce

Při přístupu ke stavovým údajům stroje ($A...) generuje řídící systém interní zastavení předběžného zpracování. Následující blok bude zpracován teprve tehdy, až jsou zcela zpracovány všechny bloky, které již dříve byly připraveny a uloženy do paměti. Předcházející blok bude zastaven s přesným najetím (jako při G9).

Příklad:

Programový kód	Komentář
...	
N40 POSA[X]=100	
N50 IF $AA_IM[X]==R100 GOTOF MARKE1	; Přístup ke stavovým údajům stroje ($A...), řídící systém generuje interní zastavení předběžného zpracování.
N60 G0 Y100	
N70 WAITP(X)	
N80 MARKE1:	
...	

3.15 Ostatní informace

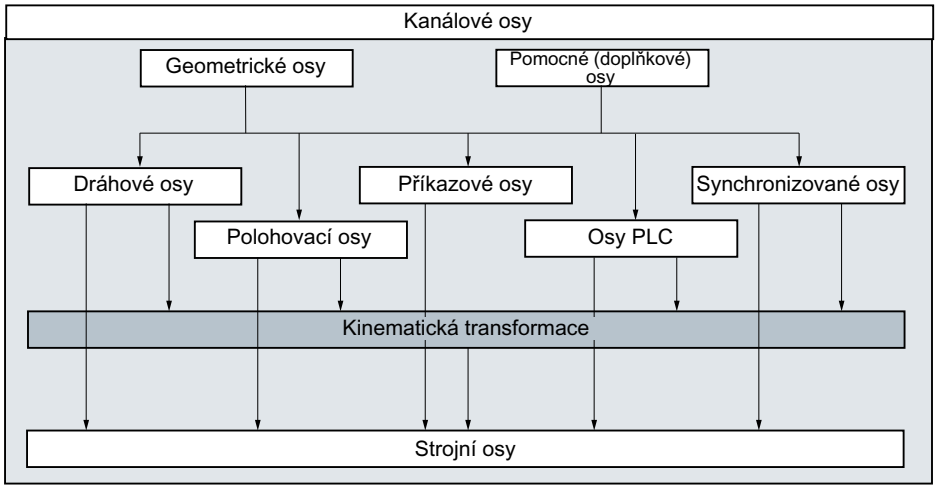
3.15.1 Osy

Typy os

V rámci programování je potřeba rozlišovat následující typy os:

- Hlavní osy / geometrické osy
- Pomocné (doplňkové) osy
- Hlavní vřeteno, řídící vřeteno
- Strojní osy

- Kanálové osy
- Dráhové osy
- Polohovací osy
- Synchronizované osy
- Příkazové osy
- Osy PLC / konkurenční polohovací osy
- Spřažené osy
- Řídící spřažené osy



3.15.1.1 Hlavní osy / geometrické osy

Hlavní osy určují pravotočivý pravoúhlý souřadný systém. V tomto souřadném systému jsou programovány pohyby nástroje.

V NC technologii jsou hlavní osy nazývány osami geometrickými. Tento pojem se používá také v této Příručce programování.

Přepínatelné geometrické osy

Pomocí funkce „Přepínatelné geometrické osy" (viz příručka Popis funkcí, Pro pokročilé) je možné výrobním programem změnit prostřednictvím strojních parametrů nastavené přiřazení svazku geometrických os. Přitom může být libovolná geometrická osa nahrazena kanálovou osou definovanou jako pomocná synchronní osa.

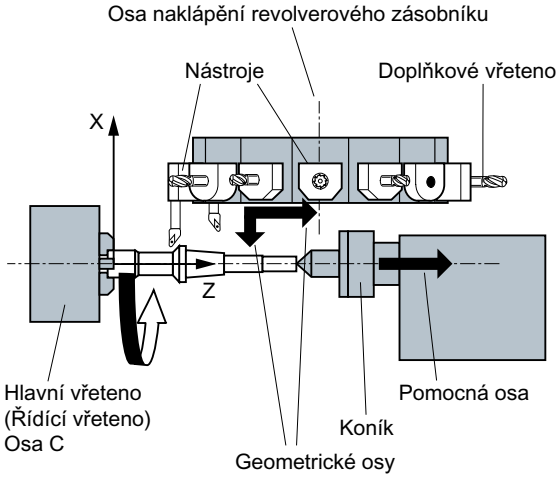
Identifikátor osy

Název/identifikátor geometrické osy může být definován prostřednictvím následujícího strojního parametru:

MD20060 $MC_AXCONF_GEOAX_NAME_TAB (název geometrické osy v kanálu)

Standardní označení u soustruhů:

1. Geometrická osa: X
2. Geometrická osa: Z



Standardní označení u frézek:

1. Geometrická osa: X
2. Geometrická osa: Y
3. Geometrická osa: Z

Další informace

Pro programování framů a geometrie obrobku (kontura) se používají maximálně tři osy.

Identifikátory pro geometrické a kanálové osy smí být stejné, pokud je takové přiřazení možné.

Názvy geometrických a kanálových os mohou být ve všech kanálech stejné. Díky tomu může být program zpracován ve kterémkoli libovolném kanálu.

3.15.1.2 Pomocné (doplňkové) osy

Oproti geometrickým osám není v případě pomocných os definována žádná geometrická souvislost mezi osami.

Typickými pomocnými osami jsou:

- Osy revolverového zásobníku nástrojů
- Osy otočného stolu
- Osy naklápěcí hlavičky
- Osy zakladače

Identifikátor osy

U soustruhu s revolverovým zásobníkem jsou pomocnými osami například:

- Poloha revolverového zásobníku U
- Koník V

Příklad programování

Programový kód	Komentář
N10 G1 X100 Y20 Z30 A40 F300	; pohyby dráhových os
N20 POS[U]=10POS[X]=20 FA[U]=200 FA[X]=350	; pohyby polohovacích os
N30 G1 X500 Y80 POS[U]=150FA[U]=300 F550	; dráhová a polohovací osa
N40 G74 X1=0 Z1=0	; najíždění na referenční bod

3.15.1.3 Hlavní vřeteno, řídící vřeteno

To, které vřeteno je považováno za hlavní, je dáno kinematikou stroje. Tato vřetena jsou zpravidla deklarována jako řídící prostřednictvím strojních parametrů.

Toto přiřazení může být změněno programovým příkazem `SETMS (<číslo vřetena>)`. Pomocí příkazu `SETMS` bez udání čísla vřetena je možné přepnout zpět na řídící vřeteno definované ve strojním parametru.

Pro řídící vřeteno platí speciální funkce, jako např. pro řezání závitů.

Identifikátor vřetena

S nebo S0

3.15.1.4 Osy stroje

Osy stroje jsou osy, které na stroji fyzicky existují.

Naprogramovaný pohyb dráhové nebo doplňkové osy může ovlivňovat prostřednictvím transformace (TRANSMIT, TRACYL nebo TRAORI) aktivní v daném kanálu i více os stroje.

Osy stroje jsou přímo v programu uváděny jen ve zvláštních případech (např. při najíždění na referenční nebo na pevný bod).

Identifikátor osy

Název/identifikátor strojní osy může být definován prostřednictvím následujícího strojního parametru, který je specifický pro daný NC systém:

MD10000 $MN_AXCONF_MACHAX_NAME_TAB (název osy stroje)

Standardní nastavení: X1, Y1, Z1, A1, B1, C1, U1, V1

Strojní osy mají kromě toho pevný identifikátor osy, který je možno kdykoli použít nezávisle na názvu nastaveném ve strojním parametru:

AX1, AX2, ..., AX<n>

3.15.1.5 Kanálové osy

Výrazem kanálové osy jsou označovány všechny geometrické, doplňkové a strojní osy, které jsou přiřazeny danému kanálu.

Identifikátor osy

Specifický kanálový název/identifikátor geometrické a doplňkové osy může být definován prostřednictvím následujícího strojního parametru:

MD20080 $MC_AXCONF_CHANAX_NAME_TAB (název kanálové osy)

Standardní nastavení: X, Y, Z, A, B, C, U, V

Přiřazení, které určuje, která strojní osa odpovídá dané geometrické nebo doplňkové ose v kanálu, je definováno v následujícím strojním parametru:

MD20070 $MC_AXCONF_MACHAX_USED (použité strojní osy)

3.15.1.6 Dráhové osy

Dráhové osy popisují úsek dráhy a tím pádem také pohyb nástroje v prostoru.

Podél této dráhy je v platnosti naprogramovaný posuv. Osy podílející se na této dráze dosáhnou své pozice současně. Zpravidla se jedná o geometrické osy.

To, které osy jsou dráhovými osami, a tedy osami, u nichž je regulovanou veličinou rychlost, je definováno předdefinovanými parametry.

V NC programu mohou být dráhové osy definovány pomocí příkazu `FGROUP`.

Pokud budete potřebovat další informace o příkazu `FGROUP`, viz "Posuv (G93, G94, G95, F, FGROUP, FL, FGREF) (Strana 106)".

3.15.1.7 Polohovací osy

Interpolace polohovacích os se provádí odděleně, tzn. každá polohovací osa má svůj vlastní osový interpolátor a svůj vlastní posuv. Interpolace polohovacích os neprobíhá spolu s dráhovými osami.

Pohyb polohovacích os je ovládán NC programem nebo z PLC. Jestliže má být osa ovládána současně jak NC programem, tak i z PLC, objeví se chybové hlášení.

Typickými polohovacími osami jsou:

- Podavač pro přísun surových obrobků
- Podavač pro odkládání obrobených obrobků
- Zásobník nástrojů/revolver

Typy

Je potřeba rozlišovat mezi polohovacími osami se synchronizací na konci bloku a se synchronizací přes více bloků:

Osy typu POS

Přechod na další blok se na konci předešlého bloku uskutečňuje tehdy, jestliže všechny dráhové a polohovací osy naprogramované v tomto bloku dosáhly svého koncového bodu.

Osy typu POSA

Pohyby těchto polohovacích os mohou probíhat i ve více blocích.

Osy typu POSP

Pohyby těchto polohovacích os pro najíždění na koncovou pozici se provádějí po menších úsecích.

Poznámka

Polohovací osy se stanou synchronizovanými osami, jestliže se s nimi pohybuje bez zvláštního identifikátoru POS/POSA.

Režim řízení pohybu po dráze (G64) pro dráhové osy je možný jen tehdy, jestliže polohovací osy (POS) dosáhly své koncové polohy ještě před dráhovými osami.

Dráhové osy, které byly naprogramovány pomocí příkazů `POS`/`POSA`, jsou po dobu trvání tohoto bloku vyjmuty ze svazku dráhových os.

Pokud budete potřebovat další informace týkající se příkazů`POS`, `POSA` a `POSP`, viz "Najíždění polohovacími osami (POS, POSA, POSP, FA, WAITP, WAITMC) (Strana 114)".

3.15.1.8 Synchronizované osy

Synchronizované osy se pohybují synchronně po dráze z počátečního bodu do naprogramovaného koncového bodu.

Posuv naprogramovaný pomocí příkazu `F` platí pro všechny dráhové osy naprogramované v daném bloku, ne však pro synchronizované osy. Synchronizované osy potřebují pro svou dráhu stejný čas jako dráhové osy.

Synchronizovanou osou může být kupříkladu kruhová osa, která se pohybuje synchronně s dráhovou interpolací.

3.15.1.9 Příkazové osy

Příkazové osy se spouští ze synchronizovaných akcí na základě nějaké události (příkaz). Jejich polohování, spouštění a zastavování může probíhat zcela asynchronně vzhledem k výrobnímu programu. Osa se nemůže pohybovat současně prostřednictvím výrobního programu a synchronní akce.

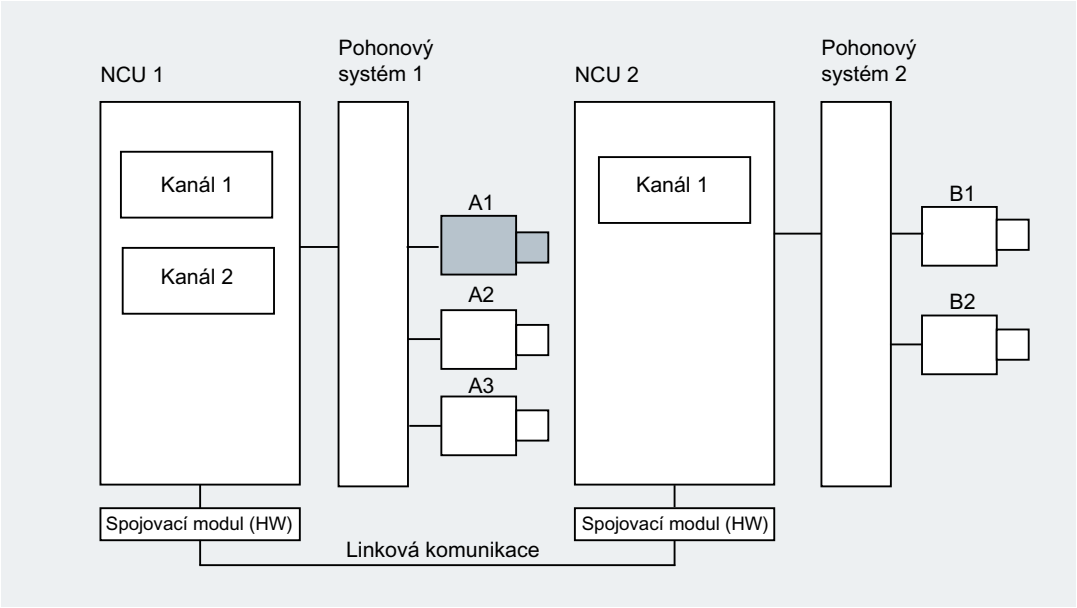
Interpolace příkazových os se provádí odděleně, tzn. každá příkazová osa má svůj vlastní osový interpolátor a svůj vlastní posuv.

3.15.1.10 Osy PLC

PLC osy jsou ovládány z PLC prostřednictvím speciálních funkčních bloků v základním programu a mohou se pohybovat asynchronně vůči všem ostatním osám. Také tyto posuvné pohyby probíhají nezávisle na dráhových a synchronních pohybech.

3.15.1.11 Spřažené osy

Spřažené osy jsou osy, které jsou fyzicky připojeny na jinou NCU a podléhají její polohové regulaci. Spřažené osy mohou být dynamicky přiřazeny kanálům **jiné** NCU. Spřažené osy nejsou z pohledu určité NCU lokálními osami.



Dynamická změna přiřazení určité NCU je založena na koncepci **osového zásobníku**. Výměna pomocí příkazů `GET` a `RELEASE` z výrobního programu **není** pro spřažené osy proveditelná.

Další informace

Předpoklady

- Příslušné řídící jednotky NCU1 a NCU2 musí být spojeny přes své linkové moduly pomocí rychlé linkové komunikace.
- Osa musí být prostřednictvím strojních parametrů náležitě konfigurována.
- Musí být k dispozici volitelný doplněk "Spřažená osa“.

Popis

Regulace polohy se uskutečňuje na NCU, na které je osa fyzicky spojena s pohonem. Zde se nachází také příslušné rozhraní VDI osy. Požadovaná hodnota polohy je však u spřažených os vytvářena na jiné NCU, která je předávána linkovým spojením mezi oběma NCU.

Linková komunikace musí zabezpečovat souhru mezi interpolátory a regulátory polohy nebo rozhraním PLC. Požadované hodnoty vypočítávané interpolátory se musí přenášet do smyčky regulace polohy v domovské NCU a naopak musí být skutečné hodnoty odtud vraceny zpět do interpolátorů.

Osový zásobník

Osový zásobník je kruhová zásobníková datová struktura, ve které se provádí přiřazení lokálních os a/nebo spřažených os kanálům. Záznamy v kruhové paměti umožňují **cyklický posun**.

Kromě přímého odkazu na lokální osy nebo spřažené osy umožňuje konfigurace spřažené osy v logickém zobrazení osy stroje také odkazy na osový zásobník. Tento typ odkazu má tyto části:

- Číslo zásobníku **a**
- Slot (místo - pozice v odpovídajícím kruhovém zásobníku)

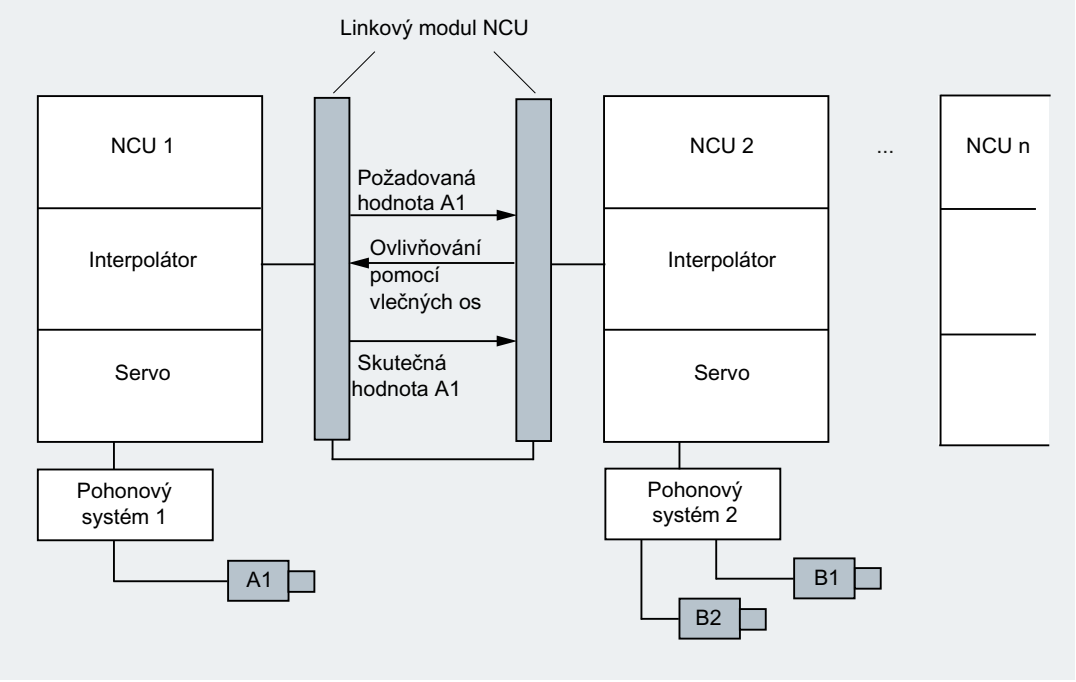
Záznam v místě v kruhovém zásobníku obsahuje:

- lokální osu **nebo**
- spřaženou osu

Záznamy v osovém zásobníku obsahují lokální osy stroje nebo spřažené osy z hlediska příslušné NCU. Záznamy v logickém zobrazení os stroje (MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB) dané NCU jsou pevně dané.

3.15.1.12 Řídící spřažené osy

Řídící spřaženou osou je osa, která je interpolována jednou z NCU a která je používána jednou nebo několika NCU jako řídící osa pro ovládání vlečných (podřízených) os.



Alarm regulátoru polohy dané osy se posílá do všech ostatních NCU, jež jsou připojeny k postižené ose pomocí řídící spřažené osy.

NCU, které jsou závislé na řídící spřažené ose, mohou díky ní využívat následující vazby:

- Řídící hodnota (požadovaná, skutečná, simulovaná řídící hodnota)
- Vlečení
- Tangenciální sledování

- Elektronická převodovka (ELG)
- Synchronní vřeteno

Programování

Nadřazená NCU:

Pouze NCU, která je řídící spřažené ose fyzicky přiřazena, může programovat posuvy pro tuto osu. Programování nesmí obsahovat žádné speciální funkce nebo operace.

NCU podřízených os:

Programování na NCU podřízených os nesmí obsahovat žádné příkazy pohybu pro řídící spřaženou osu (hodnotu nadřazené osy). Pokud se tak stane, aktivuje se alarm.

Řídící spřažená osa je adresována obvyklým způsobem pomocí identifikátoru kanálové osy. Stavy řídící spřažené osy mohou být zjišťovány pomocí zvolených systémových proměnných.

Další informace

Předpoklady

- Příslušné řídící jednotky NCU1 až NCU<n> (<n> maximálně 8) musí být spojeny přes své linkové moduly pomocí rychlé linkové komunikace.
  Literatura:
  Příručka k přístroji, Konfigurace NCU
- Osa musí být prostřednictvím strojních parametrů náležitě konfigurována.
- Musí být k dispozici volitelný doplněk "Spřažená osa".
- Pro všechny jednotky NCU, které se podílejí na zpracování, musí být v konfiguraci nastaven stejný takt interpolátoru.

Omezení

- Nadřazená osa, která je řídící spřaženou osou, nemůže být sama podřízenou spřaženou osou, tzn. nemůže být ovládána jinými NCU, než je její vlastní NCU.
- Nadřazená osa, která je řídící spřaženou osou, nesmí být osou zásobníku, tzn. nemůže být ovládána střídavě různými jednotkami NCU.
- Řídící spřažená osa nemůže být naprogramovanou řídící osou ve skupině gantry.
- Spojení s řídící spřaženou osou nemůže být vedeno přes několik po sobě jdoucích stupňů (kaskádové zapojení).
- Výměna osy může být implementována jen v rámci domovské NCU řídící spřažené osy.

Systémové proměnné

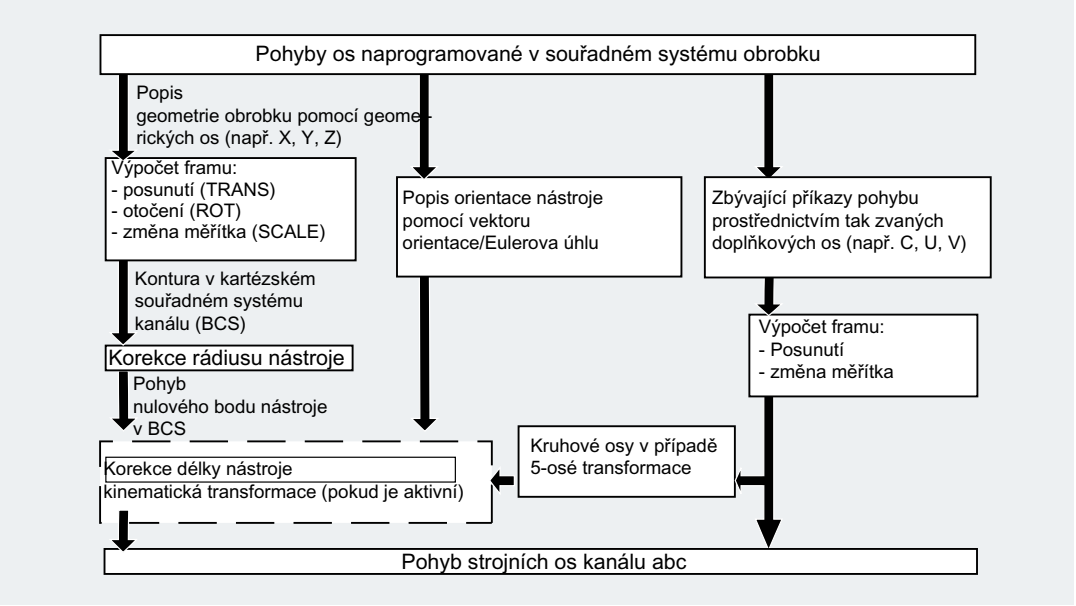
Pomocí identifikátoru kanálové osy mohou být používány následující systémové proměnné:

Systémová proměnná	Význam
$AA_LEAD_SP	Simulovaná řídící hodnota - poloha
$AA_LEAD_SV	Simulovaná řídící hodnota - rychlost

Pokud jsou systémové proměnné aktualizovány pomocí NCU řídící spřažené osy, budou se nové hodnoty přenášet také do jednotek NCU, které si přejí ovládat podřízené osy na základě této nadřazené osy.

3.15.2 Od příkazu pohybu k pohybu stroje

Souvislost mezi naprogramovanými pohyby os (příkazy pohybu) a z nich vyplývající pohyby stroje má ozřejmit následující obrázek:

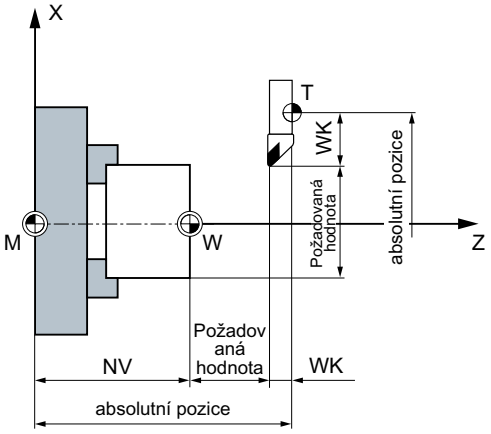


3.15.3 Výpočet dráhy

Výpočet dráhy zjišťuje úseky dráhy, které je potřeba v daném bloku urazit, přičemž se berou v úvahu také všechna posunutí a korekce.

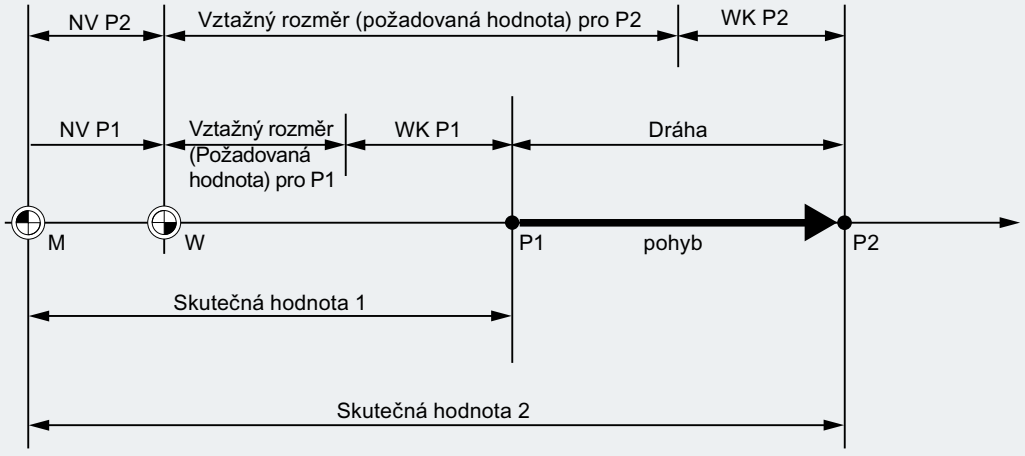
Obecně platí:

Dráha = požadovaná hodnota – skutečná hodnota + posunutí počátku (ZO) + korekce nástroje (TO)



Pokud je v novém programovém bloku naprogramováno nové posunutí počátku a nová korekce nástroje, potom platí:

- při zadávání absolutních rozměrů:
  dráha = (abs. rozměr P2 – abs. rozměr P1) + (ZO P2 – ZO P1) + (TO P2 – TO P1).
- Při zadávání inkrementálních rozměrů:
  dráha = řetězová kóta + (ZO P2 – ZO P1) + (TO P2 – TO P1).



3.15.4 Adresy

Pevné adresy

Tyto adresy jsou pevně definovány, tzn. adresové znaky není možné měnit.

Výpis se nachází v tabulce "Pevné adresy (Strana 1252)".

Nastavitelné adresy

Výrobce stroje může těmto adresám přiřadit prostřednictvím strojních parametrů jiný název.

Poznámka

Nastavitelné adresy musí být v rámci řídícího systému jednoznačné, tzn. jeden adresový identifikátor nesmí být použit pro různé typy adres (hodnoty osy a koncové body, orientace nástroje, interpolační parametry, ...).

Výpis se nachází v tabulce "Nastavitelné adresy (Strana 1257)".

Adresy s modální/blokovou platností

Adresy s modální platností si zachovávají svou naprogramovanou hodnotu (ve všech následujících blocích) tak dlouho, dokud není pod stejnou adresou naprogramována nová hodnota.

Adresy s blokovou platností se uplatňují pouze v bloku, v němž byly naprogramovány.

Příklad:

Programový kód	Komentář
`N10 G01 F500 X10`	
`N20 X10`	`; posuv F z bloku N10 je v platnosti tak dlouho, dokud není zadána nová hodnota`

Adresy s axiálním rozšířením

U adres s axiálním rozšířením se název osy nachází v hranatých závorkách za adresou, která definuje přiřazení k ose.

Příklad:

Programový kód	Komentář
`FA[U]=400`	`; specifický osový posuv pro osu U`

Viz také tabulka "Pevné adresy (Strana 1252)".

Rozšířený způsob zápisu adresy

Rozšířený způsob zápisu adresy nabízí možnost organizovat v systému větší počet os a vřeten.

Rozšířená adresa se skládá z numerického rozšíření a z aritmetického výrazu, který je přiřazen znakem "=". Numerické rozšíření je vždy jedno- nebo dvojmístné a vždy kladné.

Rozšířený způsob zápisu adresy je přípustný pouze pro následující jednoduché adresy:

Adresa	Význam
X, Y, Z, ...	Adresa osy
I, J, K	Interpolační parametr
S	Otáčky vřetena

SPOS, SPOSA	Poloha vřetena
M	Doplňkové funkce
H	Pomocné funkce
T	Číslo nástroje
F	Posuv

Příklady:

Programový kód	**Komentář**
`X7`	`; žádné znaménko "=" není zapotřebí; 7 je hodnota; znak "=" však může být zapsán i zde`
`X4=20`	`; osa X4; znak "=" je nutný`
`CR=7.3`	`; 2 písmena ; znak "=" je zapotřebí`
`S1=470`	`; otáčky pro 1. vřeteno: 470 ot/min`
`M3=5`	`; zastavení pro 3. vřeteno`

U adres M, H, S, ale i u příkazů SPOS a SPOSA, může být numerické rozšíření nahrazeno proměnnou. Identifikátor proměnné se přitom nachází v hranatých závorkách.

Příklady:

Programový kód	**Komentář**
`S[SPINU]=470`	`; otáčky pro vřeteno, jehož číslo je uloženo v proměnné SPINU`
`M[SPINU]=3`	`; vřeteno se otáčí vpravo, jeho číslo je uloženo v proměnné SPINU`
`T[SPINU]=7`	`; předvolba nástroje pro vřeteno, jehož číslo je uloženo v proměnné SPINU`

3.15.5 Název

Příkazy jsou podle normy DIN 66025 prostřednictvím vyššího jazyka NC systému doplňovány mimo jiné objekty se zadaným názvem.

Těmito objekty mohou být např.:

- Systémové proměnné
- Uživatelem definované proměnné
- Osy / vřetena
- Podprogramy
- Klíčová slova
- Návěští skoků
- Makra

Poznámka

Identifikátor musí být jednoznačný. Stejný identifikátor se **nesmí** používat pro různé objekty.

Pravidla pro názvy

Název může být libovolný, musí však být dodržena následující pravidla:

- Přípustné znaky:
  - Písmena: A ... Z, a ... z
  - Číslice: 0 ... 9
  - Znak podtržení: _
- První dva znaky by měly být písmena nebo znak podtržení.
- Maximální délka:
  - Názvy programů (Strana 44): 24 znaků
  - Názvy os: 8 znaků
  - Názvy proměnných: 31 znaků

Poznámka

Rezervovaná klíčová slova nesmí být použita jako identifikátor.

Cykly

Aby se zabránilo konfliktům názvů, doporučujeme vám při sestavování názvů uživatelských cyklů dodržovat následující pravidla:

Řetězec znaků	rezervováno pro názvy
• CYCLE • CUST_ • GROUP_ • _ • S_ • E_ • F_	Cykly firmy Siemens
• CCS_	Cykly překladače firmy SIEMENS
• CC_	Uživatelské cykly překladače

Uživatelské cykly

Pro názvy uživatelských cyklů doporučujeme, aby začínaly řetězcem „U_“.

Proměnné

Informace týkající se sestavování názvů proměnných naleznete v následujících kapitolách:

- Systémová data (Strana 390)
- Definice uživatelských proměnných (DEF) (Strana 397)

3.15.6 Konstanty

Konstanta (všeobecná)

Konstanta je datový objekt, jehož hodnota se při zpracovávání daného programu nemění, např. přiřazení hodnoty určité adrese.

Desítková konstanta

Číselná hodnota desítkové konstanty se vypisuje v desítkové soustavě.

Konstanta typu INTEGER

Konstanta typu INTEGER je celočíselná hodnota, tzn. posloupnost číslic bez desetinné tečky, se znaménkem nebo bez něj.

Příklady:

`X10`	Přiřazení hodnoty + 10 adrese X
`X-35`	Přiřazení hodnoty -35 adrese X
`X0`	Přiřazení hodnoty 0 adrese X **Upozornění:** X0 není možné nahradit pouhým X.

Konstanty typu REAL

Konstanta typu REAL je posloupnost číslic s desetinnou tečkou a se znaménkem nebo bez něj a s exponentem nebo bez.

Příklady:

`X10.25`	Přiřazení hodnoty +10.25 adrese X
`X-10.25`	Přiřazení hodnoty -10.25 adrese X
`X0.25`	Přiřazení hodnoty +0.25 adrese X
`X.25`	Přiřazení hodnoty +0.25 adrese X bez "0" na začátku
`X=-.1EX-3`	Přiřazení hodnoty $-0.1*10^{-3}$ adrese X

Poznámka

Pokud je adrese, která připouští zadávání desetinných čísel, specifikováno více desetinných míst, než kolik je pro tuto adresu ve skutečnosti k dispozici, bude toto číslo zaokrouhleno, aby zaplnilo všechna dostupná desetinná místa.

Hexadecimální konstanta

Jsou možné také konstanty, které jsou interpretovány jako hexadecimální čísla, tzn v soustavě, jejímž základem je číslo 16. Písmena „A" až „F" jsou přitom považována za hexadecimální čísla s desítkovými hodnotami 10 až 15.

Hexadecimální konstanty jsou uzavřeny apostrofy a začínají písmenem „H" následovaným hodnotou v hexadecimálním formátu. Mezi písmeny a číslicemi jsou povoleny oddělovací znaky.

Příklad:

Programový kód	**Komentář**
`$MC_TOOL_MANAGEMENT_MASK='H7F'`	`; přiřazením hexadecimální konstanty se ve strojním parametru nastavují bity 0-7`

Poznámka

Maximální počet znaků je omezen rozsahem hodnot datového typu Integer.

Binární konstanta

Je možné používat také konstanty, které jsou interpretovány binárně. Používají se přitom pouze čísla „0" a „1".

Binární konstanty jsou označeny apostrofy a začínají písmenem „B" následovaným hodnotou v binárním tvaru. Oddělovací znaky mezi číslicemi jsou povoleny.

Příklad:

Programový kód	**Komentář**
`$MN_AUXFU_GROUP_SPEC='B10000001'`	`; přiřazením binární konstanty se ve strojním parametru nastavují bit 0 a bit 7`

Poznámka

Maximální počet znaků je omezen rozsahem hodnot datového typu Integer.

3.15.7 Operátory a matematické funkce

Operátory

Matematické operátory

Systémové proměnné typu REAL nebo INT mohou být spojovány pomocí následujících operátorů:

Operátor	Význam
+	sečítání
-	odečítání
*	násobení
/	• Dělení v synchronních akcích: INT / INT ⇒ **INT** • Dělení v synchronních akcích s výsledkem typu REAL s využitím funkce ITOR(): ITOR(INT) / ITOR(INT) ⇒ **REAL** • Dělení v NC programech: INT / INT ⇒ **REAL**
DIV	Celočíselné dělení: INT / INT ⇒ **INT**
MOD	Poskytuje zbytek po celočíselném dělení (pouze pro typ INT), dělení modulo Příklad: 3 MOD 4 = 3

Poznámka

Mohou být spojovány pouze proměnné stejného typu.

Relační operátory

Operátor	Význam
==	rovná se
>	nerovná se
<	je menší než
>	je větší než
<=	je menší nebo rovno
>=	je větší nebo rovno

Booleovské operátory

Operátor	Význam
NOT	negace
AND	logické A
OR	logické NEBO
XOR	logické XOR

Bitové logické operace

Operátor	Význam
B_OR	bitové logické NEBO
B_AND	bitové logické A
B_XOR	bitové logické XOR
B_NOT	bitová negace

Priorita operátorů

Při zpracovávání v synchronních akcích mají operátory následující priority (nejvyšší priorita: 1):

Prio.	operátorů	Význam
1	NOT, B_NOT	Negace, bitová negace
2	*, /, DIV, MOD	Násobení, dělení
3	+, -	Sečítání, odečítání
4	B_AND	bitové logické A
5	B_XOR	bitové logické XOR
6	B_OR	bitové logické NEBO
7	AND	logické A
8	XOR	logické XOR
9	OR	logické NEBO
10	<<	Zřetězení řetězců, typ výsledku STRING
11	==, <>, <, >, >=, <=	Relační operátory

Poznámka

Důrazně vám doporučujeme, abyste v případě použití většího počtu operátorů v jednom výrazu pomocí kulatých závorek (...) jednoznačně stanovili prioritu jednotlivých operátorů.

Příklad podmínky s výrazem s několika operátory:

Programový kód

```
... WHEN ($AA_IM[X] > WERT) AND ($AA_IM[Y] > WERT1) DO ...
```

Matematické funkce

Operátor	Význam
SIN()	sinus
COS()	kosinus
TAN()	tangens
ASIN()	arkus sinus
ACOS()	arkus kosinus
ATAN2(,)	arkus tangens na druhou
SQRT()	druhá odmocnina
ABS()	absolutní hodnota

Operátor	Význam
POT()	2. mocnina (čtverec)
TRUNC()	celočíselná část Nastavitelná přesnost v příkazech pro porovnávání pomocí funkce TRUNC
ROUND()	Zaokrouhlení na celé číslo
LN()	přirozený logaritmus
EXP()	exponenciální funkce

Indexování

Index systémové proměnné typu „pole z ...“ může být opět systémovou proměnnou. Index se přitom vyhodnocuje v rámci zpracování hlavní větve programu v taktu interpolátoru.

Příklad

Programový kód

```
... WHEN ... DO $AC_PARAM[ $AC_MARKER[1] ] = 3
```

Omezení

- Vnoření indexování pomocí dalších systémových proměnných není povoleno.
- Index nesmí být vytvořen prostřednictvím proměnných předběžného zpracování. Následující příklad je proto **nepřípustný**, protože $P_EP je proměnná předběžného zpracování: $AC_PARAM[1] = $P_EP[$AC_MARKER[0]]

Viz také

Matematické funkce (Strana 441)

Operace (Strana 441)

Pro pokročilé

4

4.1 Flexibilní programování NC systémů

4.1.1 Proměnné

Prostřednictvím použití proměnných z oblastí systémových dat a uživatelských dat, zejména ve spojení s matematickými funkcemi a řídícími strukturami, mohou být sestavovány flexibilní NC programy a cykly.

⚠ VÝSTRAHA

Hmotné škody a úrazy pracovníků v důsledku změněných proměnných

Když jsou v NC-programu používány proměnné, je potřeba mít na paměti, že pracovník obsluhující stroj nebo nepovolané osoby s odpovídajícími přístupovými právy mohou změnit hodnoty proměnných, a tím ovlivňovat zpracování programu. To může mít za následek hmotné škody nebo úrazy pracovníků.

- Aby se zabránilo negativnímu ovlivňování zpracování programu v důsledku změn hodnot proměnných, je potřeba mít v programu zavedeny odpovídající kontroly dat („validace vstupních hodnot").

- Systémová data
  V systémových datech se nacházejí proměnné, které jsou v systému předem definovány. Tyto proměnné mají definovaný význam. Používají se především v rámci programového vybavení systému. Uživatel může v MC programech a v cyklech tyto proměnné číst a může do nich i zapisovat. Příklad: Strojní parametry, nastavované parametry, systémové proměnné.
  I když je význam systémových dat pevně předem stanoven, jejich vlastnosti může uživatel prostřednictvím opětovných definic v určité míře měnit.
  Viz "Opětovná definice systémových dat, uživatelských dat a příkazů NC jazyka (REDEF) (Strana 403)".
- Uživatelská data
  V uživatelských datech jsou uloženy proměnné definované uživatelem a jejich význam je určen pouze uživatelem. Systém je nevyhodnocuje.
  Uživatelská data jsou rozdělena následujícím způsobem:
  - Předdefinované uživatelské proměnné
    Předdefinované uživatelské proměnné jsou proměnné, které již jsou v systému definovány, a jejich počet může být nastaven pomocí strojních parametrů. Vlastnosti těchto proměnných mohou být uživatelem upravovány.
    Viz "Opětovná definice systémových dat, uživatelských dat a příkazů NC jazyka (REDEF) (Strana 403)".
  - Uživatelem definované proměnné
    Uživatelem definované proměnné jsou proměnné, které jsou definovány uživatelem a v systému jsou založeny při zpracovávání programu. Jejich počet, datový typ, možnosti zobrazování a všechny ostatní vlastnosti jsou definovány výlučně uživatelem.
    Viz "Definice uživatelských proměnných (DEF) (Strana 397)".

4.1.1.1 Systémová data

Systémová data zahrnují v systému předem definované proměnné, které ve NC programech a cyklech umožňují přístup k aktuálnímu nastavení parametrů řídícího systému, ale také ke stavovým informacím o stroji, řídícím systému a o procesech.

Proměnné předběžného zpracování

Proměnné předběžného zpracování jsou systémová data, která jsou čtena a do kterých se zapisuje v souvislosti s předběžným zpracováním, tzn. v okamžiku interpretace bloku, v němž jsou tyto proměnné naprogramovány. Proměnné předběžného zpracování nezpůsobují jeho zastavení.

Proměnné hlavního zpracování

Proměnné hlavního zpracování jsou systémová data, která jsou čtena nebo do kterých se zapisuje v souvislosti s hlavním zpracováním, tzn. v okamžiku zpracování bloku, v němž jsou tyto proměnné naprogramovány. Proměnné hlavního zpracování následující:

- Proměnné, které mohou být naprogramovány v synchronních akcích (čtení/zápis)
- Proměnné, které mohou být naprogramovány v NC programu a které spouštějí zastavení předběžného zpracování (čtení/zápis)
- Proměnné, které mohou být naprogramovány v NC programu a jejichž hodnota se zjišťuje v rámci předběžného zpracování, zapisuje se do nich ale až v rámci hlavního zpracování (synchronizovaně s hlavním zpracováním: jen zápis)

Systém předpon

Kvůli zvláštnímu označování systémových dat se za normálních okolností k jejich názvu připojuje předpona, která se skládá ze znaku $, za nímž následuje jedno nebo dvě písmena a znak podtržení:

$ + 1. Písmeno	**Význam: Druh dat**
Data předběžného zpracování (systémová data, jejichž čtení/zápis se uskutečňují v rámci předběžného zpracování)	
$M	Strojní parametry [1]
$S	Nastavované parametry, chráněné oblasti [1]
$T	Parametry správy nástrojů
$P	Programovatelné hodnoty
$C	Proměnné cyklů v cyklech ISO
$O	Data volitelných doplňků
R	R-Parametry (početní parametry) [2]
Data hlavního zpracování (systémová data, jejichž čtení/zápis se uskutečňují v rámci hlavního zpracování)	
$$M	Strojní parametry [1]
$$S	Nastavované parametry [1]
$A	Aktuální data hlavního zpracování
$V	Data regulátoru polohy
$R	R-Parametry (početní parametry) [2]
[1] To, zda se se strojními nebo s nastavovanými parametry zachází jako s proměnnými předběžného nebo hlavního zpracování programu, závisí na tom, zda jsou zapsány s jedním nebo se dvěma znaky $. Způsob zápisu si můžete zvolit v závislosti na specifické aplikaci. [2] Když jsou R-parametry ve výrobním programu / cyklu používány jako proměnné předběžného zpracování, nezapisuje se žádná předpona, např. R10. V případě použití v synchronní akci jako proměnná hlavního zpracování se předpona zapisuje s jedním znakem $, např. $R10.	

2. písmeno	**Význam: Zobrazování**
N	Globální proměnná NC systému (**N**CK)
C	Kanálová proměnná (**C**hannel)
A	Osová proměnná (**A**xis)

Okrajové podmínky

Výjimky ze systému předpon

Následující systémové proměnné se odlišují od výše popisovaného systému předpon:

- $TC_...: 2. písmeno v tomto případě neodkazuje na kanálové proměnné, nýbrž na systémové proměnné vztahující se k držáku nástroje (TC = Tool Carrier).
- $P_ ...: Kanálové systémové proměnné

Použití strojních a nastavovaných parametrů v synchronních akcích

Při použití strojních a nastavovaných parametrů v synchronních akcích je možné pomocí předpony stanovit, zda se má čtení/zápis daného strojního/nastavovaného parametru uskutečňovat synchronně s předběžným nebo hlavním zpracováním.

Pokud zůstává parametr v průběhu zpracovávání nezměněn, je možno jej číst v průběhu předběžného zpracování. Předpona strojního nebo nastavovaného parametru se pro tento účel zapisuje s jedním znakem $.

```
ID=1 WHENEVER $AA_IM[z] < $SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

Pokud se parametr v průběhu zpracovávání mění, musí jeho čtení/zápis probíhat synchronně s hlavním zpracováním. Předpona strojního nebo nastavovaného parametru se pro tento účel zapisuje se dvěma znaky $.

```
ID=1 WHENEVER $AA_IM[z] < $$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

Poznámka

Zapisování do strojních a nastavovaných parametrů

Při zapisování do strojních nebo nastavovaných parametrů je nutno dávat pozor na to, aby aktivní úroveň přístupových oprávnění při zpracovávání výrobního programu / cyklu umožňovala přístup za účelem zápisu a aby byl v platnosti parametr "IMMEDIATE".

Viz také

Proměnné (Strana 389)

4.1.1.2 Předdefinované uživatelské proměnné: Kanálový početní parametr (R)

Kanálové početní parametry nebo R-parametry jsou předem definované uživatelské proměnné s označením R definované jako pole s datovým typem REAL. Z historických důvodů je pro R-parametry vedle způsobu zápisu s indexem pole, např. `R[10]`, přípustný také způsob zápisu bez indexu pole, např. `R10`.

Při použití v synchronních akcích musí být na začátku uveden znak $, např. `$R10`.

Syntaxe

V případě použití jako proměnná předběžného zpracování:
`R<n>`

```
R[<výraz>]
```

V případě použití jako proměnná hlavního zpracování:

```
$R<n>
$R[<výraz>]
```

Význam

R:	Identifikátor při použití jako proměnná předběžného zpracování, např. ve výrobním programu	
$R:	Identifikátor při použití jako proměnná hlavního zpracování, např. v synchronních akcích	
	Typ:	REAL
	Rozsah hodnot:	V případě neexponenciálního způsobu zápisu: ± (0.000 0001 ... 9999 9999) **Upozornění:** Povoleno je maximálně 8 desetinných míst.
		V případě exponenciálního způsobu zápisu: ± ($1*10^{-300}$ ... $1*10^{+300}$) **Upozornění:** • Způsob zápisu: <mantisa>EX<exponent> např. 8.2EX-3 • Je povoleno maximálně 10 znaků včetně znaménka a desetinné tečky.
<n>:	Číslo R-Parametru	
	Typ:	INT
	Rozsah hodnot:	0 - MAX_INDEX **Upozornění:** MAX_INDEX vyplývá z počtu R-parametrů, který je definován parametrem: MAX_INDEX = (MD28050 $MN_MM_NUM_R_PARAM) - 1
<výraz>:	Index pole Jako index pole může být uveden libovolný výraz, pokud ale výsledek tohoto výrazu může být převeden na datový typ INT (INT, REAL, BOOL, CHAR).	

Příklad:

Přiřazení R-parametru a použití R-parametrů v matematických funkcích:

```
Programový kód              Komentář
R0=3.5678                   ; přiřazení v předběžném zpracování
R[1]=-37.3                  ; přiřazení v předběžném zpracování
R3=-7                       ; přiřazení v předběžném zpracování
$R4=-0.1EX-5                ; přiřazení v hlavním zpracování: R4 = -0.1 * 10^-5
$R[6]=1.874EX8              ; přiřazení v hlavním zpracování: R6 = 1.874 * 10^8
R7=SIN(25.3)                ; přiřazení v předběžném zpracování

R[R2]=R10                   ; nepřímé adresování pomocí R-parametru
```

Programový kód	Komentář
`R[(R1+R2)*R3]=5`	`; nepřímé adresování pomocí matematického výrazu`
`X=(R1+R2)`	`; najíždění osou X na pozici, která vyplývá se součtu R1 a R2`
`Z=SQRT(R1*R1+R2*R2)`	`; najíždění osou Z na pozici odpovídající druhé odmocnině z výrazu (R1^2 + R2^2)`

Viz také

Proměnné (Strana 389)

4.1.1.3 Předdefinované uživatelské proměnné: Globální početní parametry (RG)

Funkce

Kromě kanálových R-parametrů jsou uživateli k dispozici také globální R-parametry. V rámci řídícího systému existují jen jednou a je možno je číst a zapisovat do nich ve všech kanálech.

Globální R-parametry se používají např. pro přenášení informací z jednoho kanálu do druhého. Dalším příkladem mohou být globální nastavení, která mají být vyhodnocována ve všech kanálech, jako je např. vzdálenost o kterou surový obrobek vyčnívá z vřetena.

Čtení a zapisování globálních R-parametrů se uskutečňuje prostřednictvím uživatelského rozhraní nebo při zpracovávání NC-programu. Použití v synchronních akcích nebo v technologických cyklech není možné.

Poznámka

Při čtení a zapisování globálních R-parametrů se mezi kanály neuskutečňuje **žádná** synchronizace.

Protože se čtení a zapisování provádí v rámci předběžného zpracování, okamžik, ve kterém je hodnota zapisovaná v jednom kanálu v platnosti ve druhém kanálu, není definován.

Příklad:

V kanálu 1 je zpracovávána programová smyčka, ve které se R-parametr používá jako počítadlo průchodů. V kanálu 2 se zapisuje do tohoto globálního R-parametru hodnota, která způsobuje přerušení zpracování smyčky v kanálu 1. Všechny smyčky, které jsou až do tohoto okamžiku interpretovány v rámci předběžného zpracování v kanálu 1, jsou však ještě zpracovávány. Kolik ale těchto smyček je, není definováno a závisí to mimo jiné na vytížení daného kanálu.

Synchronizaci mezi kanály musí tedy uživatel realizovat sám, např. pomocí značek WAIT!

Syntaxe

Zápis v NC programu

```
RG[<n>]=<hodnota>
RG[<výraz>]=<hodnota>
```

Čtení v NC programu
`R...=RG[<n>]`
`R...=RG[<výraz>]`

Význam

`RG`:	Předdefinovaný název adresy NC systému pro **globální** R-parametry **Poznámka:** Název této NC-adresy může být nastaven pomocí strojního parametru MD15800 $MN_R_PARAM_NCK_NAME.	
`<n>`:	Číslo globálního R-parametru	
	Typ:	INT
	Rozsah hodnot:	0 ... MAX_INDEX **Poznámka** MAX_INDEX vyplývá z počtu globálních R-parametrů, který je definován parametrem: MAX_INDEX = (MD18156 $MN_MM_NUM_R_PARAM_NCK) - 1
`<výraz>`:	Jako index pole může být uveden libovolný výraz, pokud ale výsledek tohoto výrazu může být převeden na datový typ INT (INT, REAL, BOOL, CHAR).	
`<hodnota>`:	Hodnota globálního R-parametru	
	Typ:	REAL
	Rozsah hodnot:	V případě neexponenciálního způsobu zápisu: ± (0.000 0001 ... 9999 9999) **Poznámka:** Povoleno je maximálně 8 desetinných míst.
		V případě exponenciálního způsobu zápisu: ± ($1*10^{-300}$ ... $1*10^{+300}$) **Poznámka:** • Způsob zápisu: <mantisa>EX<exponent> např. 8.2EX-3 • Je povoleno maximálně 10 znaků včetně znaménka a desetinné tečky.

4.1.1.4 Předdefinované uživatelské proměnné: linkové proměnné

Prostřednictvím linkových proměnných mohou být v rámci funkce "NCU-Link" cyklicky vyměňována data mezi jednotkami NCU, které jsou spolu spojeny pomocí sítě. Tyto jednotky přitom umožňují přístup do paměti linkových proměnných v závislosti na formátu těchto dat. Velikost a také datová struktura paměti linkových proměnných je definována uživatelem / výrobcem zařízení tak, aby vyhovovala danému zařízení.

Linkové proměnné jsou globální systémové uživatelské proměnné, jejichž čtení a zápis je možné v případě linkové komunikace nastavené v konfiguraci provádět všemi jednotkami NCU linkového svazku pomocí výrobních programů a cyklů. Oproti globálním uživatelským proměnným (GUD) se mohou linkové proměnné používat také v synchronních akcích.

U zařízení bez aktivního modulu NCU-Link mohou být linkové proměnné používány lokálně v řídícím systému vedle globálních uživatelských proměnných (GUD) jako další globální uživatelské proměnné.

Syntaxe

```
$A_DLB[<Index>]
$A_DLW[<Index>]
$A_DLD[<Index>]
$A_DLR[<Index>]
```

Význam

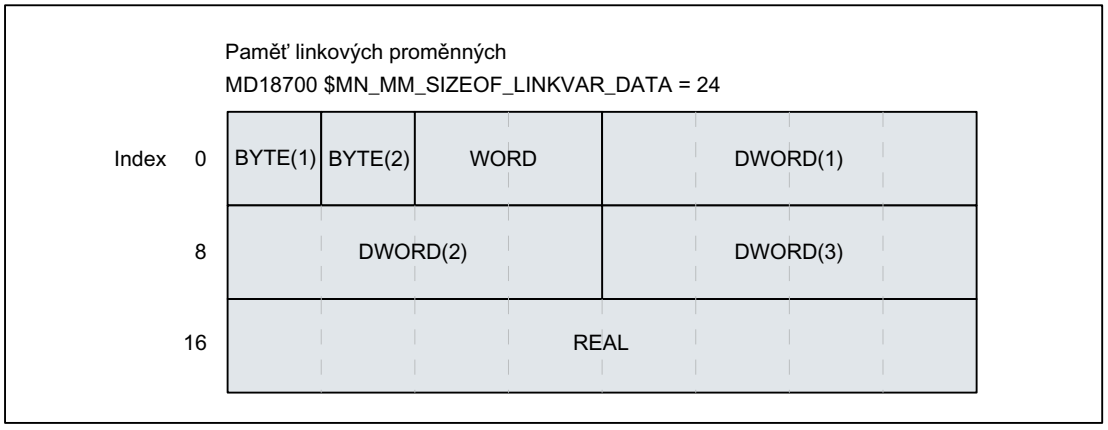
`$A_DLB`:	Linková proměnná pro datový formát BYTE (1 byte)	
	Datový typ:	UINT
	Rozsah hodnot:	0 ... 255
`$A_DLW`:	Linková proměnná pro datový formát WORD (2 byty)	
	Datový typ:	INT
	Rozsah hodnot:	-32768 ... 32767
`$A_DLD`:	Linková proměnná pro datový formát DWORD (4 byty)	
	Datový typ:	INT
	Rozsah hodnot:	-2147483648 ... 2147483647
`$A_DLR`:	Linková proměnná pro datový formát REAL (8 bytů)	
	Datový typ:	REAL
	Rozsah hodnot:	$\pm(2{,}2*10^{-308}$ ... $1{,}8*10^{+308})$
`<Index>`:	Index adresy v bytech, vypočítaný od začátku paměti linkových proměnných	
	Datový typ:	INT
	Rozsah hodnot:	0 - MAX_INDEX **Upozornění** • MAX_INDEX vyplývá z velikosti paměti linkových proměnných, která je definována parametrem: MAX_INDEX = (MD18700 $MN_MM_SIZEOF_LINKVAR_DATA) - 1 • V programech se smí používat jedině indexy, takže byty adresované v paměti linkových proměnných leží na hranici datového formátu => Index = n * byty, kde n = 0, 1, 2, ... – `$A_DLB[i]`: i = 0, 1, 2, ... – `$A_DLW[i]`: i = 0, 2, 4, ... – `$A_DLD[i]`: i = 0, 4, 8, ... – `$A_DLR[i]`: i = 0, 8, 16, ...

Příklad:

V automatickém zařízení jsou 2 jednotky NCU (NCU1 a NCU2). K NCU1 je připojena osa stroje AX2, jejíž pohyb má být ovládán z NCU2 mechanismem linkové osy.

NCU1 cyklicky zapisuje skutečnou hodnotu proudu ($VA_CURR) osy AX2 do paměti linkových proměnných. NCU2 cyklicky čte skutečnou hodnotu proudu přenášenou pomocí linkové komunikace a při překročení mezní hodnoty vypisuje alarm 61000.

Datová struktura v paměti linkových proměnných je uvedena na následujícím obrázku. Skutečná hodnota proudu se přenáší pomocí hodnoty typu REAL.



NCU1

NCU1 cyklicky v taktu IPO zapisuje v rámci statické synchronní akce skutečnou hodnotu proudu osy AX2 prostřednictvím linkové proměnné $A_DLR[16] do paměti linkových proměnných.

Programový kód

```
N111 IDS=1 WHENEVER TRUE DO $A_DLR[16]=$VA_CURR[AX2]
```

NCU2

NCU2 cyklicky v taktu IPO čte v rámci statické synchronní akce skutečnou hodnotu proudu osy AX2 prostřednictvím linkové proměnné $A_DLR[16] z paměti linkových proměnných. Pokud je skutečná hodnota proudu větší než 23,0 A, aktivuje se alarm 61000.

Programový kód

```
N222 IDS=1 WHEN $A_DLR[16] > 23.0 DO SETAL(61000)
```

4.1.1.5 Definice uživatelských proměnných (DEF)

Pomocí příkazu `DEF` mohou být specifické uživatelské proměnné nebo uživatelské proměnné (User Data) definovány a je možno jim dosadit hodnoty.

V závislosti na oblasti jejich platnosti, tzn. oblasti, ve které se proměnná zobrazuje, existují následující kategorie uživatelských proměnných:

- Lokální uživatelské proměnné (LUD)
  Lokální uživatelské proměnné (LUD) jsou proměnné, které jsou definovány v NC programu, který v okamžiku zpracovávání není hlavním programem. Jsou založeny při vyvolávání NC programu a jsou vymazány s resetem na konci programu, příp. při následujícím spouštění řídícího systému. Přístup k LUD je možný pouze v rámci NC programu, v němž jsou definovány.
- Programově globální uživatelské proměnné (PUD)
  Programově globální uživatelské proměnné (PUD) jsou proměnné, které jsou definovány v NC programu, který je používán jako hlavní program. Jsou založeny při vyvolávání NC programu a jsou vymazány s resetem na konci programu, příp. při následujícím spouštění řídícího systému. Přístup k PUD je k dispozici v hlavním programu a ve všech podprogramech.

  Poznámka

  Dostupnost programových globálních uživatelských proměnných (PUD)

  Programové globální uživatelské proměnné (PUD) definované v hlavním programu jsou k dispozici i v podprogramech jen tehdy, pokud je nastaven následující strojní parametr:

  MD11120 $MN_LUD_EXTENDED_SCOPE = 1

  Pokud je MD11120 = 0, jsou programové globální uživatelské proměnné definované v hlavním programu k dispozici pouze v tomto hlavním programu.

- Globální uživatelské proměnné (GUD)
  Globální uživatelské proměnné (GUD) jsou v globálními proměnnými v NC systému, příp. v kanálu, které jsou definovány v datovém modulu (SGUD, MGUD, UGUD, GUD4 ... GUD9) a které zůstávají zachovány i po resetu na konci programu, příp. po následujícím spuštění řídícího systému. Přístup ke GUD je k dispozici ve všech NC programech.

Uživatelské proměnné musí být před svým použitím (čtení / zápis) definovány. Přitom je potřeba dbát na následující pravidla:

- GUD musí být definována v definičním souboru, např. _N_DEF_DIR/_N_UGUD_DEF.
- PUD a LUD musí být definovány v definiční části NC programu.
- Definice dat se musí provádět v samostatném bloku.
- V každé definici dat smí být použit jen jeden datový typ.
- V každé definici dat může být definován větší počet proměnných stejného datového typu.

Syntaxe

LUD a PUD

```
DEF <typ> <fyzikální jednotka> <mezní hodnota> <název>[<hodnota_1>,
<hodnota_2>, <hodnota_3>]=<počáteční_hodnota>
```

GUD

```
DEF <oblast> <VL_Stop> <přístupová oprávnění> <datová třída> <typ>
<fyzikální jednotka> <mezní hodnota> <název>[<hodnota_1>,
<hodnota_2>, <hodnota_3>]=<počáteční_hodnota>
```

Význam

DEF:	Příkaz pro definici uživatelských proměnných GUD, PUD, LUD	
<oblast>:	Oblast platnosti, týká se pouze GUD:	
	NC:	Uživatelská proměnná globální v NC systému
	CHAN:	Uživatelská proměnná globální v kanálu
<VL_Stop>:	Zastavení předběžného zpracování, týká se pouze GUD (není nutné)	
	SYNR:	Zastavení předběžného zpracování při čtení
	SYNW:	Zastavení předběžného zpracování při zápisu
	SYNRW:	Zastavení předběžného zpracování při čtení/zápisu
<přístupová oprávnění>:	Úroveň ochrany pro čtení / zápis GUD prostřednictvím NC programu nebo BTSS (není nutné)	
	APRP <úroveň ochrany>:	Čtení: NC program
	APWP <úroveň ochrany>:	Zápis: NC program
	APRB <úroveň ochrany>:	Čtení: BTSS
	APWB <úroveň ochrany>:	Zápis: BTSS
	<úroveň ochrany>:	Rozsah hodnot: 0 ... 7
	Viz "Atribut: Přístupová oprávnění (APR, APW, APRP, APWP, APRB, APWB) (Strana 413)".	
<Datová třída>:	Přiřazení datové třídy (pouze SINUMERIK 828D!)	
	DCM:	Datová třída M (= Manufacturer (výrobce))
	DCI:	Datová třída I (= Individual (individuální))
	DCU:	Datová třída U (= User (Uživatel))
<typ>:	Datový typ:	
	INT:	Celočíselné hodnoty se znaménkem
	REAL:	Reálné číslo (LONG REAL podle IEEE)
	BOOL:	Boolovská hodnota TRUE (1) / FALSE (0)
	CHAR:	Znaky ASCII
	STRING[<Max. délka>]:	Řetězec znaků definované délky
	AXIS:	Identifikátor osy/vřetena
	FRAME:	Geometrické údaje pro statickou transformaci souřadného systému
	Viz "Datové typy (Strana 424)".	
<fyzikální jednotka>:	Fyzikální jednotka (není nutné)	
	PHU <jednotka>:	fyzikální jednotka
	Viz "Atribut: Fyzikální jednotka (PHU) (Strana 411)".	
<mezní hodnota>:	horní / dolní mezní hodnota (není nutné)	
	LLI <mezní hodnota>:	spodní mezní hodnota (lower limit)
	ULI <mezní hodnota>:	horní mezní hodnota (upper limit)
	Viz "Atribut: Mezní hodnoty (LLI, ULI) (Strana 409)".	

<název>:	Název proměnné **Poznámka** • Maximálně 31 znaků • První dva znaky musí být písmena a/nebo znak podtržení. • Znak "$" je vyhrazen pro systémové proměnné a nesmí se používat.
[<hodnota_1>, <hodnota_2>, <hodnota_3>]:	Údaje o velikosti pole pro 1- až 3-rozměrné proměnné typu pole (není nutné) Pokud budete potřebovat informace o inicializaci proměnných typu pole, viz "Definice a inicializace proměnných typu pole (DEF, SET, REP) (Strana 419)".
<počáteční hodnota>:	Inicializační hodnota (není nutné) viz "Atribut: Inicializační hodnota (Strana 406)". Pokud budete potřebovat informace o inicializaci proměnných typu pole, viz "Definice a inicializace proměnných typu pole (DEF, SET, REP) (Strana 419)".

Příklady

Příklad 1: Definice uživatelských proměnných v datovém modulu pro výrobce stroje

Programový kód	**Komentář**
%_N_MGUD_DEF	; Modul GUD: Výrobce stroje
$PATH=/_N_DEF_DIR	
DEF CHAN REAL PHU 24 LLI 0 ULI 10 STROM_1, STROM_2	
; popis	
; definice dvou GUD: STROM_1, STROM_2	
; oblast platnosti: v kanálu	
; datový typ: REAL	
; zastavení předběžného zpracování: není naprogramováno => předdefinovaná hodnota = žádné zastavení předběžného zpracování (VL-Stop)	
; fyzikální Jednotka: 24 = [A]	
; mezní hodnota: Low = 0.0, High = 10.0	
; přístupová oprávnění: není naprogramováno --> předdefinovaná hodnota = 7 = poloha přepínače na klíč 0	
; inicializační hodnota: není naprogramováno --> předdefinovaná hodnota = 0.0	
DEF NCK REAL PHU 13 LLI 10 APWP 3 APRP 3 APWB 0 APRB 2 ZEIT_1=12, ZEIT_2=45	
; popis	
; definice dvou GUD: ZEIT_1, ZEIT_2	
; oblast platnosti: V celém NC systému	
; datový typ: REAL	
; zastavení předběžného zpracování: není naprogramováno => předdefinovaná hodnota = žádné zastavení předběžného zpracování (VL-Stop)	
; fyzikální Jednotka: 13 = [s]	
; mezní hodnota: Low = 10.0, High = není naprogramováno => horní hranice definiční oblasti	
; přístupová oprávnění:	

Programový kód	Komentář
; NC program: Čtení/zápis = 3 = uživatel	
; BTSS: Zápis = 0 = Siemens, Čtení = 3 = uživatel	
; inicializační hodnota: ZEIT_1 = 12.0, ZEIT_2 = 45.0	
DEF NCK APWP 3 APRP 3 APWB 0 APRB 3 STRING[5] GUD5_NAME = "COUNTER"	
; popis	
; definice jednoho GUD: GUD5_NAME	
; oblast platnosti: V celém NC systému	
; datový typ: STRING, max. 5 znaků	
; zastavení předběžného zpracování: není naprogramováno => předdefinovaná hodnota = žádné zastavení předběžného zpracování (VL-Stop)	
; fyzikální Jednotka: není naprogramováno --> předdefinovaná hodnota = 0 = žádná fyzikální jednotka	
; mezní hodnota: není naprogramováno => hranice definiční oblasti: Low = 0, High = 255	
; přístupová oprávnění:	
; NC program: Čtení/zápis = 3 = uživatel	
; BTSS: Zápis = 0 = Siemens, Čtení = 3 = uživatel	
; inicializační hodnota: "COUNTER"	
M30	

Příklad 2: V programu globální a lokální uživatelské proměnné (PUD / LUD)

Programový kód	Komentář
PROC MAIN	; hlavní program:
DEF INT VAR1	; definice PUD
...	
SUB2	; volání podprogramu
...	
M30	

Programový kód	Komentář
PROC SUB2	; podprogram SUB2
DEF INT VAR2	; DEFINICE LUD
...	
IF (VAR1==1)	; čtení PUD
VAR1=VAR1+1	; čtení a zápis PUD
VAR2=1	; zápis LUD
ENDIF	
SUB3	; volání podprogramu
...	
M17	

Programový kód	Komentář
PROC SUB3	; podprogram SUB3
...	

Programový kód	Komentář
IF (VAR1==1)	; čtení PUD
VAR1=VAR1+1	; čtení a zápis PUD
VAR2=1	; chyba: LUD ze SUB2 není známa
ENDIF	
...	
M17	

Příklad 3: Definice a použití uživatelských proměnných datového typu AXIS

Programový kód	Komentář
DEF AXIS ABSZISSE	; 1. Geometrická osa
DEF AXIS SPINDLE	; vřeteno
...	
IF ISAXIS(1) == FALSE GOTOF WEITER	
ABSZISSE = $P_AXN1	
WEITER:	
...	
SPINDLE=(S1)	; 1. Vřeteno
OVRA[SPINDLE]=80	; korekce vřetena = 80%
SPINDLE=(S3)	; 3. Vřeteno

Okrajové podmínky

Globální uživatelské proměnné (GUD)

V rámci definice globálních uživatelských proměnných (GUD) je potřeba mít na paměti následující strojní parametry:

Č.	identifikátoru: $MN_	Význam
11140	GUD_AREA_ SAVE_TAB	Doplňkové zálohování modulů GUD
18118 [1)]	MM_NUM_GUD_MODULES	Počet souborů GUD v aktivním systému souborů
18120 [1)]	MM_NUM_GUD_NAMES_NCK	Počet názvů globálních GUD
18130 [1)]	MM_NUM_GUD_NAMES_CHAN	Počet názvů kanálových GUD
18150 [1)]	MM_GUD_VALUES_MEM	Paměťový prostor pro hodnoty globálních GUD
18660 [1)]	MM_NUM_SYNACT_GUD_REAL	Počet GUD typu REAL, které mohou být nastaveny v konfiguraci
18661 [1)]	MM_NUM_SYNACT_GUD_INT	Počet GUD typu INT, které mohou být nastaveny v konfiguraci
18662 [1)]	MM_NUM_SYNACT_GUD_BOOL	Počet GUD typu BOOL, které mohou být nastaveny v konfiguraci
18663 [1)]	MM_NUM_SYNACT_GUD_AXIS	Počet GUD typu AXIS, které mohou být nastaveny v konfiguraci

Č.	identifikátoru: $MN_	Význam
18664 [1)]	MM_NUM_SYNACT_GUD_CHAR	Počet GUD typu CHAR, které mohou být nastaveny v konfiguraci
18665 [1)]	MM_NUM_SYNACT_GUD_STRING	Počet GUD typu STRING, které mohou být nastaveny v konfiguraci

[1)] V systému SINUMERIK 828D je tento MD jen ke čtení!

Použití uživatelských proměnných datového typu AXIS globálních v NC systému přes hranice kanálu

V NC globální uživatelské proměnné datového typu `AXIS`, které byly při definici datového modulu inicializovány identifikátorem osy, mohou být používány v různých kanálech NC systému jedině tehdy, pokud má osa v těchto kanálech stejné číslo kanálové osy.

Jestliže tomu tak není, musí být proměnná načtena na začátku programu NC systému, jak ukazuje následující příklad, musí být použita funkce AXNAME(...) (viz "Osové funkce (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Strana 887)").

Programový kód	Komentář
`DEF NCK STRING[5] ACHSE="X"`	`; definice v datovém modulu`
`...`	
`N100 AX[AXNAME(ACHSE)]=111 G00`	`; použití v NC programu`

4.1.1.6 Opětovná definice systémových dat, uživatelských dat a příkazů NC jazyka (REDEF)

Pomocí příkazu `REDEF` mohou být změněny atributy systémových dat, uživatelských dat a příkazů NC jazyka. Základním předpokladem pro změnu definice je, že musí být uskutečněna časově až po odpovídající definici.

V rámci opětovné definice není možné měnit větší počet atributů současně. Pro každý atribut, který má být změněn, musí být naprogramován vlastní příkaz `REDEF`.

Jestliže je naprogramováno několik vzájemně si odporujících změn atributu, je v platnosti vždycky ta poslední změna.

Obnovení původních hodnot atributů

Pro atributy pro přístupová oprávnění, které byly změněny příkazem `REDEF`, a okamžik inicializace mohou být obnovena jejich původní předdefinovaná nastavení prostřednictvím opětovného naprogramování příkazu `REDEF`, za nímž následuje název proměnné nebo příkaz NC jazyka.

- Přístupová oprávnění: Úroveň ochrany 7
- Okamžik inicializace: žádná inicializace, příp. zachování aktuálních hodnot

Atributy, jejichž definice může být změněna

Viz "Přehled definovatelných a znovu definovatelných atributů (Strana 417)".

Lokální uživatelské proměnné (PUD / LUD)

Pro lokální uživatelské proměnné (PUD / LUD) se žádné změny definice nesmí provádět.

Syntaxe

```
REDEF <název> <VL_Stop>
REDEF <název> <fyzikální jednotka>
REDEF <název> <mezní hodnota>
REDEF <název> <přístupová oprávnění>
REDEF <název> <okamžik inicializace>
REDEF <název> <okamžik inicializace> <počáteční hodnota>
REDEF <název> <Datová třída>
REDEF <název>
```

Význam

REDEF:	Příkaz pro opětovnou definici určitého atributu, příp. pro obnovení původního nastavení atributu "Přístupová oprávnění" a/nebo "Okamžik inicializace" pro systémové proměnné, uživatelské proměnné a příkazy NC jazyka	
<název>:	Název již definované proměnné nebo příkazu NC jazyka	
<VL-Stop>:	Zastavení předběžného zpracování	
	SYNR:	Zastavení předběžného zpracování při čtení
	SYNW:	Zastavení předběžného zpracování při zápisu
	SYNRW:	Zastavení předběžného zpracování při čtení/zápisu
<fyzikální jednotka>:	Fyzikální jednotka	
	PHU <jednotka>:	fyzikální jednotka
	Viz "Atribut: Fyzikální jednotka (PHU) (Strana 411)". **Poznámka:** Změny definice jsou nepřípustné pro následující objekty: • Systémové proměnné • Globální uživatelská data (GUD) následujících datových typů: BOOL, AXIS, STRING, FRAME	
<mezní hodnota>:	horní / dolní mezní hodnota	
	LLI <mezní hodnota>:	spodní mezní hodnota (lower limit)
	ULI <mezní hodnota>:	horní mezní hodnota (upper limit)
	Viz "Atribut: Mezní hodnoty (LLI, ULI) (Strana 409)". **Poznámka:** Změny definice jsou nepřípustné pro následující objekty: • Systémové proměnné • Globální uživatelská data (GUD) následujících datových typů: BOOL, AXIS, STRING, FRAME	

<přístupová oprávnění>:	Přístupová oprávnění pro čtení / zápis prostřednictvím výrobního programu nebo BTSS	
	APX <úroveň ochrany>:	Spouštění: Prvek NC jazyka
	APRP <úroveň ochrany>:	Čtení: Výrobní program
	APWP <úroveň ochrany>:	Zápis: Výrobní program
	APRB <úroveň ochrany>:	Čtení: BTSS
	APWB <úroveň ochrany>:	Zápis: BTSS
	<úroveň ochrany>:	Rozsah hodnot: 0 ... 7
	Viz "Atribut: Přístupová oprávnění (APR, APW, APRP, APWP, APRB, APWB) (Strana 413)".	
<okamžik inicializace>:	Okamžik, ve kterém je proměnná znovu inicializována	
	INIPO:	Power On
	INIRE:	Konec hlavního programu, reset NC systému nebo Power-On
	INICF:	Příkaz NEWCONF nebo konec hlavního programu, reset NC systému nebo Power-On
	PRLOC:	Konec hlavního programu, reset NC systému po lokální změně nebo Power-On
	Viz "Atribut: Inicializační hodnota (Strana 406)".	
<počáteční hodnota>:	Inicializační hodnota Při opětovné definici inicializační hodnoty musí být vždy zadán také časový okamžik inicializace (viz <okamžik inicializace>). Viz "Atribut: Inicializační hodnota (Strana 406)". Pokud budete potřebovat informace o inicializaci proměnných typu pole, viz "Definice a inicializace proměnných typu pole (DEF, SET, REP) (Strana 419)". **Poznámka:** Změny definice jsou nepřípustné pro systémové proměnné, kromě nastavovaných parametrů.	
<Datová třída>:	Přiřazení datové třídy (pouze SINUMERIK 828D!)	
	DCM:	Datová třída M (= Manufacturer (výrobce))
	DCI:	Datová třída I (= Individual (individuální))
	DCU:	Datová třída U (= User (Uživatel))

Příklad:

Opětovná definice systémové proměnné $TC_DPCx v datovém modulu pro výrobce stroje

Programový kód

```
%_N_MGUD_DEF                                ; Modul GUD: Výrobce stroje
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
```

k bloku N100:	Přístupová oprávnění pro zápis: BTSS = úroveň ochrany 2, výrobní program = úroveň ochrany 3
k bloku N200:	Fyzikální jednotka [%]
k bloku N300:	spodní mezní hodnota = 0, horní mezní hodnota = 200
k bloku N400:	Proměnná typu pole se při zapnutí systému (Power On) inicializuje danými čtyřmi hodnotami.
k bloku N800/ N900:	Obnovení původního nastavení atributu "Přístupová oprávnění" a/nebo "Okamžik inicializace"

Poznámka

Použití souborů typu ACCESS

Jestliže se používají soubory typu ACCESS, musí být opětovná definice přístupových oprávnění přenesena z _N_MGUD_DEF do _N_MACCESS_DEF.

Okrajové podmínky

Granularita

Opětovná definice se vždy vztahuje na celé, svými názvy jednoznačně určené proměnné. Není tedy možné např. u proměnných typu pole, přiřazovat jednotlivým prvkům pole odlišné hodnoty atributů.

4.1.1.7 Atribut: Inicializační hodnota

Definice (DEF) uživatelských proměnných

Při definici je možné pro následující uživatelské proměnné předem zadat inicializační hodnotu:

- globální uživatelské proměnné (GUD)
- programově globální uživatelské proměnné (PUD)
- lokální uživatelské proměnné (LUD)

Opětovná definice (REDEF) systémových a uživatelských proměnných

Při opětovné definici je možné pro následující proměnné předem zadat inicializační hodnotu:

- Systémová data
  - Nastavované parametry
- Uživatelská data
  - R-Parametry
  - Proměnné synchronních akcí ($AC_MARKER, $AC_PARAM, $AC_TIMER)
  - GUD synchronních akcí (SYG_xy[ ], kde x = R, I, B, A, C, S a y = S, M, U, 4, ..., 9)
  - Parametry EPS
  - Parametry nástroje OEM
  - Parametry zásobníku OEM
  - globální uživatelské proměnné (GUD)

Okamžik opětovné inicializace

Při opětovné definici může být zadán časový okamžik, ve kterém má být proměnná znovu inicializována, tzn. kdy má být znovu dosazena inicializační hodnota:

- `INIPO` (Power On)
  Proměnná se znovu inicializuje při zapnutí systému (Power On).
- `INIRE` (Reset)
  Proměnná bude znovu inicializována při resetu NC systému, resetu BAG, na konci výrobního programu (`M02` / `M30`) nebo při zapnutí systému (Power On).
- `INICF` (NEWCONF)
  Proměnná bude znovu inicializována při aktivování funkce „Aktivování strojních parametrů" prostřednictvím HMI, při zpracování příkazu `NEWCONF` ve výrobním programu, při resetu NC systému, resetu BAG, na konci výrobního programu (M02 / M30) nebo při zapnutí systému (Power On).
- `PRLOC` (programově lokální změna)
  Proměnná se při resetu NC systému, resetu BAG nebo na konci výrobního programu (M02 / M30) znovu inicializuje jen tehdy, jestliže byla v rámci aktuálního výrobního programu změněna.
  Atribut `PRLOC` se smí používat jedině v souvislosti s programovatelnými nastavovanými parametry (viz následující tabulka).

Tabulka 4-1 Programovatelný nastavovaný parametr

Číslo	Identifikátor	G-příkaz [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`

Číslo	Identifikátor	G-příkaz [1)]
43300	$SA_ASSIGN_FEED_PER_REV_SOURCE	FPRAON
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) pomocí tohoto G-příkazu je nastavovaný parametr vyžádán		

Okrajové podmínky

Inicializační hodnota: globální uživatelské proměnné (GUD)

- Pro globální uživatelské proměnné (GUD), jejichž oblastí platnosti je NC, je možné jako okamžik inicializace zadat jedině INIPO (Power On).
- Pro globální uživatelské proměnné (GUD), jejichž oblastí platnosti je CHAN, je možné jako okamžik inicializace zadat vedle INIPO (Power On) také INIRE (Reset) nebo INICF (NEWCONF).
- U globálních uživatelských proměnných (GUD), které mají oblast platnosti CHAN a okamžik inicializace INIRE (Reset) nebo INICF (NEWCONF), jsou v případě resetu NC systému, resetu BAG a příkazu „Aktivování strojních parametrů“ proměnné nově inicializovány pouze v těch kanálech, v nichž byla výše uvedená událost uskutečněna.

Inicializační hodnota: Datový typ FRAME

Pro proměnné datového typu FRAME nesmí být žádná inicializační hodnota zadávána. Proměnné datového typu FRAME jsou vždy implicitně inicializovány předdefinovaným framem.

Inicializační hodnota: Datový typ CHAR

Pro proměnné datového typu CHAR může být namísto ASCII kódu (0...255) naprogramován také odpovídající znak ASCII v uvozovkách, např. "A".

Inicializační hodnota: Datový typ STRING

U proměnných datového typu STRING musí být dosazen řetězec znaků v uvozovkách, např.: ...= "MACHINE_1".

Inicializační hodnota: Datový typ AXIS

Pro proměnné datového typu `AXIS` musí být v případě rozšířeného způsobu zápisu adresy dosazen identifikátor osy v závorkách, např.: ...= (X3).

Inicializační hodnota: Systémová proměnná

Pro systémové proměnné nemohou být prostřednictvím opětovné definice zadávány žádné specifické uživatelské inicializační hodnoty. Inicializační hodnoty systémových proměnných jsou pevně definovány v systému. Opětovnou definicí může být ale změněn okamžik (`INIRE`, `INICF`), kdy má být systémová proměnná znovu inicializována.

Implicitní inicializační hodnota: Datový typ AXIS

Pro proměnné datového typu `AXIS` se používají následující implicitní inicializační hodnoty:

- Systémová data: "první geometrická osa"
- GUD synchronních akcí (označení: SYG_A*), PUD, LUD:
  Identifikátor osy ze strojního parametru: MD20082
  $MC_AXCONF_CHANAX_DEFAULT_NAME

Implicitní inicializační hodnota: Data nástrojů a zásobníků

Pro data nástrojů a zásobníků mohou být inicializační hodnoty předem definovány prostřednictvím následujícího strojního parametru: MD17520
$MN_TOOL_DEFAULT_DATA_MASK

Poznámka

Synchronizace

Za synchronizaci událostí, které spouštějí opětovnou inicializaci globální proměnné, se čtením této proměnné na jiném místě nese veškerou odpovědnost uživatel / výrobce stroje.

Viz také

Proměnné (Strana 389)

4.1.1.8 Atribut: Mezní hodnoty (LLI, ULI)

Horní a dolní mezní hodnotu definiční oblasti je možné zadat jen pro následující datové typy.

- INT
- REAL
- CHAR

Definice (DEF) uživatelských proměnných: Mezní hodnoty a implicitní inicializační hodnoty

Jestliže není při definici uživatelských proměnných jednoho z výše uvedených datových typů definována žádná explicitní inicializační hodnota, bude proměnné dosazena implicitní inicializační hodnota datového typu:

- INT: 0
- REAL : 0.0
- CHAR: 0

Jestliže implicitní inicializační hodnota leží mimo definiční oblast stanovenou naprogramovanými mezními hodnotami, bude proměnná inicializována tou mezní hodnotou, která leží blíže k implicitní inicializační hodnotě:

- implicitní inicializační hodnota < spodní mezní hodnota (LLI) =>
  Inicializační hodnota = spodní mezní hodnota
- implicitní inicializační hodnota > horní mezní hodnota (ULI) =>
  Inicializační hodnota = horní mezní hodnota

Příklady:

Programový kód	**Komentář**

`DEF REAL GUD1`	`; spodní mezní hodnota = hranice definiční oblasti`
	`; horní mezní hodnota = hranice definiční oblasti`
	`; žádná inicializační hodnota není naprogramována`
	`; --> implicitní inicializační hodnota = 0.0`

`DEF REAL LLI 5.0 GUD2`	`; spodní mezní hodnota = 5,0`
	`; horní mezní hodnota = hranice definiční oblasti`
	`; => inicializační hodnota = 5,0`

`DEF REAL ULI -5 GUD3`	`; spodní mezní hodnota = hranice definiční oblasti`
	`; horní mezní hodnota = -5,0`
	`; => inicializační hodnota = -5,0`

Opětovná definice (REDEF) uživatelských proměnných: mezní hodnoty a aktuální skutečné hodnoty

Jestliže jsou při opětovné definici mezních hodnot uživatelské proměnné tyto mezní hodnoty změněny tak, že aktuální skutečná hodnota leží mimo novou definiční oblast, aktivuje se alarm a mezní hodnoty se nepřevezmou.

Poznámka

Opětovná definice (REDEF) uživatelských proměnných

Při opětovné definici mezních hodnot uživatelské proměnné je potřeba dávat pozor, aby změny následujících hodnot byly konzistentní:

- Mezní hodnoty
- Skutečná hodnota
- Inicializační hodnota při opětovné definici a při automatické opětovné inicializaci na základě INIPO, INIRE nebo INICF

Viz také

Proměnné (Strana 389)

4.1.1.9 Atribut: Fyzikální jednotka (PHU)

Fyzikální jednotka může být zadána pouze pro proměnné následujících datových typů:

- INT
- REAL

Programovatelná fyzikální jednotka (PHU)

Fyzikální jednotka se zadává jako číslo s pevnou řádovou čárkou: `PHU <jednotka>`

Mohou být naprogramovány následující fyzikální jednotky:

<Jednotka>	Význam	Fyzikální jednotka
0	Žádná fyzikální jednotka	-
1	Lineární nebo úhlová poloha [1)2)]	[mm], [palce], [stupně]
2	Lineární poloha [2)]	[mm], [palce]
3	Úhlová poloha	[stupně]
4	Lineární nebo úhlová rychlost [1)2)]	[mm/min], [palce/min], [ot/min]
5	Lineární rychlost [2)]	[mm/min]
6	Úhlová rychlost	[ot/min]
7	Lineární nebo úhlové zrychlení [1)2)]	[m/s^2], [$palce/s^2$], [ot/s^2]
8	Lineární zrychlení [2)]	[m/s^2], [$palce/s^2$]
9	Úhlové zrychlení	[ot/s^2]
10	Lineární nebo úhlový ryv [1)2)]	[m/s^3], [$palce/s^3$], [ot/s^3]
11	Lineární ryv [2)]	[m/s^3], [$palce/s^3$]
12	Úhlový ryv	[ot/s^3]
13	Čas	[s]
14	Zesílení regulátoru polohy	[16.667/s]
15	Otáčkový posuv [2)]	[mm/ot], [palce/ot]
16	Teplotní kompenzace [1)2)]	[mm], [palce]
18	Síla	[N]
19	Hmotnost	[kg]
20	Moment setrvačnosti [3)]	[kgm^2]
21	Procenta	[%]
22	Frekvence	[Hz]
23	Napětí	[V]
24	Proud	[A]
25	Teplota	[°C]
26	Úhel	[stupně]
27	KV	[1000/min]
28	Lineární nebo úhlová poloha [3)]	[mm], [palce], [stupně]

<Jednotka>	Význam	Fyzikální jednotka
29	Řezná rychlost [2)]	[m/min], [stopy/min]
30	Obvodová rychlost [2)]	[m/s], [stopy/s]
31	Odpor	[ohm]
32	indukčnost	[mH]
33	Točivý moment [3)]	[Nm]
34	Konstanta točivého momentu [3)]	[Nm/A]
35	Zesílení proudového regulátoru	[V/A]
36	Zesílení regulátoru otáček [3)]	[Nm/(rad*s)]
37	Otáčky	[ot/min]
42	Výkon	[kW]
43	Proud, malý	[µA]
46	Točivý moment, malý [3)]	[µNm]
48	Promile	-
49	-	[Hz/s]
65	Průtok	[l/min]
66	Tlak	[bar]
67	Objem [3)]	[cm^3]
68	Dráhové zesílení [3)]	[mm/(V*min)]
69	Dráhové zesílení regulátoru síly	[N/V]
155	Stoupání závitu [3)]	[mm/ot], [palce/ot]
156	Změna stoupání závitu [3)]	[mm/ot / ot], [palce/ot / ot]
1) Fyzikální jednotka závisí na typu osy: lineární nebo kruhová osa		
2) Přepínání systému jednotek G70/G71(palce/metrické jednotky) Po přepnutí základního systému (MD10240 $MN_SCALING_SYSTEM_IS_METRIC) pomocí příkazu G70/G71 se v případě přístupu k systémovým a uživatelským proměnným souvisejícím s délkou za účelem čtení/zápisu neuskutečňuje **žádné** přepočítávání hodnot (skutečná hodnota, předdefinovaná hodnota a mezní hodnoty). G700/G710(palce/metrické jednotky) Po přepnutí základního systému (MD10240 $MN_SCALING_SYSTEM_IS_METRIC) pomocí příkazu G700/G710 se v případě přístupu k systémovým a uživatelským proměnným souvisejícím s délkou za účelem čtení/zápisu **uskutečňuje** přepočítávání hodnot (skutečná hodnota, předdefinovaná hodnota a mezní hodnoty).		
3) Proměnná **není** automaticky přepočítávána do momentálně nastaveného systému jednotek NC systému (palce/metrické jednotky). Za přepočítání nese plnou odpovědnost uživatel / výrobce stroje.		

Poznámka

Přeběh roviny v důsledku přepočítání formátu

Interně je pro ukládání všech uživatelských proměnných (GUD / PUD / LUD) s fyzikálními jednotkami vztahujícími se k délce používán metrický formát. Nadměrné používání proměnných tohoto druhu v hlavní větvi NCK, např. v synchronních akcích, může mít při přepnutí systému měřicích jednotek za následek přetečení výpočetního času roviny interpolátoru, alarm 4240.

Poznámka

Kompatibilita jednotek

Při používání proměnných (přiřazení, porovnávání, výpočty atd.) se neprovádí žádná kontrola, zda jsou jednotky podílející se na operaci kompatibilní. Za přepočítání, které je eventuálně nezbytné, nese plnou odpovědnost uživatel / výrobce stroje.

Viz také

Proměnné (Strana 389)

4.1.1.10 Atribut: Přístupová oprávnění (APR, APW, APRP, APWP, APRB, APWB)

Označení

Označení přístupového atributu AP... se skládá z těchto částí:

1. A: **A**ccess (přístup)
2. P: **P**rotection (ochrana)
3. R / W: **R**ead (čtení) / **W**rite (zápis)
4. P / B: **P**rogram / **B**TSS

Přístupová oprávnění / úrovně ochrany

Přístupová oprávnění odpovídají následujícím úrovním ochrany, které se zadávají při programování:

Přístupová oprávnění	Úroveň ochrany
Heslo systému	0
Heslo výrobce stroje	1
Heslo servisní služby	2
Heslo uživatele	3
3. poloha přepínače na klíč	4
2. poloha přepínače na klíč	5
1. poloha přepínače na klíč	6
0. poloha přepínače na klíč	7

Definice (DEF) uživatelských dat

Přístupová oprávnění (APR... / APW...) mohou být definována pro následující data:

- Globální uživatelská data (GUD)

Opětovná definice (REDEF) systémových a uživatelských dat

Přístupová oprávnění (APR... / APW...) mohou být opětovně definována pro následující data:

- Systémová data
  - Strojní parametry

    Poznámka

    Změna definice oprávnění ke čtení strojních parametrů

    Úroveň ochrany pro čtení strojních parametrů může být nastavována pomocí klíčového slova APR pouze společně pro výrobní program a BTSS.

    Klíčová slova APRP a APRB nejsou pro změnu definice oprávnění ke čtení podporována a mají za následek alarm 12490 s hlášením „Přístupová oprávnění APRP/APRB <úroveň ochrany> nebyla nastavena".
  - Nastavované parametry
  - Systémová proměnná
  - Data procesů
  - Data zásobníku
  - Parametry nástroje
- Uživatelská data
  - R-Parametry
  - Proměnné synchronních akcí ($AC_MARKER, $AC_PARAM, $AC_TIMER)
  - GUD synchronních akcí (SYG_xy[ ], kde x = R, I, B, A, C, S a y = S, M, U, 4, ..., 9)
  - Parametry EPS
  - Parametry nástroje OEM
  - Parametry zásobníku OEM
  - globální uživatelské proměnné (GUD)

    Poznámka

    Při opětovné definici mohou být přístupová oprávnění k určité proměnné volně nastavitelná v rozmezí mezi nejnižším stupněm ochrany 7 až po vlastní stupeň ochrany, např. 1 (výrobce stroje).

Opětovná definice (REDEF) příkazů NC jazyka

Oprávnění k přístupu, příp. ke spouštění (APX) mohou být opětovně definována pro následující příkazy NC jazyka:

- Příkazy G-funkce / podmínky dráhy (Strana 172)
- Předem definované funkce (Strana 1307)
- Vyvolávání předem definovaných podprogramů

- Příkaz DO u synchronních akcí
- Programový identifikátor cyklů
  Cyklus musí být uložen v adresáři cyklů a musí obsahovat příkaz PROC.

Přístupová oprávnění týkající se NC programů a cyklů (APRP, APWP)

Jednotlivá přístupová oprávnění mají pro přístup z NC programu, příp. z cyklu, následující účinky:

- APRP 0 / APWP 0
  - Při zpracovávání NC programu musí být zadáno heslo systému.
  - Cyklus musí být uložen v adresáři _N_CST_DIR (systém).
  - Pro adresář _N_CST_DIR musí být ve strojním parametru MD11160 $MN_ACCESS_EXEC_CST nastaveno oprávnění pro zpracovávání na "systém".
- APRP 1 / APWP 1, příp. APRP 2 / APWP 2
  - Při zpracovávání NC programu musí být zadáno heslo výrobce stroje, příp. servisní služby.
  - Cyklus musí být uložen v adresáři _N_CMA_DIR (výrobce stroje) nebo _N_CST_DIR.
  - Pro adresáře _N_CMA_DIR, příp. _N_CST_DIR, musí být ve strojních parametrech MD11161 $MN_ACCESS_EXEC_CMA, příp. MD11160 $MN_ACCESS_EXEC_CST nastaveno oprávnění pro zpracovávání minimálně na "výrobce stroje".
- APRP 3 / APWP 3
  - Při zpracovávání NC programu musí být zadáno heslo uživatele.
  - Cyklus musí být uložen v adresáři _N_CUS_DIR (uživatel), _N_CMA_DIR nebo _N_CST_DIR.
  - Pro adresáře _N_CUS_DIR, _N_CMA_DIR, příp. _N_CST_DIR, musí být ve strojních parametrech MD11162 $MN_ACCESS_EXEC_CUS, MD11161 $MN_ACCESS_EXEC_CMA, příp. MD11160 $MN_ACCESS_EXEC_CST nastaveno oprávnění pro zpracovávání minimálně na "uživatele".
- APRP 4...7 / APWP 4...7
  - Při zpracovávání NC programu musí být přepínač na klíč nastaven v poloze 3 ... 0.
  - Cyklus musí být uložen v adresáři _N_CUS_DIR, _N_CMA_DIR nebo _N_CST_DIR.
  - Pro adresáře _N_CUS_DIR, _N_CMA_DIR příp. _N_CST_DIR musí být ve strojních parametrech MD11162 $MN_ACCESS_EXEC_CUS, MD11161 $MN_ACCESS_EXEC_CMA, příp. MD11160 $MN_ACCESS_EXEC_CST nastaveno oprávnění pro zpracovávání minimálně na odpovídající polohu přepínače na klíč.

Přístupová oprávnění týkající se BTSS (APRB, APWB)

Přístupová oprávnění (APRB, APWB) omezují přístup k systémovým a uživatelským proměnným prostřednictvím BTSS stejnou měrou pro všechny systémové komponenty (HMI, PLC, externí počítač, služby EPS atd.).

Poznámka

Lokální přístupová oprávnění HMI

Při změnách přístupových oprávnění k systémovým datům je nutno dávat pozor, aby byly tyto změny v souladu s přístupovými oprávněními definovanými pomocí mechanismů HMI.

Přístupové atributy APR / APW

Z důvodů kompatibility jsou atributy APR a APW implicitně přenášeny do atributů APRP / APRB a APWP / APWB.

- APR x ⇒ APRP x APRB x
- APW y ⇒ APWP y APWB y

Přístupová oprávnění prostřednictvím souborů typu ACCESS

V případě, že jsou pro předávání přístupových oprávnění používány soubory typu ACCESS, smí být opětovné definice přístupových oprávnění pro systémová data, uživatelská data a příkazy NC jazyka prováděny výlučně v souborech typu ACCESS. Výjimku tvoří globální uživatelská data (GUD). V jejich případě musí být opětovná definice přístupových oprávnění i nadále uskutečňována v odpovídajících definičních souborech *_DEF.

Aby byla přístupová ochrana konzistentní, musí být odpovídajícím způsobem přizpůsobeny strojní parametry pro oprávnění ke spouštění funkcí a pro přístupovou ochranu pro příslušné adresáře.

Z toho vyplývají následující postupy:

1. Vytvořte potřebné definiční soubory:
   - _N_DEF_DIR/_N_SACCESS_DEF
   - _N_DEF_DIR/_N_MACCESS_DEF
   - _N_DEF_DIR/_N_UACCESS_DEF
2. Nastavte parametry oprávnění k zápisu pro tyto definiční soubory na hodnotu, která je zapotřebí pro opětovnou definici:
   - MD11170 $MN_ACCESS_WRITE_SACCESS = <úroveň ochrany>
   - MD11171 $MN_ACCESS_WRITE_MACCESS = <úroveň ochrany>
   - MD11172 $MN_ACCESS_WRITE_UACCESS = <úroveň ochrany>

3. Za účelem přístupu k chráněným prvkům z cyklů je nutné přizpůsobit oprávnění pro spouštění funkcí a pro zápis pro adresáře cyklů _N_CST_DIR, _N_CMA_DIR a _N_CST_DIR.
   Oprávnění ke spouštění funkcí
   - MD11160 $MN_ACCESS_EXEC_CST = <úroveň ochrany>
   - MD11161 $MN_ACCESS_EXEC_CMA = <úroveň ochrany>
   - MD11162 $MN_ACCESS_EXEC_CUS = <úroveň ochrany>

   Oprávnění k zápisu
   - MD11165 $MN_ACCESS_WRITE_CST = <úroveň ochrany>
   - MD11166 $MN_ACCESS_WRITE_CMA = <úroveň ochrany>
   - MD11167 MN_ACCESS_WRITE_CUS = <úroveň ochrany>

   Oprávnění pro spouštění funkcí musí být nastaveno minimálně na stejnou úroveň ochrany, jaká je nejvyšší úroveň ochrany používaného prvku.
   Oprávnění pro zápis musí být nastaveno minimálně na stejnou úroveň ochrany jako oprávnění pro spouštění funkcí.
4. Oprávnění k zápisu pro lokální adresáře cyklů HMI musí být nastaveno minimálně na stejnou úroveň ochrany, jaká je pro lokální adresáře cyklů v NC systému.

Vyvolávání podprogramů v souborech typu ACCESS

Kvůli dalšímu strukturování ochrany proti přístupu mohou být v souborech typu ACCESS také vyvolávány podprogramy (identifikace SPF nebo MPF). Tyto podprogramy přitom dědí oprávnění ke spouštění funkcí ze souboru typu ACCESS, z něhož byly vyvolány.

Poznámka

V souborech typu ACCESS mohou být znovu definována pouze přístupová oprávnění. Všechny ostatní atributy musí být i nadále naprogramovány, příp. znovu definovány v odpovídajících definičních souborech.

Viz také

Proměnné (Strana 389)

4.1.1.11 Přehled definovatelných a znovu definovatelných atributů

Následující tabulky ukazují, u kterých druhů dat mohou být definovány (`DEF`) a/nebo znovu definovány (`REDEF`) jednotlivé druhy atributů.

Systémová data

Druh dat	Počáteční hodnota	Mezní hodnoty	fyzikální jednotka	Přístupová oprávnění	Datová třída (pouze systém 828D)
Strojní parametry	---	---	---	REDEF	REDEF
Nastavované parametry	REDEF	---	---	REDEF	---

Druh dat	Počáteční hodnota	Mezní hodnoty	fyzikální jednotka	Přístupová oprávnění	Datová třída (pouze systém 828D)
Data typu FRAME	---	---	---	REDEF	---
Data procesů	---	---	---	REDEF	---
Kompenzace chyby stoupání vřetena (EEC)	---	---	---	REDEF	---
Kompenzace průvěsu (CEC)	---	---	---	REDEF	---
Kompenzace chyby kvadrantu (QEC)	---	---	---	REDEF	---
Data zásobníku	---	---	---	REDEF	---
Parametry nástroje	---	---	---	REDEF	---
Chráněné oblasti	---	---	---	REDEF	---
Orientovatelný držák nástroje	---	---	---	REDEF	---
Kinematické řetězce	---	---	---	REDEF	---
3D chráněné oblasti	---	---	---	REDEF	---
Ohraničení pracovního pole	---	---	---	REDEF	---

Uživatelská data

Druh dat	Počáteční hodnota	Mezní hodnoty	fyzikální jednotka	Přístupová oprávnění	Datová třída
R-parametry	REDEF	REDEF	REDEF	REDEF	---
Proměnné synchronních akcí ($AC_...)	REDEF	REDEF	REDEF	REDEF	---
GUD synchronních akcí (SYG_...)	REDEF	REDEF	REDEF	REDEF	---
Parametry EPS	REDEF	REDEF	REDEF	REDEF	---
Parametry nástroje OEM	REDEF	REDEF	REDEF	REDEF	---
Parametry zásobníku OEM	REDEF	REDEF	REDEF	REDEF	---
globální uživatelské proměnné (GUD)	DEF / REDEF	DEF	DEF	DEF / REDEF	DEF / REDEF
lokální uživatelské proměnné (PUD / LUD)	DEF	DEF	DEF	---	---

Viz také

Proměnné (Strana 389)

4.1.1.12 Definice a inicializace proměnných typu pole (DEF, SET, REP)

Uživatelská proměnná může být definována i jako 1- až maximálně 3-rozměrné pole (Array):

- 1-rozměrné: `DEF <datový typ> <název proměnné>[<n>]`
- 2-rozměrné: `DEF <datový typ> <název proměnné>[<n>,<m>]`
- 3-rozměrné: `DEF <datový typ> <název proměnné>[<n>,<m>,<o>]`

Poznámka

Uživatelské proměnné datového typu STRING mohou být definovány jako maximálně 2-rozměrné pole.

Datové typy

Uživatelské proměnné typu pole mohou být definovány pro následující datové typy: BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME

Přiřazování hodnoty prvku pole

Přiřazování hodnot prvkům pole je možné uskutečňovat v následujících okamžicích:

- při inicializaci pole (inicializační hodnoty)
- v průběhu zpracování programu

Přiřazování hodnot se přitom může provádět následujícími způsoby:

- explicitní zadání prvku pole
- explicitní zadání prvku pole jako prvku počátečního a uvedení seznamu hodnot (`SET`)
- explicitní zadání prvku pole jako prvku počátečního a uvedení hodnoty spolu s četností jejího opakování (`REP`)

Poznámka

Uživatelským proměnným datového typu FRAME není možné přiřadit žádné inicializační hodnoty.

Syntaxe (DEF)

```
DEF <datový typ> <název proměnné>[<n>,<m>,<o>]
DEF  STRING[<délka řetězce>] <název proměnné>[<n>,<m>]
```

Syntaxe (DEF...=SET...)

Použití seznamu hodnot:

- při definici:
  ```
  DEF <datový typ> <název proměnné>[<n>,<m>,<o>] =
  SET(<hodnota1>,<hodnota2>,...)
  ```
  stejný význam má:
  ```
  DEF <datový typ> <název proměnné>[<n>,<m>,<o>] =
  (<hodnota1>,<hodnota2>,...)
  ```

  Poznámka

  Při inicializaci prostřednictvím seznamu hodnot je použití příkazu `SET` nepovinné.

- při přiřazování hodnot:
  ```
  <název proměnné>[<n>,<m>,<o>] = SET(<hodnota1>,<hodnota2>,...)
  ```

Syntaxe(DEF...=REP...)

Použití hodnoty s opakováním

- při definici:
  ```
  DEF <datový typ> <název proměnné>[<n>,<m>,<o>] = REP(<hodnota>)
  DEF <datový typ> <název proměnné>[<n>,<m>,<o>] =
  REP(<hodnota>,<počet prvků pole>)
  ```
- při přiřazování hodnot:
  ```
  <název proměnné>[<n>,<m>,<o>] = REP(<hodnota>)
  <název proměnné>[<n>,<m>,<o>] = REP(<hodnota>,<počet prvků pole>)
  ```

Význam

`DEF`:	Příkaz pro definici proměnné	
`<datový typ>`:	Datový typ proměnné	
	Rozsah hodnot: • u systémové proměnné: BOOL, CHAR, INT, REAL, STRING, AXIS • u proměnných GUD nebo LUD: BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME	
`<délka řetězce>`:	Maximální počet znaků u datového typu STRING	
`<název proměnné>`:	Název proměnné	
`[<n>,<m>,<o>]`:	Velikost pole, příp. indexy pole	
`<n>`:	Velikost pole, příp. index pole pro 1. rozměr	
	Typ:	INT (u systémových proměnných také AXIS)
	Rozsah hodnot:	Max. velikost pole: 65535 Index pole: 0 ≤ n ≤ 65534
`<m>`:	Velikost pole, příp. index pole pro 2. rozměr	
	Typ:	INT (u systémových proměnných také AXIS)
	Rozsah hodnot:	Max. velikost pole: 65535 Index pole: 0 ≤ m ≤ 65534

<o>:	Velikost pole, příp. index pole pro 3. rozměr	
	Typ:	INT (u systémových proměnných také AXIS)
	Rozsah hodnot:	Max. velikost pole: 65535 Index pole: 0 ≤ o ≤ 65534
SET:	Přiřazení hodnot prostřednictvím seznamu	
(<hodnota1>,<hodnota2>,...):	Seznam hodnot	
REP:	Přiřazení hodnoty uvedené v poli <hodnota>	
<hodnota>:	Hodnota, která se má při inicializaci zapisovat pomocí příkazu REP do prvků pole.	
<počet prvků pole>:	Počet prvků pole, do kterých se má zapsat hodnota uvedená v poli <hodnota>. Pro zbývající prvky pole platí v závislosti na daném okamžiku: • Inicializace při definici pole: --> Do zbývajících prvků pole se dosazuje nula. • Přiřazení v průběhu zpracování programu: --> Momentální hodnoty prvků pole zůstávají zachovány. Pokud tento parametr není naprogramován, do prvků pole je dosazována hodnota uvedená v poli <hodnota>. Jestliže je tento parametr roven nule, platí v závislosti na daném okamžiku: • Inicializace při definici pole: --> Všem prvkům je dosazena nulová hodnota • Přiřazení v průběhu zpracování programu: --> Momentální hodnoty prvků pole zůstávají zachovány.	

Index pole

Implicitní posloupnost prvků pole, např. při přiřazování hodnot pomocí příkazů SET nebo REP, se odvíjí od postupného přičítání indexů pole zprava doleva.

Příklad: Inicializace 3-rozměrného pole s 24 prvky:

```
DEF INT FELD[2,3,4] = REP(1,24)
  FELD[0,0,0] = 1      1. Prvek pole
  FELD[0,0,1] = 1      2. Prvek pole
  FELD[0,0,2] = 1      3. Prvek pole
  FELD[0,0,3] = 1      4. Prvek pole
  ...
  FELD[0,1,0] = 1      5. Prvek pole
  FELD[0,1,1] = 1      6. Prvek pole
  ...
  FELD[0,2,3] = 1      12. Prvek pole
  FELD[1,0,0] = 1      13. Prvek pole
  FELD[1,0,1] = 1      14. Prvek pole
  ...
  FELD[1,2,3] = 1      24. Prvek pole
```

čemuž je ekvivalentní:

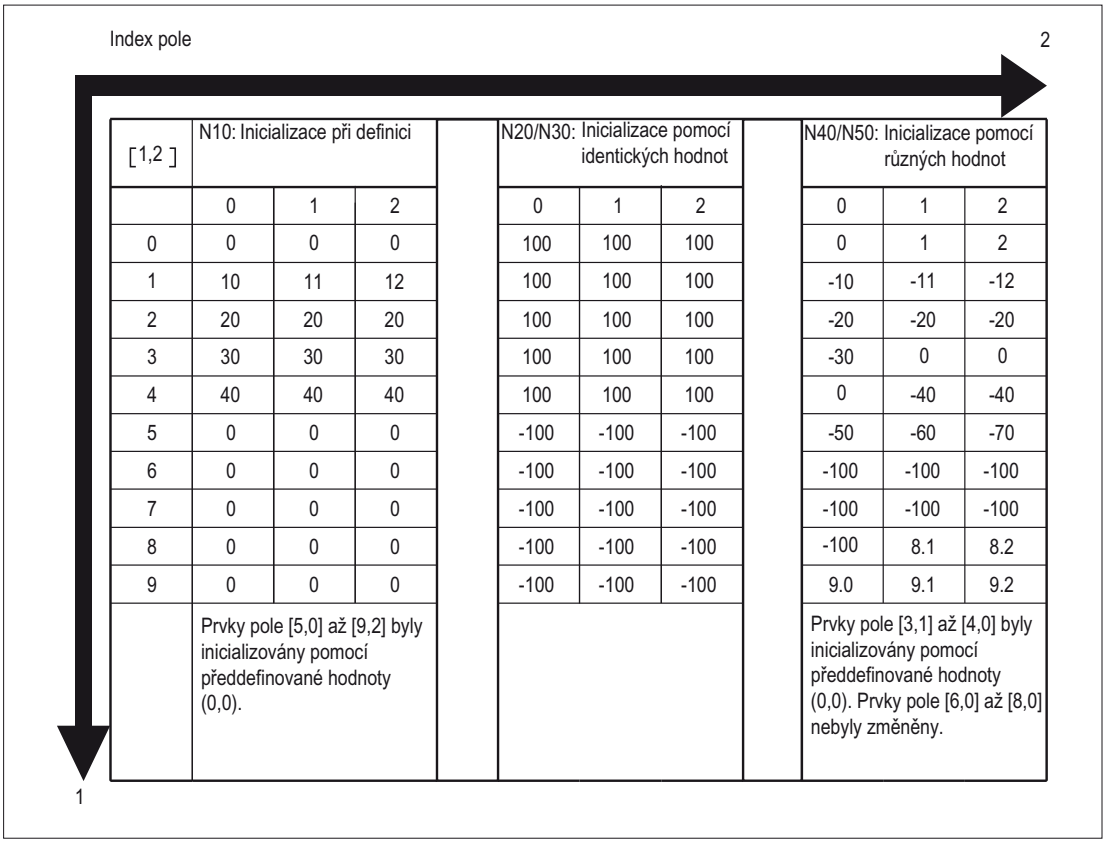
```
FOR n=0 TO 1
  FOR m=0 TO 2
     FOR o=0 TO 3
       FELD[n,m,o] = 1
     ENDFOR
  ENDFOR
ENDFOR
```

Příklad: Inicializace kompletního pole proměnných

Aktuální dosazení viz obrázek.

Programový kód

```
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: Inicializace při definici			N20/N30: Inicializace pomocí identických hodnot			N40/N50: Inicializace pomocí různých hodnot		
	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
	Prvky pole [5,0] až [9,2] byly inicializovány pomocí předdefinované hodnoty (0,0).						Prvky pole [3,1] až [4,0] byly inicializovány pomocí předdefinované hodnoty (0,0). Prvky pole [6,0] až [8,0] nebyly změněny.		

Viz také

Definice a inicializace proměnných typu pole (DEF, SET, REP) (Strana 419)

Proměnné (Strana 389)

Další informace (SET)

Inicializace při definici:

- Počínaje od 1. prvku bude inicializováno hodnotami ze seznamu tolik prvků pole, kolik prvků je naprogramováno v seznamu hodnot.
- Prvkům pole bez explicitně uvedené hodnoty v seznamu (mezery v seznamu hodnot) je dosazována 0.
- U proměnných datového typu AXIS jsou mezery v seznamu hodnot nepřípustné.
- Pokud seznam obsahuje více hodnot, než kolik je definováno prvků pole, aktivuje se alarm.

Přiřazování hodnot během zpracovávání programu

Při přiřazování hodnot v průběhu zpracování programu platí pravidla uvedená výše v rámci definic. Kromě toho existují ještě následující možnosti:

- Jako prvky v seznamu hodnot jsou přípustné také výrazy.
- Přiřazování hodnot začíná u naprogramovaného indexu pole. Takto se dají cíleně obsazovat hodnotami dílčí pole.

Příklad:

Programový kód	**Komentář**
`DEF INT FELD[5,5]`	`; definice pole`
`FELD[0,0]=SET(1,2,3,4,5)`	`; přiřazení hodnoty pro prvních 5 prvků pole [0,0] - [0,4]`
`FELD[0,0]=SET(1,2, , ,5)`	`; přiřazení hodnoty s mezerami pro prvních 5 prvků pole [0,0] - [0,4], prvky pole [0,2] a [0,3] = 0`

`FELD[2,3]=SET(VARIABLE,4*5.6)`	`; přiřazování hodnot s proměnnou a výrazem od indexu pole [2,3]:`
	`[2,3] = VARIABLE`
	`[2,4] = 4 * 5.6 = 22.4`

Další informace (REP)

Inicializace při definici:

- Všechny nebo libovolně zadaný počet prvků pole je inicializován uvedenou hodnotou (konstanta).
- Proměnné datového typu FRAME nemohou být inicializovány.

Příklad:

Programový kód	Komentář
`DEF REAL varName[10]=REP(3.5,4)`	`; definice pole a inicializace prvků pole [0] až [3] s hodnotou 3,5`

Přiřazování hodnot během zpracovávání programu

Při přiřazování hodnot v průběhu zpracování programu platí pravidla uvedená výše v rámci definic. Kromě toho existují ještě následující možnosti:

- Jako prvky v seznamu hodnot jsou přípustné také výrazy.
- Přiřazování hodnot začíná u naprogramovaného indexu pole. Takto se dají cíleně obsazovat hodnotami dílčí pole.

Příklady:

Programový kód	Komentář
`DEF REAL varName[10]`	`; definice pole`
`varName[5]=REP(4.5,3)`	`; prvky pole [5] až [7] = 4,5`
`R10=REP(2.4,3)`	`; R-parametr R10 až R12 = 2,4`
`DEF FRAME FRM[10]`	`; definice pole`
`FRM[5] = REP(CTRANS (X,5))`	`; prvky pole [5] až [9] = CTRANS(X,5)`

Viz také

Definice a inicializace proměnných typu pole (DEF, SET, REP) (Strana 419)

4.1.1.13 Datové typy

U NC systémů jsou k dispozici následující datové typy:

Datový typ	Význam	Rozsah hodnot
INT	celočíselné hodnoty se znaménkem	-2147483648 ... +2147483647
REAL	Reálné číslo (LONG REAL podle IEEE)	±(~2,2*10^{-308} ... ~1,8*10^{+308})
BOOL	Logická hodnota TRUE (1) a FALSE (0)	1, 0
CHAR	Znaky ASCII	Kód ASCII 0 ... 255
STRING	Řetězec znaků definované délky	Maximálně 200 znaků (žádné speciální znaky)
AXIS	Identifikátor osy/vřetena	Identifikátor kanálové osy
FRAME	Geometrické údaje pro statické transformace souřadného systému (posunutí, otočení, změna měřítka, zrcadlové převrácení)	---

Implicitní převody datového typu

Jsou možné následující převody datových typů a při přiřazováních a při přenášení parametrů se implicitně uskutečňují:

z ↓ / do →	REAL	INT	BOOL
REAL	x	o	&
INT	x	x	&
BOOL	x	x	x
x: možné bez omezení o: Je možná ztráta dat v důsledku překročení rozsahu hodnot ⇒ alarm; Zaokrouhlení: Hodnota za desetinnou čárkou ≥ 0,5 --> zaokrouhleno nahoru, hodnota za desetinnou čárkou < 0,5 --> zaokrouhleno dolů &: Hodnota ≠ 0 ⇒ TRUE, hodnota == 0 ⇒ FALSE			

Viz také

Proměnné (Strana 389)

4.1.1.14 Minimum, maximum a rozsah proměnných (MINVAL, MAXVAL, BOUND)

Pomocí příkazů `MINVAL` a `MAXVAL` mohou být mezi sebou porovnávány hodnoty dvou proměnných. Jako výsledek je na výstupu menší hodnota (v případě funkce `MINVAL`), příp. větší hodnota (v případě funkce `MAXVAL`).

Pomocí příkazu `BOUND` je možné zkontrolovat, zda se hodnota zkoušené proměnné nachází v rámci definovaného rozmezí hodnot.

Syntaxe

```
<menší hodnota>=MINVAL(<proměnná1>,<proměnná2>)
<větší hodnota>=MAXVAL(<proměnná1>,<proměnná2>)
<vracená hodnota>=<BOUND>(<minimum>,<maximum>,<zkoušená proměnná>)
```

Význam

`MINVAL`:	Zjišťuje **menší** hodnotu ze dvou proměnných (`<proměnná1>`, `<proměnná2>`)
`<menší hodnota>`:	Proměnná výsledku pro příkaz `MINVAL` Je jí dosazena hodnota menší proměnné.
`MAXVAL`:	Zjišťuje **větší** hodnotu ze dvou proměnných (`<proměnná1>`, `<proměnná2>`)
`<větší hodnota>`:	Proměnná výsledku pro příkaz `MAXVAL` Je jí dosazena hodnota větší proměnné.
`BOUND`:	Zkontroluje, zda proměnná (`<zkoušená proměnná>`) leží v rámci definovaného rozsahu hodnot.
`<minimum>`:	Proměnná, která definuje minimální hodnotu rozsahu hodnot.

<maximum>:	Proměnná, která definuje maximální hodnotu rozsahu hodnot.
<vracená hodnota>:	Proměnná výsledku pro příkaz BOUND Pokud hodnota zkoušené proměnné leží v rámci definovaného rozsahu hodnot, potom se do proměnné výsledku dosadí hodnota zkoušené proměnné. Pokud je hodnota zkoušené proměnné větší než maximální hodnota, potom se do proměnné výsledku dosadí maximální hodnota definovaného rozsahu. Pokud je hodnota zkoušené proměnné menší než minimální hodnota, potom se do proměnné výsledku dosadí minimální hodnota definovaného rozsahu.

Poznámka

Funkce MINVAL, MAXVAL a BOUND mohou být naprogramovány i v synchronních akcích.

Poznámka

Chování v případě rovnosti

V případě rovnosti budou funkce MINVAL/MAXVAL vracet tuto stejnou hodnotu. V případě funkce BOUND bude výsledkem opět hodnota zkoušené proměnné.

Příklad:

```
Programový kód                                    Komentář
DEF REAL rVar1=10.5, rVar2=33.7, rVar3, rVar4,  rVar5, rValMin, rValMax, rRetVar
rValMin=MINVAL(rVar1,rVar2)                       ; proměnná rValMin se nastaví na hodnotu 10,5
rValMax=MAXVAL(rVar1,rVar2)                       ; proměnná rValMax se nastaví na hodnotu 33,7
rVar3=19.7
rRetVar=BOUND(rVar1,rVar2,rVar3)                  ; proměnná rVar3 leží v rámci hranic, rRetVar se nastaví na hodnotu 19,7
rVar3=1.8
rRetVar=BOUND(rVar1,rVar2,rVar3)                  ; proměnná rVar3 leží pod hranicí minima, rRetVar se nastaví na hodnotu 10,5
rVar3=45.2
rRetVar=BOUND(rVar1,rVar2,rVar3)                  ; proměnná rVar3 leží nad hranicí maxima, rRetVar se nastaví na hodnotu 33,7
```

4.1.1.15 Kontrola existence proměnné (ISVAR)

Prostřednictvím předefinované funkce ISVAR je možné zkontrolovat, zdali je systémová/uživatelská proměnná (např. strojní parametr, nastavovaný parametr, systémová proměnná, všeobecná proměnná, jako např. GUD) v NC známa.

Proměnná

Proměnná, která je předmětem dotazu, má následující strukturu:

Bezrozměrná proměnná:	<proměnná>
Jednorozměrná proměnná bez indexu pole	<proměnná>[ ]
Jednorozměrná proměnná s indexem pole	<proměnná>[<n>]
Dvourozměrná proměnná bez indexu pole	<proměnná>[,]
Dvourozměrná proměnná s indexy pole n a m:	<proměnná>[<n>,<m>]

Syntaxe

```
<výsledek>=ISVAR(<proměnná>[<n>,<m>])
```

Význam

<výsledek>:	Výsledná hodnota		
	Datový typ:	BOOL	
	Rozsah hodnot:	1	Proměnná existuje
		0	Proměnná je neznámá
ISVAR:	Kontrola, zda je uvedená systémová/uživatelská proměnná v NC systému známa.		
<proměnná>:	Název systémové/uživatelské proměnné		
	Datový typ:	STRING	
<n>:	Index prvního rozměru (**není nutné**)		
	Datový typ:	INT	
<m>:	Index druhého rozměru (**není nutné**)		
	Datový typ:	INT	

V závislosti na předávaném parametru jsou prováděny následující zkoušky:

- Je název znám
- Je proměnná typu pole
- Je to jedno- nebo dvourozměrné pole
- Je příslušný index pole v rozsahu přípustných hodnot

Když odpovědi na všechny tyto otázky jsou kladné, je výsledkem hodnota TRUE (1).

Jestliže je odpověď na jednu z těchto otázek záporná nebo pokud se vyskytla syntaktická chyba, je výsledkem hodnota FALSE (0).

Příklady

Programový kód	**Komentář**
DEF INT VAR1	
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("VAR1")	; IS_VAR je v tomto případě TRUE

Programový kód	Komentář
DEF REAL VARARRAY[10,10]	
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("VARARRAY[,]")	; IS_VAR je v tomto případě TRUE, je to dvourozměrné pole
N20 IS_VAR=ISVAR("VARARRAY")	; IS_VAR je TRUE, proměnná existuje
N30 IS_VAR=ISVAR("VARARRAY[8,11]")	; IS_VAR je FALSE, index pole je nepřípustný
N40 IS_VAR=ISVAR("VARARRAY[8,8")	; IS_VAR je FALSE, chybí znak "]" (syntaktická chyba)
N50 IS_VAR=ISVAR("VARARRAY[,8]")	; IS_VAR je TRUE, index pole je přípustný
N60 IS_VAR=ISVAR("VARARRAY[8,]")	; IS_VAR je TRUE, index pole je přípustný

Programový kód	Komentář
DEF BOOL IS_VAR=FALSE	
N100 IS_VAR=ISVAR("$MC_GCODE_RESET_VALUES[1]"	; předávaný parametr je strojní parametr, IS_VAT je TRUE

Programový kód	Komentář
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("$P_EP")	; IS_VAR je v tomto případě TRUE
N20 IS_VAR=ISVAR("$P_EP[X]")	; IS_VAR je v tomto případě TRUE

4.1.1.16 Načítání hodnot atributu/datového typu (GETVARPHU, GETVARAP, GETVARLIM, GETVARDIM, GETVARDFT, GETVARTYP)

Pomocí předem definovaných funkcí GETVARPHU, GETVARAP, GETVARLIM, GETVARDIM a GETVARDFT je možné načítat hodnoty atributů systémových/uživatelských proměnných, pomocí funkce GETVARTYP se zjišťuje datový typ systémové/uživatelské proměnné.

Načítání fyzikálních jednotek

Syntaxe:
<výsledek>=GETVARPHU(<název>)

Význam:

<výsledek>:	Číselná hodnota fyzikální jednotky		
	Datový typ:	INT	
	Rozsah hodnot:	viz tabulka v kapitole "Atribut: Fyzikální jednotka (PHU) (Strana 411)"	
		V případě chyby:	
		- 2	Zadaný název proměnné není přiřazen žádnému systémovému parametru a žádné uživatelské proměnné.
GETVARPHU:	Čtení fyzikálních jednotek systémové/uživatelské proměnné		

<název>:	Název systémové/uživatelské proměnné	
	Datový typ:	STRING

Příklad:

NC obsahuje následující proměnné GUD:

DEF CHAN REAL PHU 42 LLI 0 ULI 10000 electric

Programový kód	**Komentář**
DEF INT result=0	
result=GETVARPHU("electric")	; zjišťování fyzikálních jednotek proměnných typu GUD
IF (result < 0) GOTOF error	

Jako výsledek systém vrací hodnotu 42. To odpovídá fyzikální jednotce [kW].

Poznámka

Pomocí funkce GETVARPHU se dá např. zkontrolovat, zda při přiřazování hodnot proměnným a = b mají obě proměnné očekávané fyzikální jednotky.

Načítání přístupových oprávnění

Syntaxe:
<výsledek>=GETVARAP(<název>,<přístup>)

Význam:

<výsledek>:	Úroveň ochrany pro zadaný parametr <přístup>		
	Datový typ:	INT	
	Rozsah hodnot:	0 ... 7	Viz "Atribut: Přístupová oprávnění (APR, APW, APRP, APWP, APRB, APWB) (Strana 413)".
		V případě chyby:	
		- 1	zápis není možný (má smysl jen v případě, že pro druhy přístupu je nastaveno „WP“ a „WB“)
		- 2	Zadaný název proměnné není přiřazen žádnému systémovému parametru a žádné uživatelské proměnné.
		- 3	nesprávná hodnota pro parametr <přístup>
GETVARAP:	Čtení přístupových oprávnění k systémové/uživatelské proměnné		
<název>:	Název systémové/uživatelské proměnné		
	Datový typ:	STRING	

<přístup>:	Druh přístupu		
	Datový typ:	STRING	
	Rozsah hodnot:	"RP"	Čtení prostřednictvím výrobního programu
		"WP"	Zápis prostřednictvím výrobního programu
		"RB"	Čtení prostřednictvím BTSS
		"WB"	Zápis prostřednictvím BTSS

Příklad:

Programový kód	**Komentář**
DEF INT result=0	
result=GETVARAP("$TC_MAP8","WB")	; zjištění ochrany proti přístupu pro systémový parametr "Magazinposition" (Poloha zásobníku), pokud jde o zápis pomocí BTSS
IF (result < 0) GOTOF error	

Jako výsledek systém vrací hodnotu 7. To odpovídá poloze 0 přepínače na klíč (= žádná ochrana proti přístupu).

Poznámka

Pomocí funkce GETVARAP je možné realizovat například kontrolní program, který zkouší přístupová oprávnění očekávaná aplikací.

Čtení mezních hodnot

Syntaxe:
<Status>=GETVARLIM(<název>,<mezní hodnota>,<výsledek>)

Význam:

<Status>:	Stav funkce		
	Datový typ:	INT	
	Rozsah hodnot:	1	o. k.
		-1	není definována žádná mezní hodnota (u proměnných typu AXIS, STRING, FRAME)
		-2	Zadaný název proměnné není přiřazen žádnému systémovému parametru a žádné uživatelské proměnné.
		-3	nesprávná hodnota pro parametr <mezní hodnota>
GETVARLIM:	Čtení dolní/horní mezní hodnoty systémové/uživatelské proměnné		
<název>:	Název systémové/uživatelské proměnné		
	Datový typ:	STRING	
<mezní hodnota>:	Udává, která mezní hodnota má být načtena.		
	Datový typ:	CHAR	
	Rozsah hodnot:	"L"	dolní mezní hodnota
		"U"	horní mezní hodnota

<výsledek>:	Výsledná mezní hodnota	
	Datový typ:	VAR REAL

Příklad:

Programový kód	**Komentář**
DEF INT state=0	
DEF REAL result=0	
state=GETVARLIM("$MA_MAX_AX_VELO","L",result)	; zjištění dolní mezní hodnoty pro parametr MD32000 $MA_MAX_AX_VELO
IF (result < 0) GOTOF error	

Čtení hodnoty rozměru

Syntaxe:
<výsledek>=GETVARDIM(<název>,<index>)

Význam:

<výsledek>:	Hodnota rozměru, který je stanoven parametrem <index>		
	Datový typ:	INT	
GETVARDIM:	Čtení hodnoty 1., 2. nebo 3. rozměru systémové/uživatelské proměnné typu pole		
<název>:	Název systémové/uživatelské proměnné		
	Datový typ:	STRING	
<index>:	Index pole		
	Datový typ:	INT	
	Rozsah hodnot:	1 ... 3	
		1	Index pro 1. rozměr pole
		2	Index pro 2. rozměr pole
		3	Index pro 3. rozměr pole

Příklad:

Programový kód	**Komentář**
N5 DEF REAL myReal[5,4]	
N10 R1=GETVARDIM("myReal",1)	; Zjištění hodnoty 1. rozměru pole.
	; Výsledek: R1 = 5
N15 R2=GETVARDIM("myReal",2)	; Zjištění hodnoty 2. rozměru pole.
	; Výsledek: R2 = 4

Načtení standardní hodnoty

Syntaxe:
<Status>=GETVARDFT(<název>,<výsledek>[,<index_1>,<index_2>,<index_3>])

Význam:

<Status>:	Stav funkce		
	Datový typ:	INT	
	Rozsah hodnot:	1	o. k.
		-1	Není k dispozici žádná standardní hodnota (např. protože definice datového typu proměnné pro výsledek neodpovídá datovému typu systémové/uživatelské proměnné)
		-2	Zadaný název proměnné není přiřazen žádnému systémovému parametru a žádné uživatelské proměnné.
		-3	Nesprávná hodnota pro parametr <index_1>, rozměr je menší než jedna (⇒ žádné pole, nýbrž skalární proměnná)
		-4	nesprávná hodnota pro parametr <index_2>
		-5	nesprávná hodnota pro parametr <index_3>
GETVARDFT:	Čtení standardní hodnoty systémové/uživatelské proměnné		
<název>:	Název systémové/uživatelské proměnné		
	Datový typ:	STRING	
<výsledek>:	Výsledná standardní hodnota		
	Datový typ:	VAR REAL (při čtení standardní hodnoty proměnných typů INT, REAL, BOOL, AXIS)	
		VAR STRING (při čtení standardní hodnoty proměnných typů STRING a CHAR)	
		VAR FRAME (při čtení standardní hodnoty proměnných typu FRAME)	
<index_1>:	Index pro 1. rozměr pole (nepovinné)		
	Datový typ:	INT	
	Pokud není naprogramováno, znamená = 0		
<index_2>:	Index pro 2. rozměr pole (nepovinné)		
	Datový typ:	INT	
	Pokud není naprogramováno, znamená = 0		
<index_3>:	Index pro 3. rozměr pole (nepovinné)		
	Datový typ:	INT	
	Pokud není naprogramováno, znamená = 0		

Příklad:

Programový kód	**Komentář**
`DEF INT state=0`	
`DEF REAL resultR=0`	`; proměnná pro převzetí standardní hodnoty typu INT, REAL, BOOL, AXIS`
`DEF FRAME resultF=0`	`; proměnná pro převzetí standardní hodnoty typu FRAME`
`IF (GETVARTYP("$MA_MAX_AX_VELO") <> 4) GOTOF error`	
`state=GETVARDFT("$MA_MAX_AX_VELO",resultR, AXTOINT(X))`	`; zjištění standardní hodnoty osy "X".`

Programový kód	Komentář
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)	

Načtení datového typu

Syntaxe:
<výsledek>=GETVARTYP(<název>)

Význam:

<výsledek>:	Datový typ uvedené systémové/uživatelské proměnné		
	Datový typ:	INT	
	Rozsah hodnot:	1	= BOOL
		2	= CHAR
		3	= INT
		4	= REAL
		5	= STRING
		6	= AXIS
		7	= FRAME
		V případě chyby:	
		< 0	Zadaný název proměnné není přiřazen žádnému systémovému parametru a žádné uživatelské proměnné.
GETVARTYP:	Čtení datového typu systémové/uživatelské proměnné		
<název>:	Název systémové/uživatelské proměnné		
	Datový typ:	STRING	

Příklad:

Programový kód	Komentář
DEF INT result=0	
DEF STRING name="R"	
result=GETVARTYP(name)	; zjištění typu R-parametru
IF (result < 0) GOTOF error	

Jako výsledek systém vrací hodnotu 4. To odpovídá datovému typu REAL.

4.1.1.17 Možné převádění typů

Konstantní číselná hodnota, proměnná nebo výraz, který je přiřazen proměnné, musí být kompatibilní s tímto typem proměnné. Je-li tomu tak, je při přiřazení typ automaticky převáděn.

Možné konverze typu

na	REAL	INT	BOOL	CHAR	STRING	AXIS	FRAME
z							
REAL	ano	ano*	ano[1])	ano*	–	–	–
INT	ano	ano	ano[1])	ano[2])	–	–	–
BOOL	ano	ano	ano	ano	ano	–	–
CHAR	ano	ano	ano[1])	ano	ano	–	–
STRING	–	–	ano[4])	ano[3])	ano	–	–
AXIS	–	–	–	–	–	ano	–
FRAME	–	–	–	–	–	–	ano

Vysvětlení

* Při konverzi z typu REAL na typ INT se desetinná část >= 0,5 zaokrouhluje nahoru, jinak dolů (srov. s funkcí ROUND).

[1]) Hodnotě 0 odpovídá TRUE, hodnotě == 0 odpovídá FALSE

[2]) Když hodnota leží v přípustném intervalu čísel

[3]) Když jen jeden znak

[4]) Délka řetězce 0 = FALSE, jinak TRUE

Poznámka

Je-li při konverzi některá hodnota větší než cílová oblast, bude to mít za následek chybové hlášení.

Objeví-li se v jednom výrazu smíšené typy, tak je automaticky provedeno přizpůsobení typu. Převody datových typů jsou možné i v synchronních akcích, viz kapitola "Pohybové synchronní akce, implicitní konverze typů".

4.1.2 Nepřímé programování

4.1.2.1 Nepřímé programování adres

Při nepřímém programování adres (Strana 1251) je rozšířená adresa (<Index>) nahrazována proměnnou vhodného typu.

Poznámka

Nepřímé programování není možné u těchto adres:

- N (číslo bloku)
- L (podprogram)
- Nastavitelné adresy
  (např. X[1] namísto X1 není přípustné)

Syntaxe

```
<ADRESA>[<Index>]
```

Význam

<ADRESA>[...]:	Pevná adresa s rozšířením (index)
<Index>:	Proměnná, např. pro číslo vřetena, osy, ...

Příklady

Příklad 1: Nepřímé programování čísla vřetena

Přímé programování:

Programový kód	Komentář
S1=300	; otáčky 300 ot/min pro vřeteno s číslem 1

Nepřímé programování:

Programový kód	Komentář
DEF INT SPINU=1	; definice proměnné typu INT a přiřazení hodnoty
S[SPINU]=300	; otáčky 300 ot/min pro vřeteno, jehož číslo je uloženo v proměnné SPINU (v tomto příkladu je to vřeteno s číslem 1)

Příklad 2: Nepřímé programování osy

Přímé programování:

Programový kód	Komentář
FA[U]=300	; posuv 300 pro osu "U"

Nepřímé programování:

Programový kód	Komentář
DEF AXIS AXVAR2=U	; definice proměnné typu AXIS a přiřazení hodnoty
FA[AXVAR2]=300	; posuv 300 pro osu, jejíž adresový název je uložen v proměnné s názvem AXVAR2

Příklad 3: Nepřímé programování osy

Přímé programování:

Programový kód	Komentář
$AA_MM[X]	; načtení hodnoty změřené měřicí sondou (MCS) pro osu "X"

Nepřímé programování:

Programový kód	Komentář
DEF AXIS AXVAR3=X	; definice proměnné typu AXIS a přiřazení hodnoty
$AA_MM[AXVAR3]	; načtení hodnoty změřené měřicí sondou (MCS) pro osu, jejíž název je uložen v proměnné AXVAR3

Příklad 4: Nepřímé programování osy

Přímé programování:

Programový kód
X1=100 X2=200

Nepřímé programování:

Programový kód	Komentář
DEF AXIS AXVAR1 AXVAR2	; definice dvou proměnných typu AXIS
AXVAR1=(X1) AXVAR2=(X2)	; přiřazení názvů os
AX[AXVAR1]=100 AX[AXVAR2]=200	; pohyb osami, jejichž adresové názvy jsou uloženy v proměnných s názvy AXVAR1 a AXVAR2

Příklad 5: Nepřímé programování osy

Přímé programování:

Programový kód
G2 X100 I20

Nepřímé programování:

Programový kód	Komentář
DEF AXIS AXVAR1=X	; definice proměnné typu AXIS a přiřazení hodnoty
G2 X100 IP[AXVAR1]=20	; nepřímé programování zadání středu pro osu, jejíž adresovací název je uložen v proměnné s názvem AXVAR1

Příklad 6: Nepřímé programování prvků pole

Přímé programování:

Programový kód	Komentář
DEF INT FELD1[4,5]	; definice pole 1

Nepřímé programování:

Programový kód	Komentář
DEFINE DIM1 AS 4	; v případě rozměrů pole musí být velikosti pole uvedeny jako pevné hodnoty
DEFINE DIM2 AS 5	
DEF INT FELD[DIM1,DIM2]	
FELD[DIM1-1,DIM2-1]=5	

Příklad 7: Nepřímé volání podprogramu

Programový kód	Komentář
CALL "L" << R10	; Volání programu, jehož číslo je uloženo v R10 (skládání řetězců).

4.1.2.2 Nepřímé programování příkazů G-funkcí

Nepřímé programování příkazů G-funkcí (Strana 1263) umožňuje efektivní programování cyklů.

Syntaxe

```
G[<skupina>]=<číslo>
```

Význam

`G[...]`:	G-příkaz s rozšířením (index)	
`<skupina>`:	Indexový parametr: G-skupina	
	Typ:	INT
`<číslo>`:	Proměnná pro číslo příkazu G-funkce	
	Typ:	INT nebo REAL

Poznámka

Nepřímo mohou být zpravidla programovány pouze příkazy G-funkcí, které nejsou syntakticky určující.

Ze syntakticky určujících příkazů G-funkcí jsou možné pouze kódy z 1. skupiny G-funkcí. Příkazy G-funkcí, které určují syntaxi, ze skupin G-funkcí 2, 3 a 4 možné nejsou.

Poznámka

V nepřímém programování příkazů G-funkcí nejsou dovoleny žádné aritmetické funkce. Nutný výpočet čísla příkazu G-funkce musí proběhnout na samostatném řádku výrobního programu před nepřímým programováním příkazu G-funkce.

Příklady

Příklad 1: Nastavitelné posunutí počátku (skupina G-funkcí č. 8)

```
Programový kód                          Komentář
N1010 DEF INT INT_VAR
N1020 INT_VAR=2
...
N1090 G[8]=INT_VAR G1 X0 Y0             ; G54
N1100 INT_VAR=INT_VAR+1                 ; výpočet příkazu G-funkce
N1110 G[8]=INT_VAR G1 X0 Y0             ; G55
```

Příklad 2: Volba roviny (skupina G-funkcí č. 6)

```
Programový kód                          Komentář
N2010 R10=$P_GG[6]                      ; načtení aktivního příkazu G-funkce ze
                                        skupiny 6
...
```

Programový kód	**Komentář**
N2090 G[6]=R10	

4.1.2.3 Nepřímé programování atributů polohy (GP)

Atributy polohy, jako např. inkrementální nebo absolutní programování poloh os, mohou být ve spojení s klíčovým slovem GP programovány nepřímo jako proměnné.

Aplikace

Nepřímé programování atributů polohy nachází uplatnění v **nahrazovacích cyklech**, protože zde oproti programování atributů polohy jako klíčových slov (např. IC, AC, ...) existuje následující výhoda:

Díky nepřímému programování, které využívá proměnné, není zapotřebí **žádný** příkaz CASE, který by zajišťoval větvení na všechny možné atributy polohy.

Syntaxe

```
<PŘÍKAZ POLOHOVÁNÍ>[<osa/vřeteno>]=
GP(<poloha>,<atribut polohy)
<osa/vřeteno>=GP(<poloha>,<atribut polohy>)
```

Význam

<PŘÍKAZ POLOHOVÁNÍ>[]:	Spolu s klíčovým slovem GP mohou být naprogramovány následující polohovací příkazy: POS, POSA, SPOS, SPOSA Kromě toho je možné zadat: • všechny identifikátory os/vřeten vyskytující se v kanálu: <osa/vřeteno> • proměnný identifikátor osy/vřetena AX
<osa/vřeteno>:	Osa/vřeteno, jehož poloha má být nastavena
GP():	Klíčové slovo pro polohování
<pozice>:	Parametr 1 Pozice osy/vřetena jako konstanta nebo jako proměnná
<atribut polohy>:	Parametr 2 Atribut polohy (např. režim najíždění na pozici) jako proměnná (např. $P_SUB_SPOSMODE) nebo jako klíčové slovo (IC, AC, ...)

Hodnoty předávané proměnnými mají následující význam:

Hodnota	**Význam**	**Přípustné pro:**
0	Žádná změna atributu polohy	
1	AC	POS, POSA, SPOS, SPOSA, AX, adresa osy
2	IC	POS, POSA, SPOS, SPOSA, AX, adresa osy
3	DC	POS, POSA, SPOS, SPOSA, AX, adresa osy

Hodnota	Význam	Přípustné pro:
4	ACP	POS, POSA, SPOS, SPOSA, AX, adresa osy
5	ACN	POS, POSA, SPOS, SPOSA, AX, adresa osy
6	OC	-
7	PC	-
8	DAC	POS, POSA, AX, adresa osy
9	DIC	POS, POSA, AX, adresa osy
10	RAC	POS, POSA, AX, adresa osy
11	RIC	POS, POSA, AX, adresa osy
12	CAC	POS, POSA
13	CIC	POS, POSA
14	CDC	POS, POSA
15	CACP	POS, POSA
16	CACN	POS, POSA

Příklad:

Jestliže je aktivní synchronní vazba mezi řídícím vřetenem S1 a vlečným vřetenem S2, příkazem `SPOS` v hlavním programu se vyvolává následující nahrazovací cyklus, který polohování vřeten zajišťuje.

Polohování se uskutečňuje prostřednictvím příkazu v bloku N2230:
`SPOS[1]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE)`
`SPOS[2]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE)`

Poloha, na kterou se má najet, se načítá ze systémové proměnné $P_SUB_SPOSIT, režim najíždění na pozici se načítá ze systémové proměnné $P_SUB_SPOSMODE.

```
Programový kód                                  Komentář
N1000 PROC LANG_SUB DISPLOF SBLOF
...
N2100 IF($P_SUB_AXFCT==2)
N2110                                           ; nahrazení příkazu SPOS / SPOSA / M19 v
                                                případě aktivní synchronní vazby
N2185 DELAYFSTON                                ; začátek oblasti zastavení-prodleva
N2190 COUPOF(S2,S1)                             ; deaktivování synchronní vazby mezi vř-
                                                eteny
N2200                                           ; polohování řídícího a vlečného vřetena
N2210 IF($P_SUB_SPOS==TRUE) OR
($P_SUB_SPOSA==TRUE)
N2220                                           ; polohování vřetena s příkazem SPOS:
N2230    SPOS[1]=GP($P_SUB_SPOSIT,
$P_SUB_SPOSMODE)
         SPOS[2]=GP($P_SUB_SPOSIT,
$P_SUB_SPOSMODE)
N2250 ELSE
N2260                                           ; polohování vřetena s příkazem M19:
```

Programový kód	**Komentář**
N2270 M1=19 M2=19	; polohování řídícího a vlečného vřetena
N2280 ENDIF	
N2285 DELAYFSTOF	; konec oblasti zastavení-prodleva
N2290 COUPON(S2,S1)	; aktivování synchronní vazby mezi vřeteny
N2410 ELSE	
N2420	; dotaz na další nahrazování
...	
N3300 ENDIF	
...	
N9999 RET	

Okrajové podmínky

V synchronních akcích není nepřímé programování atributů polohy možné.

4.1.2.4 Nepřímé programování řádků výrobního programu (EXECSTRING)

Pomocí příkazu výrobního programu EXECSTRING je možné spustit zpracování dříve sestavené proměnné typu STRING, jako by se jednalo o řádek výrobního programu.

Syntaxe

Příkaz EXECSTRING musí být naprogramován na samostatném řádku výrobního programu.
EXECSTRING (<proměnná typu string>)

Význam

EXECSTRING:	Příkaz pro zpracování proměnné typu SRTING, jako kdyby šlo o řádek výrobního programu
<proměnná typu String>:	Proměnná typu STRING, která obsahuje řádek výrobního programu, který může být samostatně zpracován.

Poznámka

Pomocí příkazu EXECSTRING mohou být předávány veškeré konstrukce výrobního programu, které mohou být naprogramovány **v programové části výrobního programu**, s výjimkou řídících struktur (Strana 476). Vyloučeny jsou tedy příkazy PROC a DEF, jakož i obecně použití souborů INI a DEF.

Příklad:

Programový kód	**Komentář**
N100 DEF STRING[100] MY_BLOCK	; definice proměnné typu String pro sestavení zpracovatelného řádku výrobního programu
N110 DEF STRING[10] MFCT1="M7"	

Programový kód	**Komentář**
...	
N200 EXECSTRING(MFCT1 << "M4711")	; zpracování řádku výrobního programu "M7 M4711"
...	
N300 R10=1	
N310 MY_BLOCK="M3"	
N320 IF(R10)	
N330 MY_BLOCK = MY_BLOCK << MFCT1	
N340 ENDIF	
N350 EXECSTRING(MY_BLOCK)	; zpracování řádku výrobního programu "M3 M7"

4.1.3 Operace

4.1.3.1 Matematické funkce

Operátory / matematické funkce	**Význam**
+	sečítání
–	odečítání
*	násobení
/ [1)]	Dělení [1)]
DIV [1)]	Celočíselné dělení [1)]
MOD [1)]	Dělení Modulo (výsledkem je zbytek po celočíselném dělení) [1)]
:	Řetězcový operátor pro proměnné typu FRAME
SIN()	sinus
COS()	kosinus
TAN()	tangens
ASIN()	arkus sinus
ACOS()	arkus kosinus
ATAN2(,) [1)]	Arkus tangens na druhou [1)]
SQRT()	druhá odmocnina
ABS()	absolutní hodnota
POT()	2. mocnina (čtverec)
TRUNC()	celočíselná část Nastavení přesnosti při příkazech porovnávání pomocí funkce TRUNC (viz "Korekce přesnosti při příkazech porovnávání (TRUNC) (Strana 446)")
ROUND()	zaokrouhlování na celá čísla
LN()	přirozený logaritmus
EXP()	exponenciální funkce

`MINVAL()`	menší hodnota ze dvou proměnných (viz "Minimum, maximum a rozsah proměnných (MINVAL, MAXVAL, BOUND) (Strana 425)")
`MAXVAL()`	větší hodnota ze dvou proměnných (viz "Minimum, maximum a rozsah proměnných (MINVAL, MAXVAL, BOUND) (Strana 425)")
`BOUND()`	Hodnota proměnné, která leží v definovaném intervalu hodnot (viz "Minimum, maximum a rozsah proměnných (MINVAL, MAXVAL, BOUND) (Strana 425)")
`CTRANS()`	Posunutí
`CROT()`	Otočení
`CSCALE()`	změna měřítka
`CMIRROR()`	zrcadlové převrácení
1) Viz odstavec „Příklady“	

Programování

U matematických operací platí obvyklý matematický způsob zápisu. Priority při zpracování jsou určovány pomocí kulatých závorek. Pro trigonometrické a k nim inverzní funkce platí udání stupňů (pravý úhel = 90°).

Příklady

Děelení: /

(Typ REAL) = (Typ INT nebo Typ REAL) / (Typ INT nebo Typ REAL);

Příklad: 3 / 4 = 0,75

Celočíselné dělení: DIV

(Typ INT) = (Typ INT nebo REAL) / (Typ INT nebo REAL);

Příklad: 7 DIV 4.1 = 1

Dělení Modulo (výsledkem je zbytek po celočíselném dělení): MOD

(Typ REAL) = (Typ INT nebo REAL) MOD (Typ INT nebo REAL);

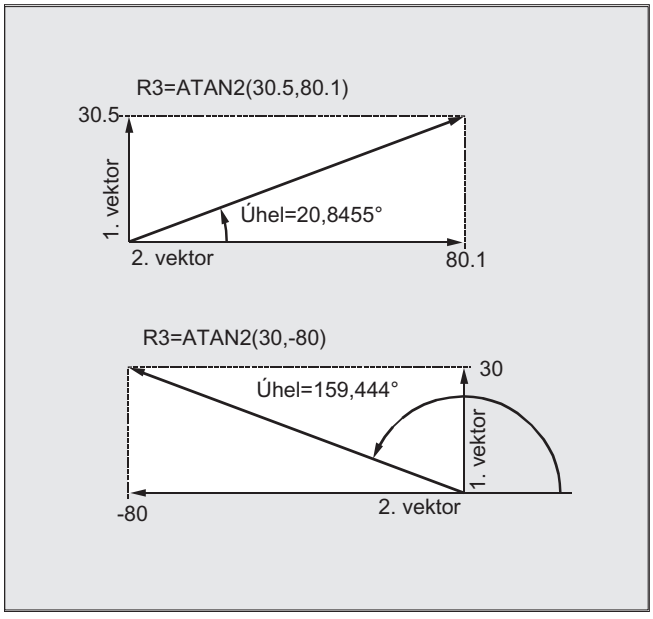
Příklad: 7 MOD 4.1 = 2.9

Arkus tangens na druhou: ATAN2

Matematická funkce ATAN2 vypočte ze dvou vůči sobě kolmo ležících vektorů úhel součtu těchto vektorů.

Výsledek vždy leží v jednom ze čtyř kvadrantů (-180° < 0 < +180°).

Základem pro úhlový vztah je vždy 2. hodnota v pozitivním směru.



Příklady programování

Programový kód	**Komentář**
R1=R1+1	; nová hodnota R1 = stará hodnota R1 + 1
R1=R2+R3 R4=R5-R6 R7=R8*R9	
R10=R11/R12 R13=SIN(25.3)	
R14=R1*R2+R3	; násobení se provádí před sečítáním
R14=(R1+R2)*R3	; výrazy v závorkách se vypočítají napřed
R15=SQRT(POT(R1)+POT(R2))	; napřed se vyhodnotí vnitřní závorky: R15 = druhá odmocnina((R1^2 + R2^2))
RESFRAME=FRAME1:FRAME2	; FRAME=spojení s řetězcovým operátorem
FRAME3=CTRANS(…):CROT(…)	Přiřazování hodnot složkám framu

4.1.3.2 Porovnávací a logické operace

Porovnávací operace se mohou používat např. pro formulování podmínky skoku. Přitom je možné porovnávat i složité výrazy.

Relační operace jsou použitelné pro proměnné typů CHAR, INT, REAL a BOOL. U typu CHAR je porovnávána kódová hodnota.
U typů STRING, AXIS a FRAME jsou možné operace == a <>, které se mohou používat také pro operace s typem STRING v synchronních akcích.

Výsledek relačních operací je vždy typu BOOL.

Logické operátory slouží ke spojení pravdivostních hodnot.

Logické operace mohou být používány jen s proměnnými typu BOOL. Prostřednictvím interního převádění typů je možno pracovat i s datovými typy CHAR, INT a REAL.

U logických (boolovských) operací platí pro datové typy BOOL, CHAR, INT a REAL:

- 0 odpovídá hodnotě: FALSE
- cokoli nerovnajícího se 0 odpovídá hodnotě: TRUE

Bitové logické operace

S proměnnými typu CHAR a INT mohou být prováděny i bitové logické operace. V případě potřeby proběhne konverze typu automaticky.

Programování

Relační operátor	Význam
==	rovná se
<>	nerovná se
>	je větší než
<	je menší než
>=	je větší nebo rovno
<=	je menší nebo rovno

Logický operátor	Význam
AND	logické a
OR	logické nebo
NOT	negace
XOR	logické XOR

Bitový logický operátor	Význam
B_AND	bitové logické a
B_OR	bitové logické nebo
B_NOT	bitová negace
B_XOR	bitové logické XOR

Poznámka

V aritmetických výrazech může být pomocí kulatých závorek definováno pořadí zpracování všech operátorů, které může být odlišné od normálních pravidel priority operací.

Poznámka

Mezi boolovskými operandy a operátory musí být zapisována mezera.

Poznámka

Operátor B_NOT se vztahuje pouze na jeden operand. Tento operand se nachází za operátorem.

Příklady

Příklad 1: Relační operátory
```
IF R10>=100 GOTOF ZIEL
```

nebo
```
R11=R10>=100
IF R11 GOTOF ZIEL
```

Výsledek porovnávání `R10>=100` se napřed ukládá do pomocné proměnné `R11`.

Příklad 2: Logické operátory
```
IF (R10<50) AND ($AA_IM[X]>=17.5) GOTOF ZIEL
```

nebo
```
IF NOT R10 GOTOB START
```

`NOT` se vztahuje pouze na jeden operand.

Příklad 3: Bitové logické operace
```
IF $MC_RESET_MODE_MASK B_AND 'B10000' GOTOF ACT_PLANE
```

4.1.3.3 Priorita operací

Každému operátoru je přiřazena priorita. Při vyhodnocení výrazu jsou stále nejprve použity operátory vyšší s prioritou. U operátorů stejné úrovně proběhne vyhodnocení zleva doprava.

V aritmetických výrazech může být pomocí kulatých závorek definováno pořadí zpracování všech operátorů, které může být odlišné od normálních pravidel priority operací.

Posloupnost operátorů

Od nejvyšší po nejnižší prioritu

1.	NOT, B_NOT	Negace, bitová negace
2.	*, /, DIV, MOD	Násobení, dělení
3.	+, –	Sečítání, odečítání
4.	B_AND	bitové logické a
5.	B_XOR	bitové logické XOR
6.	B_OR	bitové logické nebo
7.	AND	logické a
8.	XOR	logické XOR
9.	OR	logické nebo
10.	<<	Zřetězení řetězců, typ výsledku STRING
11.	==, <>, >, <, >=, <=	Relační operátory

Poznámka

Operátor zřetězení „:“ pro framy se nesmí vyskytnout s ostatními operátory v jednom výrazu. Zařazení tohoto operátoru do stupnice priority proto není nutné.

Příklad: Příkaz IF

```
If (otto==10) and (anna==20) gotof end
```

4.1.3.4 Korekce přesnosti při příkazech porovnávání (TRUNC)

Příkaz TRUNC oddělí celočíselnou část operandu po vynásobení faktorem přesnosti.

Nastavitelná přesnost u příkazů porovnávání

Data výrobního programu typu REAL jsou interně zobrazena v IEEE formátu se 64 bity. Na základě této formy zobrazení mohou být vyobrazena nepřesně desetinná čísla, která při porovnávání s ideálně vypočtenými hodnotami mohou vést k neočekávaným výsledkům.

Relativní shodnost

Aby formou zobrazení vyvolané nepřesnosti negativně neovlivňovaly běh programu, kontroluje se u příkazů porovnávání nikoli absolutní shodnost, nýbrž relativní shodnost.

Syntaxe

Korekce přesnosti při chybách porovnávání

```
TRUNC (R1*1000)
```

Význam

TRUNC:	Odříznutí míst za desetinnou čárkou

Uvažovaná relativní rovnost 10^{-12} při následujících operacích

- Rovnost: (==)
- Nerovnost: (<>)
- Je větší nebo rovno: (>=)
- Je menší nebo rovno: (<=)
- Je větší/menší: (><) s absolutní rovností
- Je větší: (>)
- Je menší: (<)

Kompatibilita

Z důvodu kompatibility může být zkouška relativní shodnosti u operací (>) a (<) nastavením strojního parametru MD10280 $MN_ PROG_FUNCTION_MASK Bit0 = 1 deaktivována.

Poznámka

Porovnávání s daty typu REAL je z výše uvedených důvodů obecně zatíženo určitou nepřesností. V případě neakceptovatelných odchylek musí být použit přepočet na typ INTEGER, při kterém jsou operandy vynásobeny faktorem přesnosti a pak jsou oříznuty pomocí funkce TRUNC.

Synchronní akce

Popisované chování relačních příkazů platí i při synchronních akcích.

Příklady

Příklad 1: Sledování shodnosti

Programový kód	Komentář
N40 R1=61.01 R2=61.02 R3=0.01	; přiřazení počátečních hodnot
N41 IF ABS(R2-R1) > R3 GOTOF FEHLER	; skok byl už dříve proveden
N42 M30	; konec programu
N43 FEHLER: SETAL(66000)	
R1=61.01 R2=61.02 R3=0.01	; přiřazení počátečních hodnot
R11=TRUNC(R1*1000) R12=TRUNC(R2*1000) R13=TRUNC(R3*1000)	; korekce přesnosti
IF ABS(R12-R11) > R13 GOTOF FEHLER	; skok se už neprovede
M30	; konec programu
FEHLER: SETAL(66000)	

Příklad 2: Vytvoření a vyhodnocení koeficientu obou operandů

Programový kód	Komentář
R1=61.01 R2=61.02 R3=0.01	; přiřazení počátečních hodnot
IF ABS((R2-R1)/R3)-1) > 10EX-5 GOTOF FEHLER	; skok se neprovede
M30	; konec programu
FEHLER: SETAL(66000)	

4.1.3.5 Zaokrouhlení (ROUNDUP)

Pomocí funkce "ROUNDUP" mohou být vstupní hodnoty typu REAL (desetinná čísla s desetinnou tečkou) zaokrouhlována na nejbližší vyšší celé číslo.

Syntaxe

```
ROUNDUP(<hodnota>)
```

Význam

ROUNDUP:	Příkaz pro zaokrouhlení vstupní hodnoty
<hodnota>:	Vstupní hodnota typu REAL

Poznámka

Vstupní hodnoty typu INTEGER (celá čísla) se přenesou do výsledku beze změny.

Příklady

Příklad 1: Různé vstupní hodnoty a výsledky jejich zaokrouhlení

Příklad:	Výsledek zaokrouhlení
`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

Příklad 2: Funkce ROUNDUP v NC programu

Programový kód

```
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])
...
```

4.1.4 Operace s řetězci

Operace s řetězci

Vedle klasických operací "přiřazování" a "Porovnávání" je s řetězci možno provádět ještě i následující operace:

- Převod do datového typu STRING (AXSTRING) (Strana 449)
- Převod z datového typu STRING (NUMBER, ISNUMBER, AXNAME) (Strana 449)
- Zřetězení proměnných typu String (<<) (Strana 450)
- Převádění na malá/velká písmena (TOLOWER, TOUPPER) (Strana 452)
- Zjištění délky řetězce (STRLEN) (Strana 452)
- Vyhledávání znaků/řetězců v řetězci (INDEX, RINDEX, MINDEX, MATCH) (Strana 453)
- Vybírání dílčího řetězce (SUBSTR) (Strana 454)
- Načítání a zápis jednotlivých znaků (Strana 455)
- Formátování řetězce (SPRINT) (Strana 456)

Zvláštní význam znaku 0

Znak 0 je interně interpretován jako identifikace konce řetězce. Je-li znak nahrazen znakem 0, je tím řetězec zkrácen.

Příklad:

Programový kód	Komentář
DEF STRING[20] STRG="Achse . steht"	
STRG[6]="X"	
MSG(STRG)	; výpis hlášení "Achse X steht"
STRG[6]=0	
MSG(STRG)	; výpis hlášení "Achse"

4.1.4.1 Převod do datového typu STRING (AXSTRING)

Pomocí funkce "Převod do datového typu STRING" se dají využít proměnné rozdílných typů jako součást jednoho hlášení (MSG).

Proběhne při použití operátoru << implicitně pro všechna data typu INT, REAL, CHAR a BOOL (viz" Zřetězení proměnných typu String (<<) (Strana 450) ").

Hodnota typu INT bude převedena do normálně čitelné formy. U hodnot typu REAL bude uvedeno až 10 míst za čárkou.

Pomocí příkazu AXSTRING mohou být na proměnné typu STRING převáděny také proměnné typu AXIS.

Syntaxe

```
<STRING_ERG> = << <libovolný typ>
<STRING_ERG> = AXSTRING(<identifikátor osy>)
```

Význam

<STRING_ERG>:	Proměnná pro výsledek převodu typu	
	Typ:	STRING
<libovolný typ>:	Proměnné typu INT, REAL, CHAR, STRING a BOOL	
AXSTRING:	Výsledkem příkazu AXSTRING je specifikovaný identifikátor osy jako řetězec.	
<identifikátor osy>:	Proměnná pro identifikátor osy	
	Typ:	AXIS

Poznámka

Proměnné typu FRAME konvertovat nelze.

4.1.4.2 Převod z datového typu STRING (NUMBER, ISNUMBER, AXNAME)

Prostřednictvím funkce NUMBER se provádí konverze z typu STRING do typu REAL. To, zda je konverze vůbec možná, lze zkontrolovat pomocí příkazu ISNUMBER.

Pomocí příkazu AXNAME je možné proměnnou typu STRING převádět na datový typ AXIS.

Syntaxe

```
<REAL_ERG>=NUMBER("<řetězec>")
<BOOL_ERG>=ISNUMBER("<řetězec>")
<AXIS_ERG>=AXNAME("<řetězec>")
```

Význam

NUMBER:	Výsledkem příkazu NUMBER je řetězcem <řetězec> představované číslo uložené jako reálná hodnota.		
<řetězec>:	Proměnná typu STRING, která má být konvertována		
<REAL_ERG>:	Proměnná pro výsledek převodu typu pomocí příkazu NUMBER		
	Typ:	REAL	
ISNUMBER:	Pomocí příkazu ISNUMBER je možné zkontrolovat, může-li být řetězec <řetězec> převeden na platné číslo.		
<BOOL_ERG>:	Proměnná pro výsledek dotazu pomocí příkazu ISNUMBER		
	Typ:	BOOL	
	Hodnota:	TRUE	Výsledkem funkce ISNUMBER je hodnota TRUE, když řetězec <řetězec> představuje podle pravidel jazyka platné reálné číslo.
		FALSE	Jestliže je výsledkem funkce ISNUMBER hodnota FALSE, je při vyvolání funkce NUMBER se stejným řetězcem <řetězec> spuštěn alarm.
AXNAME:	Funkce AXNAME převádí udaný řetězec <řetězec> na identifikátor osy. **Upozornění:** Pokud řetězec <řetězec> není možné přiřadit žádnému z identifikátorů os nastavených v konfiguraci, aktivuje se alarm.		
<AXIS_ERG>:	Proměnná pro výsledek převodu typu pomocí příkazu AXNAME		
	Typ:	AXIS	

Příklad:

Programový kód	**Komentář**
DEF BOOL BOOL_ERG	
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

4.1.4.3 Zřetězení proměnných typu String (<<)

Funkce "Zřetězení proměnných typu String" poskytuje možnost skládat řetězec z jednotlivých součástí.

Řetězení je realizováno prostřednictvím operátoru "<<". Dieser Operator hat für alle Kombinationen der Basistypen CHAR, BOOL, INT, REAL und STRING als Zieltyp STRING. Případně nutná konverze je provedena dle stávajících pravidel.

Syntaxe

```
<libovolný typ> << <libovolný typ>
```

Význam

<libovolný typ>:	Proměnné typu CHAR, BOOL, INT, REAL nebo STRING
<< :	Operátor pro zřetězení proměnných (<libovolný typ>) do takto poskládaného řetězce znaků (typ STRING). Tento operátor je k dispozici i samostatně jako tzv. „unární" varianta. Tak je možné provést explicitní změnu do typu STRING (ne pro typy FRAME a AXIS): << <libovolný typ>

Například se tak dá složit hlášení nebo povel z textových seznamů a dají se doplnit parametry (asi jedno jméno prvku):
`MSG(STRG_TAB[LOAD_IDX]<<BAUSTEIN_NAME)`

Poznámka

Mezivýsledky při zřetězení typu string nesmí překročit maximální délku řetězce.

Poznámka

Typy FRAME a AXIS nemohou být použity spolu s operátorem "<<".

Příklady

Příklad 1: Zřetězení proměnných typu String

Programový kód	Komentář
`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)`	`; Výpis: "Index:2/Wert:9.654"`
`NO_MSG:`	

Příklad 2: Explicitní konverze typu pomocí příkazu <<

Programový kód	Komentář
`DEF REAL VALUE=3.5`	
`<<VALUE`	`; Uvedená proměnná typu REAL je konvertována na typ STRING.`

4.1.4.4 Převádění na malá/velká písmena (TOLOWER, TOUPPER)

Funkce "Převádění na malá/velká písmena" umožňuje změnit všechna písmena řetězce znaků do jednotné formy.

Syntaxe

```
<STRING_ERG>=TOUPPER("<řetězec>")
<STRING_ERG>=TOLOWER("<řetězec>")
```

Význam

TOUPPER:	Prostřednictvím příkazu TOUPPER jsou všechna písmena v řetězci znaků převedena na **velká** písmena.	
TOLOWER:	Prostřednictvím příkazu TOLOWER jsou všechna písmena v řetězci znaků převedena na **malá** písmena.	
<řetězec>:	Řetězec znaků, který má být převeden.	
	Typ:	STRING
<STRING_ERG>:	Proměnná pro výsledek převodu	
	Typ:	STRING

Příklad:

Protože uživatelské vstupy mohou být zobrazovány také na uživatelském rozhraní, lze jim dát standardní formu (velká nebo malá písmena):

Programový kód

```
DEF STRING [29] STRG
...
IF "LEARN.CNC"==TOUPPER(STRG) GOTOF LOAD_LEARN
```

4.1.4.5 Zjištění délky řetězce (STRLEN)

Prostřednictvím příkazu STRLEN můžete zjistit délku řetězce znaků.

Syntaxe

```
<INT_ERG>=STRLEN("<STRING>")
```

Význam

STRLEN:	Prostřednictvím příkazu STRLEN se určuje délka zadaného řetězce znaků. Výsledkem bude počet znaků, které – počítáno od začátku řetězce znaků – nejsou znakem 0.	
<řetězec>:	Řetězec znaků, jehož délka má být určena.	
	Typ:	STRING

<INT_ERG>:	Proměnná pro výsledek určování	
	Typ:	INT

Příklad:

Funkce v souvislosti s přístupem k jednotlivým znakům umožňuje stanovit konec řetězce znaků.

Programový kód

```
IF(STRLEN (BAUSTEIN_NAME) > 10) GOTOF FEHLER
```

4.1.4.6 Vyhledávání znaků/řetězců v řetězci (INDEX, RINDEX, MINDEX, MATCH)

Tato funkce umožňuje hledat jednotlivé znaky popř. řetězec v jiné proměnné typu STRING. Výsledky této funkce udávají, na které pozici řetězce byl znak/řetězec v prohledávaném řetězci nalezen.

Syntaxe

`INT_ERG=INDEX(STRING,CHAR)` ; **typ výsledku: INT**

`INT_ERG=RINDEX(STRING,CHAR)` ; **typ výsledku: INT**

`INT_ERG=MINDEX(STRING,STRING)` ; **typ výsledku: INT**

`INT_ERG=MATCH(STRING,STRING)` ; **typ výsledku: INT**

Sémantika

Vyhledávací funkce: Zobrazují pozici v řetězci (první parametr), kde bylo hledání úspěšné. Jestliže znak/řetězec nemůže být nalezen, je vracena hodnota –1. První znak má přitom pozici 0.

Význam

INDEX:	Hledá znak zadaný jako druhý parametr (odpředu) v prvním parametru.
RINDEX:	Hledá znak zadaný jako druhý parametr (odzadu) v prvním parametru.
MINDEX:	Odpovídá funkci INDEX, s výjimkou toho, že je předán seznam znaků (jako řetězec), ze kterých je předán index prvního nalezeného znaku.
MATCH:	Hledá řetězec v řetězci.

Tak se dají řetězce rozložit podle určitých kritérií, např. podle pozic se znakem mezery nebo s oddělovacím znakem („/“) v udání cesty.

Příklad:

Rozložení zadání na název cesty a název modulu

Programový kód	Komentář
`DEF INT PFADIDX, PROGIDX`	
`DEF STRING[26] EINGABE`	

Programový kód	**Komentář**
DEF INT LISTIDX	
EINGABE = "/_N_MPF_DIR/_N_EXECU-TE_MPF"	
LISTIDX = MINDEX (EINGABE, "M,N,O,P") + 1	; jako hodnota v proměnné LISTIDX je vraceno 3, protože "N" je první znak od předu v parametru EINGABE ze seznamu možností
PFADIDX = INDEX (EINGABE, "/") +1	; přitom tedy platí: PFADIDX = 1
PROGIDX = RINDEX (EINGABE, "/") +1	; přitom tedy platí: PROGIDX = 12
	; s pomocí v další části uvedené funkce SUBSTR se dá proměnná EINGABE rozložit na součásti "cesta" a "modul":
VARIABLE = SUBSTR (EINGABE, PFADIDX, PROGIDX-PFADIDX-1)	; výsledkem pak je "_N_MPF_DIR"
VARIABLE = SUBSTR (EINGABE, PROGIDX)	; výsledkem pak je "_N_EXECUTE_MPF"

4.1.4.7 Vybírání dílčího řetězce (SUBSTR)

Pomocí funkce SUBSTRING je možné načíst libovolný úsek v rámci daného řetězce.

Syntaxe

```
<STRING_ERG>=SUBSTR(<řetězec>,<index>,<délka>)
<STRING_ERG>=SUBSTR(<řetězec>,<index>)
```

Význam

SUBSTR:	Funkce poskytuje z daného <řetězce> určitý dílčí řetězec, který začíná místem daným parametrem <index> a má délku stanovenou parametrem <délka>. Pokud parametr <délka> není uveden, vrací tato funkce dílčí řetězec, který začíná na místě daném parametrem <index> a končí posledním znakem původního řetězce.
<Index>:	Počáteční místo dílčího řetězce v původním řetězci. Leží-li počáteční pozice za koncem řetězce, je převzat prázdný (" ") řetězec. První znak řetězce: Index = 0 Rozsah hodnot: 0 ... (délka řetězce -1)
<délka>:	Délka dílčího řetězce. Pokud je zadána příliš velká délka, bude výsledkem funkce dílčí řetězec až do konce původního řetězce. Rozsah hodnot: 1 ... (délka řetězce -1)

Příklad:

Programový kód	**Komentář**
DEF STRING [29] ERG	
; 1	
; 0123456789012345678	
ERG = SUBSTR ("QUITTUNG: 10 bis 99", 10, 2)	; ERG == "10"
ERG = SUBSTR ("QUITTUNG: 10 bis 99", 10)	; ERG == "10 až 99"

4.1.4.8 Načítání a zápis jednotlivých znaků

Jednotlivé znaky v rámci určitého řetězce je možné číst i zapisovat.

Přitom je nutno mít na paměti následující okrajové podmínky:

- Je možné pracovat pouze s proměnnými definovanými uživatelem, nikoli se systémovými proměnnými.
- Jednotlivé znaky řetězce jsou předávány při volání podprogramu metodou "call by value" (volání hodnotou).

Syntaxe

```
<znak>=<řetězec>[<index>]
<znak>=<řetězec_pole>[<pole_index>,<index>]
<řetězec>[<index>]=<znak>
<řetězec_pole>[<pole_index>,<index>]=<znak>
```

Význam

<řetězec>:	Libovolný řetězec
<znak>:	Proměnná typu CHAR.
<Index>:	Poloha znaku v rámci řetězce První znak řetězce: Index = 0 Rozsah hodnot: 0 ... (délka řetězce -1)

Příklady

Příklad 1: Proměnné hlášení

```
Programový kód                          Komentář
;                          0123456789
DEF STRING [50] MELDUNG = "Osa n dosahla pozice"
MELDUNG [6] = "X"
MSG (MELDUNG)                           ; „osa X dosáhla pozice"
```

Příklad 2: Vyhodnocování systémových proměnných

```
Programový kód                          Komentář
DEF STRING [50] STRG                    ; vyrovnávací paměť pro systémové proměnné
...
STRG = $P_MMCA                          ; načítání systémové proměnné
IF STRG[0] == "E" GOTO ...              ; vyhodnocování systémové proměnné
```

Příklad 3: Předávání parametru metodou "call by value" a "call by reference"

```
Programový kód                                  Komentář
;                       0123456
DEF STRING[50] STRG = "Achse X"
```

Programový kód	**Komentář**
DEF CHAR CHR	
...	
EXTERN UP_VAL(ACHSE)	; definice podprogramu s parametrem typu "call by value"
EXTERN UP_REF(VAR ACHSE)	; definice podprogramu s parametrem typu "call by ref." Parametry
...	
UP_VAL(STRG[6])	; předávání parametru metodou "by value"
...	
CHR = STRG[6]	; uložení do vyrovnávací paměti
UP_REF(CHR)	; předávání parametru metodou "by reference"

4.1.4.9 Formátování řetězce (SPRINT)

Prostřednictvím předdefinované funkce SPRINT mohou být formátovány řetězce znaků a mohou být tak připravovány např. pro výstup na externím zařízení (viz také "Process DataShare - Výstup do externího zařízení/souboru (EXTOPEN, WRITE, EXTCLOSE) (Strana 1010)").

Syntaxe

```
"<Výsledek_řetězec>"=SPRINT("<Formát_řetězce>",<Hodnota_1>,<Hodnota_2>,..., <Hodnota_n>)
```

Význam

SPRINT:	Identifikátor předdefinované funkce, jejímž výsledkem je hodnota typu STRING.
"<Formát_řetězce>":	Řetězec znaků, který obsahuje pevné a proměnné složky. Proměnné složky jsou definovány pomocí řídícího formátovacího znaku % a následným popisem formátu.
< hodnota_1>,< hodnota_2>,…,< hodnota_n>:	Hodnota ve formě konstanty nebo proměnné NC systému, která se vkládá na místo, na němž se nachází n-tý řídící formátovací znak v souladu s popisem formátu v parametru <Formát_řetězce>.
"<Výsledek_řetězec>":	Formátovaný řetězec znaků (maximálně 400 bytů)

Popisy formátů, které jsou k dispozici

%B:	Přeměna na řetězec "TRUE", pokud se převáděná hodnota:
	• nerovná 0.
	• nerovná mezeře (v případě hodnot typu STRING).
	Přeměna na řetězec "FALSE", pokud se převáděná hodnota:
	• rovná 0.
	• je mezerou.
	Příklad:
	`N10 DEF BOOL BOOL_VAR=1`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF BOOL_VAR:%B", BOOL_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF BOOL_VAR:TRUE".

%C:	Přeměna na řetězec ASCII znaků.
	Příklad:
	`N10 DEF CHAR CHAR_VAR="X"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF CHAR_VAR:%C",CHAR_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF CHAR_VAR:X".

%D:	Přeměna na řetězec s celočíselnou hodnotou (INTEGER).
	Příklad:
	`N10 DEF INT INT_VAR=123`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF INT_VAR:%D",INT_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF INT_VAR:123".

%<m>D:	Přeměna na řetězec s celočíselnou hodnotou (INTEGER). Řetězec má minimální délku <m> znaků. Chybějící místa jsou zleva doplněna mezerami.
	Příklad:
	`N10 DEF INT INT_VAR=-123`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF INT_VAR:%6D",INT_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF INT_VAR:xx-123" ("x" v příkladu reprezentuje znak "mezera").

%F:	Přeměna na řetězec s desetinným číslem, které má 6 míst za desetinnou tečkou. Podle potřeby jsou místa za desetinnou tečkou zaokrouhlena nebo doplněna 0.
	Příklad:
	`N10 DEF REAL REAL_VAR=-1.2341234EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%F",REAL_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR: -1234.123400".

%<m>F:	Přeměna na řetězec s desetinným číslem, které má 6 míst za desetinnou tečkou a celkovou délku minimálně <m> znaků. Podle potřeby jsou místa za desetinnou tečkou zaokrouhlena nebo doplněna 0. Znaky, které chybějí, aby bylo dosaženo celkové délky <m>, jsou zleva doplněna mezerami.
	Příklad:
	`N10 DEF REAL REAL_VAR=-1.23412345678EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15F",REAL_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR: xxx-1234.123457" ("x" v příkladu reprezentuje znak "mezera").

%.<n>F:	Přeměna na řetězec s desetinným číslem, které má <n> míst za desetinnou tečkou. Podle potřeby jsou místa za desetinnou tečkou zaokrouhlena nebo doplněna 0.
	Příklad:
	`N10 DEF REAL REAL_VAR=-1.2345678EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3F",REAL_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR: -1234.568".

%<m>.<n>F:	Přeměna na řetězec s desetinným číslem, které má <n> míst za desetinnou tečkou a celkovou délku minimálně <m> znaků. Podle potřeby jsou místa za desetinnou tečkou zaokrouhlena nebo doplněna 0. Znaky, které chybějí, aby bylo dosaženo celkové délky <m>, jsou zleva doplněna mezerami.
	Příklad:
	`N10 DEF REAL REAL_VAR=-1.2341234567890EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%10.2F",REAL_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:xx-1234.12" ("x" v příkladu reprezentuje znak "mezera").

%E:	Přeměna na řetězec s desetinným číslem v exponenciálním způsobu zápisu. Mantisa je normalizována na jedno místo před desetinnou tečkou a 6 míst za desetinnou tečkou. Podle potřeby jsou místa za desetinnou tečkou zaokrouhlena nebo doplněna 0. Exponent začíná klíčovým slovem "EX". Potom následuje znaménko ("+" nebo "-") a dvoj- nebo trojmístné číslo.
	Příklad:
	`N10 DEF REAL REAL_VAR=-1234.567890`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%E",REAL_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:-1.234568EX+03".

%<m>E:	Přeměna na řetězec s desetinným číslem v exponenciálním formátu, který má celkovou délku minimálně <m> znaků. Chybějící znaky jsou zleva doplněny mezerami. Mantisa je normalizována na jedno místo před desetinnou tečkou a 6 míst za desetinnou tečkou. Podle potřeby jsou místa za desetinnou tečkou zaokrouhlena nebo doplněna 0. Exponent začíná klíčovým slovem "EX". Potom následuje znaménko ("+" nebo "-") a dvoj- nebo trojmístné číslo.
	Příklad:
	`N10 DEF REAL REAL_VAR=-1234.5`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%20E",REAL_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:xxxxxx-1.234500EX+03" ("x" v příkladu reprezentuje znak "mezera").

%.<n>E:	Přeměna na řetězec s desetinným číslem v exponenciálním způsobu zápisu. Mantisa je normalizována na jedno místo před desetinnou tečkou a <n> míst za desetinnou tečkou. Podle potřeby jsou místa za desetinnou tečkou zaokrouhlena nebo doplněna 0. Exponent začíná klíčovým slovem "EX". Potom následuje znaménko ("+" nebo "-") a dvoj- nebo trojmístné číslo.
	Příklad:
	`N10 DEF REAL REAL_VAR=-1234.5678`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.2E",REAL_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:-1.23EX+03".

%<m>.<n>E:	Přeměna na řetězec s desetinným číslem v exponenciálním formátu, který má celkovou délku minimálně <m> znaků. Chybějící znaky jsou zleva doplněny mezerami. Mantisa je normalizována na jedno místo před desetinnou tečkou a <n> míst za desetinnou tečkou. Podle potřeby jsou místa za desetinnou tečkou zaokrouhlena nebo doplněna 0. Exponent začíná klíčovým slovem "EX". Potom následuje znaménko ("+" nebo "-") a dvoj- nebo trojmístné číslo.
	Příklad:
	`N10 DEF REAL REAL_VAR=-1234.5678`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.2E", REAL_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:xx-1.23EX+03" ("x" v příkladu reprezentuje znak "mezera").

%G:	Přeměna na řetězec s desetinným číslem buď v desetinném nebo v exponenciálním způsobu zápisu, v závislosti na rozsahu hodnot: Pokud je zobrazovaná hodnota menší než 1.0EX-04 nebo větší/rovna 1.0EX+06, zvolí se exponenciální způsob zápisu, jinak se použije formát desetinného čísla. Vypisuje se maximálně šest významových míst, v případě nutnosti se číslo zaokrouhlí.
	Příklad s formátem desetinného čísla:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR: 0.000123457".
	Příklad s exponenciálním způsobem zápisu:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+06`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:1.23457EX+06".

%<m>G:	Přeměna na řetězec s desetinným číslem buď v desetinném nebo v exponenciálním způsobu zápisu, v závislosti na rozsahu hodnot (jako %G). Řetězec má celkovou délku minimálně <m> znaků. Chybějící znaky jsou zleva doplněny mezerami.
	Příklad s formátem desetinného čísla:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:xxxx0.000123457" ("x" v příkladu reprezentuje znak "mezera").
	Příklad s exponenciálním způsobem zápisu:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+06`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:xxx1.23457EX+06" ("x" v příkladu reprezentuje znak "mezera").

%.<n>G:	Přeměna na řetězec s desetinným číslem buď v desetinném nebo v exponenciálním způsobu zápisu, v závislosti na rozsahu hodnot. Vypisuje se maximálně <n> významových míst, v případě nutnosti se číslo zaokrouhlí. Pokud je zobrazovaná hodnota menší než 1.0EX-04 nebo větší/rovna 1.0EX(+<n>), zvolí se exponenciální způsob zápisu, jinak se použije formát desetinného čísla.
	Příklad s formátem desetinného čísla:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR: 0.000123".
	Příklad s exponenciálním způsobem zápisu:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT = SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:1.23EX+03".

%<m>.<n>G:	Přeměna na řetězec s desetinným číslem buď v desetinném nebo v exponenciálním způsobu zápisu, v závislosti na rozsahu hodnot (jako %.<n>G). Řetězec má celkovou délku minimálně <m> znaků. Chybějící znaky jsou zleva doplněny mezerami.
	Příklad s formátem desetinného čísla:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:xxx0.0001235" ("x" v příkladu reprezentuje znak "mezera").
	Příklad s exponenciálním způsobem zápisu:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF REAL_VAR:xx1.235EX+06" ("x" v příkladu reprezentuje znak "mezera").

%.<n>P:	Přeměna hodnoty typu REAL na hodnotu typu INTEGER, přičemž se bere v úvahu <n> míst za desetinnou tečkou. Výsledná hodnota typu INTEGER je reprezentována 32-bitovým binárním číslem. Pokud převáděná hodnota nemůže být zobrazena pomocí 32 bitů, operace se přeruší a aktivuje se alarm.
	Protože posloupnost bytů generovaná formátovacím příkazem %.<n>P může obsahovat také binární nuly, celkový vytvořený řetězec již neodpovídá konvencím pro datový typ STRING NC-systému. Proto jej není možné ukládat do proměnných typu STRING a ani jej nelze dále zpracovávat pomocí příkazů NC jazyka, které jsou určeny pro řetězce. Jediným možným použitím je předávání parametrů do příkazu WRITE s výstupem na odpovídající externí zařízení (viz následující příklad).
	Jestliže parametr <Formát_řetězce> obsahuje popis formátu typu %P, bude na výstup odeslán celý řetězec, s výjimkou binárního čísla generovaného pomocí parametru %.<n>P, v závislosti na nastavení parametru MD10750 $MN_SPRINT_FORMAT_P_CODE ve znakovém kódu ASCII, ISO (DIN 6024) nebo EIA (RS 244). Jestliže je naprogramován znak, který nemůže být převeden, bude zpracování přerušeno a aktivuje se alarm.
	Příklad:
	`N10 DEF REAL REAL_VAR=123.45`
	`N20 DEF INT ERROR`
	`N30 DEF STRING[20] EXT_DEVICE="/ext/dev/1"`
	`...`
	`N100 EXTOPEN(ERROR,EXT_DEVICE)`
	`N110 IF ERROR <> 0`
	`... ; zpracování chyby`
	`N200 WRITE(ERROR,EXT_DEVICE,SPRINT("INTEGER BINARY CODED:%.3P",REAL_VAR)`
	`N210 IF ERROR <> 0`
	`... ; zpracování chyby`
	Výsledek: Řetězec "INTEGER BINARY CODED: 'H0001E23A'" se přenese na výstupní zařízení /ext/dev/1. Hexadecimální hodnota 0x0001E23A odpovídá decimální hodnotě 123450.

%<m>.<n>P:	Převod hodnoty typu REAL na řetězec v souladu s nastavením strojního parametru MD10751 $MN_SPRINT_FORMAT_P_DECIMAL, přičemž tímto řetězcem je:
	• celé číslo s <m> + <n> místy **nebo**
	• desetinné číslo s maximální <m> místy před desetinnou tečkou a přesně <n> místy za desetinnou tečkou.
	Stejně jako v případě popisu formátu %.<n>P se celý řetězec ukládá ve znakovém kódu definovaném v parametru MD10750 $MN_SPRINT_FORMAT_P_CODE.
	Přeměna v případě MD10751 = 0:
	Hodnota typu REAL je převedena na řetězec s celým číslem, které má <m> + <n> míst. V případě potřeby jsou místa za desetinnou tečkou zaokrouhlena na <n> míst nebo doplněna 0. Chybějící místa před desetinnou tečkou jsou doplněna mezerami. Znaménko mínus se připojuje z levé strany, místo znaménka plus se vkládá mezera.
	Příklad:
	`N10 DEF REAL REAL_VAR=-123.45`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("PUNCHED TAPE FORMAT:%5.3P",REAL_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "PUNCHED TAPE FORMAT:-xx123450" ("x" v příkladu reprezentuje znak "mezera").
	Přeměna v případě MD10751 = 1:
	Hodnota typu REAL je převedena na řetězec s desetinným číslem, které má maximálně <m> míst před desetinnou tečkou a přesně <n> míst za desetinnou tečkou. V případě potřeby jsou místa před desetinnou tečkou ořízntua místa za desetinnou tečkou zaokrouhlena nebo doplněna 0. Jestliže se <n> rovná 0, odpadá i desetinná tečka.
	Příklad:
	`N10 DEF REAL REAL_VAR1=-123.45`
	`N20 DEF REAL REAL_VAR2=123.45`
	`N30 DEF STRING[80] RESULT`
	`N40 RESULT=SPRINT("PUNCHED TAPE FORMAT:%5.3P VAR2:%2.0P", REAL_VAR1,REAL_VAR2)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "PUNCHED TAPE FORMAT:-123.450 VAR2:23".

%S:	Slouží pro vkládání řetězce.
	Příklad:
	`N10 DEF STRING[16] STRING_VAR="ABCDEFG"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%S",STRING_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF STRING_VAR:ABCDEFG".

%<m>S:	Vložení řetězce s minimálně <m> znaky. Chybějící místa jsou doplněna mezerami.
	Příklad:
	`N10 DEF STRING[16] STRING_VAR="ABCDEFG"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10S",STRING_VAR)`
	Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF STRING_VAR:xxxABCDEFG" ("x" v příkladu reprezentuje znak "mezera").

%.<n>S:	Vložení <n> znaků řetězce (počínaje prvním znakem). **Příklad:** N10 DEF STRING[16] STRING_VAR="ABCDEFG" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%.3S",STRING_VAR) Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF STRING_VAR:ABC".
%<m>.<n>S:	Vložení <n> znaků řetězce (počínaje prvním znakem). Celková délka sestavovaného řetězce je minimálně <m> znaků. Chybějící místa jsou doplněna mezerami. **Příklad:** N10 DEF STRING[16] STRING_VAR="ABCDEFG" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10.5S", STRING_VAR) Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "CONTENT OF STRING_VAR:xxxABCDE" ("x" v příkladu reprezentuje znak "mezera").
%X:	Převod hodnoty typu INTEGER na řetězec v hexadecimálním způsobu zápisu. **Příklad:** N10 DEF INT INT_VAR='HA5B8' N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("INTEGER HEXADECIMAL:%X",INT_VAR) Výsledek: Do proměnné RESULT typu STRING bude zapsán řetězec "INTEGER HEXADECIMAL:A5B8".

Poznámka

Konvence jazyka NC systému týkající se rozlišování velkých a malých písmen v názvech identifikátorů a v klíčových slovech platí také pro popisy formátů. Při programování proto můžete používat velká i malá písmena, aniž by to mělo na funkce nějaký vliv.

Možnosti kombinování

V následující tabulce naleznete přehled o tom, které datové typy NC systémů a které popisy formátování mohou být kombinovány. Platí také pravidla pro implicitní převody datových typů (viz "Datové typy (Strana 424)").

	Datové typy NC systému						
	BOOL	**CHAR**	**INT**	**REAL**	**STRING**	**AXIS**	**FRAME**
%B	+	+	+	+	+	-	-
%C	-	+	-	-	+	-	-
%D	+	+	+	+	-	-	-
%F	-	-	+	+	-	-	-
%E	-	-	+	+	-	-	-
%G	-	-	+	+	-	-	-
%S	-	+	-	-	+	-	-
%X	+	+	+	-	-	-	-
%P	-	-	+	+	-	-	-

Poznámka

Tato tabulka ukazuje, že NC datové typy AXIS a FRAME není možné pomocí funkce SPRINT převádět přímo. Je ale možné:

- Datový typ AXIS je možné převádět na řetězec pomocí funkce AXSTRING, který je potom možné dále zpracovávat pomocí funkce SPRINT.
- Jednotlivé hodnoty datového typu FRAME je možné načítat pomocí funkcí pro přístup k jednotlivým složkám framu. Tímto způsobem získáte data typu REAL, která je pak možné pomocí příkazu SPRINT dále zpracovávat.

4.1.5 Programové skoky a větvení

4.1.5.1 Skok zpátky na začátek programu (GOTOS)

Pomocí příkazu GOTOS je možné skočit zpět na začátek hlavního programu nebo podprogramu za účelem jeho opakovaného zpracování.

Prostřednictvím strojních parametrů je možné nastavit, aby se při každém skoku zpět na začátek programu uskutečňovaly následující operace:

- Nastavení doby zpracovávání programu na "0".
- Zvýšení hodnoty počítadla obrobků o "1".

Syntaxe

```
GOTOS
```

Význam

GOTOS:	Příkaz skoku, kdy cílem skoku je začátek programu.	
	Vlastní realizace je ovládána signálem rozhraní NC/PLC: DB21, ... DBX384.0 (řízení větvení programu)	
	Hodnota:	Význam:
	0	Žádný skok zpět na začátek programu. Zpracovávání programu bude pokračovat následujícím blokem výrobního programu za příkazem GOTOS.
	1	Skok zpět na začátek programu. Výrobní program je opakován.

Okrajové podmínky

- Příkaz GOTOS interně spouští STOPRE (zastavení předběžného zpracování).
- U výrobních programů v definicí dat (proměnné LUD) se skok s příkazem GOTOS uskutečňuje na první programový blok za úsekem definic, tzn. definice dat se znovu neprovádějí. Definované proměnné si proto ponechávají hodnotu, kterou měly v bloku s příkazem GOTOS, a nejsou jim tedy dosazeny standardní hodnoty naprogramované v úseku definic.

- Jestliže má být pomocí funkce „Zpracování z externího zdroje" zpracován program s příkazy skoku, který se nachází v externí paměti programů, musí se cíle skoků nacházet ve vyrovnávací paměti.
  Tato podmínka představuje problém především u příkazů skoku na začátek programu (GOTOS), protože tyto programy jsou zpravidla příliš velké, takže se pro ně ve vyrovnávací paměti nenajde dost místa. Při prvním načtení se začátek programu z vyrovnávací paměti odstraní. Jestliže při následném zpracovávání programu dojde na provedení příkazu skoku na začátek programu, potom funkce není schopna cíl skoku najít. Dojde k přerušení programu a aktivování alarmu 14000.

  Poznámka

  Aby bylo možné zpracovávat externí programy bez omezení, pokud jde o naprogramované příkazy skoků, doporučujeme, abyste místo funkce „Zpracování z externího zdroje" používali funkci „Zpracování z externí paměti (EES)".

- V synchronních akcích a v technologických cyklech není příkaz GOTOS k dispozici.

Příklad

Programový kód	Komentář
N10 ...	; začátek programu
...	
N90 GOTOS	; skok na začátek programu
...	

4.1.5.2 Programové skoky na návěští skoků (GOTOB, GOTOF, GOTO, GOTOC)

Do programu je možno vkládat návěští skoků (label), na která lze potom skákat z jiných míst téhož programu pomocí příkazů GOTOF, GOTOB, GOTO, příp. GOTOC. Zpracování programu potom pokračuje příkazem, který se nachází bezprostředně za návěštím skoku. Tímto způsobem je možné realizovat větvení v rámci programu.

Kromě návěští skoků je možno jako cíl skoku používat také čísla hlavních a vedlejších bloků.

Jestliže je před příkazem kroku formulována podmínka skoku (IF ...), potom se programový skok uskuteční jen tehdy, pokud je podmínka skoku splněna.

Syntaxe

```
GOTOB <cíl skoku>
IF <podmínka skoku> == TRUE GOTOB <cíl skoku>

GOTOF <cíl skoku>
IF <podmínka skoku> == TRUE GOTOF <cíl skoku>

GOTO <cíl skoku>
IF <podmínka skoku> == TRUE GOTO <cíl skoku>

GOTOC <cíl skoku>
IF <podmínka skoku> == TRUE GOTOC <cíl skoku>
```

Význam

GOTOB:	Příkaz skoku s cílem skoku hledaným směrem k začátku programu	
GOTOF:	Příkaz skoku s cílem skoku hledaným směrem ke konci programu.	
GOTO:	Příkaz skoku s vyhledáváním cíle skoku. Vyhledávání se uskutečňuje napřed směrem ke konci programu, potom směrem k začátku programu.	
GOTOC:	Tento příkaz funguje stejně jako příkaz GOTO jen s tím rozdílem, že je potlačen alarm 14080 "Cíl skoku nenalezen". To znamená, že se zpracovávání programu v případě neúspěšného vyhledávání cíle skoku nepřeruší, nýbrž bude pokračovat na programovém řádku, který se nachází za příkazem GOTOC.	
<cíl skoku>:	Parametr cíle skoku Možné údaje jsou následující:	
	<návěští skoku>:	Cíl skoku je návěští skoku umístěné v programu, jehož název je definován uživatelem: <návěští skoku>:
	<číslo bloku>:	Cíl skoku je číslo hlavního či vedlejšího bloku (např.: 200, N300)
	Proměnná typu STRING:	Proměnný cíl skoku. Proměnná představuje návěští skoku nebo číslo bloku.
IF:	Klíčové slovo pro formulování podmínky skoku. Podmínka skoku připouští všechny porovnávací a logické operátory (výsledek: TRUE nebo FALSE). Programový skok se uskuteční, jestliže výsledek této operace je TRUE.	

Poznámka

Návěští skoku (label)

Návěští skoku se vždy nachází na začátku bloku. Jestliže existuje také číslo programu, návěští skoku se zapisuje vždy bezprostředně za číslem bloku.

Pro určování názvů návěští skoků platí následující pravidla:

- Počet znaků:
  - minimálně 2
  - maximálně 32
- Povolenými znaky jsou:
  - Písmena
  - Číslice
  - Znak podtržení
- První dva znaky musí být písmena nebo znak podtržení.
- Za názvem návěští skoku následuje dvojtečka (":“).

Okrajové podmínky

- Cílem skoku může být pouze blok s návěštím skoku nebo s číslem bloku, které se nacházejí **uvnitř** daného programu.
- Příkaz skoku bez podmínky skoku musí být naprogramován v samostatném bloku. U příkazů skoku s podmínkou skoku toto omezení neplatí. V jednom bloku může být formulováno několik příkazů skoků.
- U programů s příkazy skoků bez podmínek skoků se může stát, že se konec programu M2/M30 nebude nacházet na konci programu.
- Jestliže má být pomocí funkce „Zpracování z externího zdroje" zpracován program s příkazy skoku, který se nachází v externí paměti programů, musí se cíle skoků nacházet ve vyrovnávací paměti. Jinak nebude cíl skoku nalezen a dojde k přerušení programu a aktivování alarmu 14000.

Poznámka

Aby bylo možné zpracovávat externí programy bez omezení, pokud jde o naprogramované příkazy skoků, doporučujeme, abyste místo funkce „Zpracování z externího zdroje" používali funkci „Zpracování z externí paměti (EES)".

Příklady

Příklad 1: Skoky na návěští skoku

```
Programový kód                  Komentář
N10 …
N20 GOTOF Label_1               ; skok ve směru konce programu na
                                ; návěští skoku "Label_1".
N30 …
N40 Label_0: R1=R2+R3           ; návěští skoku "Label_0" nastaveno
N50 …
N60 Label_1:                    ; návěští skoku "Label_1" nastaveno
N70 …
N80 GOTOB Label_0               ; skok ve směru začátku programu
                                ; na návěští skoku "Label_0".
N90 …
```

Příklad 2: Nepřímý skok na číslo bloku

```
Programový kód                  Komentář
IF <podmínka> == TRUE
   R10=100                      ; přiřazení cíle skoku
ELSE
   R10=110                      ; přiřazení cíle skoku
ENDIF
; skok ve směru konce programu na blok, jehož číslo bloku je uloženo v R10
N10 GOTOF "N"<<R10
...
```

Programový kód	Komentář
N90 ...	
N100 ...	; cíl skoku
N110 ...	
...	

Příklad 3: Skok na proměnný cíl skoku

Programový kód	Komentář
DEF STRING[20] ZIEL	
IF <podmínka> == TRUE	
ZIEL = „Marke1"	; přiřazení cíle skoku
ELSE	
ZIEL = "Marke2"	; přiřazení cíle skoku
ENDIF	
; skok ve směru konce programu na proměnný cíl skoku, která je v proměnné „ZIEL"	
GOTOF ZIEL	
Marke1: T="Bohrer1"	; cíl skoku 1
...	
Marke2: T="Bohrer2"	; cíl skoku 2
...	

Příklad 4: Skok s podmínkou skoku

Programový kód	Komentář
N40 R1=30 R2=60 R3=10 R4=11 R5=50 R6=20	; přiřazení počátečních hodnot
N41 LA1: G0 X=R2*COS(R1)+R5 Y=R2*SIN(R1)+R6	; návěští skoku LA1
N42 R1=R1+R3 R4=R4-1	
; IF podmínka skoku == TRUE	
; THEN skok ve směru začátku programu na návěští skoku LA1	
N43 IF R4>0 GOTOB LA1	
N44 M30	; konec programu

4.1.5.3 Větvení programu (CASE ... OF ... DEFAULT ...)

Funkce CASE nabízí možnost překontrolovat aktuální hodnotu (typ: INT) nějaké proměnné nebo matematické funkce a v závislosti na výsledku skočit v programu na různá místa.

Syntaxe

```
CASE(<výraz>) OF <konstanta_1> GOTOF <cíl skoku_1> <konstanta_2>
GOTOF <cíl skoku_2> ... DEFAULT GOTOF <cíl skoku_n>
```

Význam

CASE:	Příkaz skoku
<výraz>:	Proměnná nebo matematická funkce

OF:	Klíčové slovo pro formulování podmíněného programového větvení.	
<konstanta_1>:	První zadaná konstantní hodnota pro proměnnou nebo matematickou funkci	
	Typ:	INT
<konstanta_2>:	Druhá zadaná konstantní hodnota pro proměnnou nebo matematickou funkci	
	Typ:	INT
DEFAULT:	Pro případy, při kterých se hodnota proměnné nebo matematická funkce nerovná žádné z uvedených konstant, může být cíl skoku určen pomocí příkazu DEFAULT. **Upozornění:** Jestliže příkaz DEFAULT není naprogramován, použije se v těchto případech jako cíl skoku blok, který následuje za příkazem CASE.	
GOTOF:	Příkaz skoku s cílem skoku hledaným směrem ke konci programu. Namísto příkazu GOTOF je možné naprogramovat také všechny ostatní příkazy GOTO (viz kapitola "Programové skoky na návěští skoku").	
<cíl skoku_1>:	Na tento cíl skoku se odbočí, jestliže hodnota proměnné nebo matematické funkce odpovídá první ze zadaných konstant. Cíl skoku může být zadán následujícím způsobem:	
	<návěští skoku>:	Cíl skoku je návěští skoku umístěné v programu, jehož název je definován uživatelem: <návěští skoku>:
	<číslo bloku>:	Cíl skoku je číslo hlavního či vedlejšího bloku (např.: 200, N300)
	Proměnná typu STRING:	Proměnný cíl skoku. Proměnná představuje návěští skoku nebo číslo bloku.
<cíl skoku_2>:	Na tento cíl skoku se odbočí, jestliže hodnota proměnné nebo matematické funkce odpovídá druhé ze zadaných konstant.	
<cíl skoku_n>:	Na tento cíl skoku se odbočí, jestliže hodnota proměnné nebo matematické funkce neodpovídá žádné z uvedených konstant.	

Okrajové podmínky

Zpracování z externího zdroje

Jestliže má být pomocí funkce „Zpracování z externího zdroje“ zpracován program s funkcí CASE, který se nachází v externí paměti programů, musí se cíle skoků nacházet ve vyrovnávací paměti. Jinak nebude cíl skoku nalezen a dojde k přerušení programu a aktivování alarmu 14000.

Poznámka

Aby bylo možné zpracovávat externí programy bez omezení, pokud jde o naprogramované příkazy skoků, doporučujeme, abyste místo funkce „Zpracování z externího zdroje“ používali funkci „Zpracování z externí paměti (EES)“.

Příklad

Programový kód
...
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
...

Příkaz CASE z bloku N30 definuje následující možnosti rozvětvení programu:

1. Jestliže se hodnota matematické funkce VAR1+VAR2-VAR3 = 7, potom se skáče na blok s definicí návěští skoku "Label_1" (--> N40).
2. Jestliže se hodnota matematické funkce VAR1+VAR2-VAR3 = 9, potom se skáče na blok s definicí návěští skoku "Label_2" (--> N50).
3. Jestliže se hodnota matematické funkce VAR1+VAR2-VAR3 nerovní ani 7 ani 9, potom se skáče na blok s definicí návěští skoku "Label_3" (--> N60).

4.1.6 Opakování části programu (REPEAT, REPEATB, ENDLABEL, P)

Opakování části programu umožňuje opětovné zpracování již napsané části v rámci daného programu v libovolné podobě.

Řádky, příp. úsek programu, který se má opakovat, je označen návěštími skoku (label).

Poznámka

Návěští skoku (label)

Návěští skoku se vždy nachází na začátku bloku. Jestliže existuje také číslo programu, návěští skoku se zapisuje vždy bezprostředně za číslem bloku.

Pro určování názvů návěští skoků platí následující pravidla:

- Počet znaků:
  - minimálně 2
  - maximálně 32
- Povolenými znaky jsou:
  - Písmena
  - Číslice
  - Znak podtržení
- První dva znaky musí být písmena nebo znak podtržení.
- Za názvem návěští skoku následuje dvojtečka (":“).

Syntaxe

1. Opakování jednoho jediného programového řádku:

```
<návěští skoku>: ...
...
REPEATB <návěští skoku> P=<n>
...
```

2. Opakování úseku programu mezi návěštím skoku a příkazem REPEAT:

```
<návěští skoku>: ...
...
REPEAT <návěští skoku> P=<n>
...
```

3. Opakování úseku mezi dvěma návěštími skoku:

```
<počáteční návěští skoku>: ...
...
<koncové návěští skoku>: ...
...
REPEAT <počáteční návěští skoku> <koncové návěští sko-
ku> P=<n>
...
```

Poznámka

Není možné, aby se příkaz REPEAT nacházel mezi počátečním a koncovým návěštím. Jestliže se <počáteční návěští skoku> nachází před příkazem REPEAT a pokud <koncové návěští skoku> není dosaženo před příkazem REPEAT, potom bude se opakován úsek programu mezi <počátečním návěštím skoku> a příkazem REPEAT.

4. Opakování úseku mezi návěštím skoku a návěštím ENDLABEL:

```
<návěští skoku>: ...
...
ENDLABEL: ...
...
REPEAT <návěští skoku> P=<n>
...
```

Poznámka

Není možné, aby se příkaz REPEAT nacházel mezi <návěštím skoku> a návěštím ENDLABEL. Jestliže se <počáteční návěští skoku> nachází před příkazem REPEAT a pokud návěští ENDLABEL není dosaženo před příkazem REPEAT, potom bude se opakován úsek programu mezi <počátečním návěštím skoku> a příkazem REPEAT.

Význam

REPEATB:	Příkaz pro opakování řádku programu.
REPEAT:	Příkaz pro opakování úseku programu.
<návěští skoku>:	Řetězec <návěští skoku> označuje: • Řádek programu, který má být opakován (v případě příkazu REPEATB) nebo • Začátek úseku programu, který má být opakován (v případě příkazu REPEAT)
	Řádek programu označený jako<návěští skoku> se může nacházet před nebo za příkazem REPEAT / REPEATB. Hledání se napřed provádí směrem k začátku programu. Pokud návěští skoku není v tomto směru nalezeno, spustí se hledání směrem ke konci programu. **Výjimka:** Jestliže má být opakován úsek programu mezi návěštím skoku a příkazem REPEAT (viz bod 2. v odstavci Syntaxe), potom se musí být programový řádek označený jako <návěští skoku> nacházet **před** příkazem REPEAT, protože v tomto případě se vyhledávání uskutečňuje **pouze** ve směru začátku programu. Jestliže řádek s návěštím <návěští skoku> obsahuje další příkazy, budou tyto příkazy znovu prováděny při každém opakování.
ENDLABEL:	Klíčové slovo, které označuje konec úseku programu, který se má opakovat. Jestliže řádek s návěštím ENDLABEL obsahuje další příkazy, budou tyto příkazy znovu prováděny při každém opakování. Příkaz ENDLABEL může být v programu použit i vícekrát.
P:	Adresa pro zadání počtu opakování
<n>:	Počet opakování úseku programu
	Typ: INT
	Opakovaný úsek programu bude opakován <n>-krát. Po posledním opakování bude zpracování programu pokračovat řádkem, který následuje za příkazy REPEAT / REPEATB. **Upozornění:** Jestliže příkaz P=<n> není uveden, bude opakovaný úsek programu opakován právě jedenkrát.

Příklady

Příklad 1: Opakování jednoho jediného programového řádku

```
Programový kód                          Komentář
N10 POSITION1: X10 Y20
N20 POSITION2: CYCLE(0,,9,8)            ; cyklus pro polohování
N30 ...
N40 REPEATB POSITION1 P=5               ; blok N10 provést pětkrát
N50 REPEATB POSITION2                   ; blok N20 provést jednou
N60 ...
N70 M30
```

Příklad 2: Opakování úseku programu mezi návěštím skoku a příkazem REPEAT

```
Programový kód                  Komentář
N5 R10=15
N10 Begin: R10=R10+1            ; šířka
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            ; úsek programu N10 až N70 provést 4x
N90 Z10
N100 M30
```

Příklad 3: Opakování úseku mezi dvěma návěštími skoku

```
Programový kód                     Komentář
N5 R10=15
N10 Begin: R10=R10+1               ; šířka
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          ; úsek programu N10 až N70 provést 3x
N110 Z10
N120 M30
```

Příklad 4: Opakování úseku mezi návěštím skoku a návěštím ENDLABEL

```
Programový kód                  Komentář
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
```

```
Programový kód                  Komentář
N130 REPEAT BEGIN3 P=3          ; úsek programu N110 až N120 provést 3x
N140 REPEAT BEGIN2 P=2          ; úsek programu N50 až N80 provést dvakrát
N150 M100
N160 REPEAT BEGIN1 P=2          ; úsek programu N20 až N80 provést dvakrát
N170 Z10
N180 X0 Y0
N190 M30
```

Příklad 5: Obrábění frézováním, opracování pozic pro vrtání pomocí různých technologií

```
Programový kód                  Komentář
N10 ZENTRIERBOHRER()            ; výměna nástroje pro navrtávání středicích důlků
N20 POS_1:                      ; pozice pro vrtání 1
N30 X1 Y1
N40 X2
N50 Y2
N60 X3 Y3
N70 ENDLABEL:
N80 POS_2:                      ; pozice pro vrtání 2
N90 X10 Y5
N100 X9 Y-5
N110 X3 Y3
N120 ENDLABEL:
N130 BOHRER()                   ; výměna nástroje za vrták a cyklus vrtání
N140 GEWINDE(6)                 ; výměna závitníku M6 a cyklus pro vrtání závitu
N150 REPEAT POS_1               ; opakování úseku programu od POS_1 do návěští ENDLABEL
                                jedenkrát
N160 BOHRER()                   ; výměna nástroje za vrták a cyklus vrtání
N170 GEWINDE(8)                 ; výměna závitníku M8 a cyklus pro vrtání závitu
N180 REPEAT POS_2               ; opakování úseku programu od POS_2 do návěští ENDLABEL
                                jedenkrát
N190 M30
```

Další informace

Vnoření

Opakování části programu může být voláno i vnořeným způsobem. Každé takové volání obsazuje jednu úroveň podprogramů.

M17 / RET

Pokud je v průběhu zpracování opakování části programu naprogramován příkaz `M17` nebo `RET`, opakování části programu se přeruší. Zpracování programu bude pokračovat blokem, který následuje za řádkem s příkazem `REPEAT`.

Zobrazení programu

V okně se zobrazením aktuálního programu se vypisuje opakování části programu jako vlastní úroveň podprogramu.

Zrušení úrovně

Jestliže během zpracování části programu dojde ke zrušení úrovně, program bude pokračovat na místě za voláním opakování úseku programu.

Příklad:

```
Programový kód                    Komentář
N5 R10=15
N10 BEGIN: R10=R10+1              ; šířka
N20 Z=10-R10
N30 G1 X=R10 F200
N40 Y=R10                         ; zrušení úrovně
N50 X=-R10
N60 Y=-R10
N70 END: Z10
N80 Z10
N90 CYCLE(10,20,30)
N100 REPEAT BEGIN END P=3
N120 Z10                          ; zpracovávání programu bude pokračovat zde
N130 M30
```

Řídící struktury a opakování částí programu

Řídící struktury a opakování částí programu mohou být používány kombinovaně. Neměly by se však mezi nimi vyskytovat žádná překrývající se místa. Opakování úseku programu by mělo být uvnitř větve řídící struktury, příp. řídící struktura by měla ležet uvnitř opakování úseku programu.

Skoky a opakování částí programu

V případě směšování skoků a opakování úseku programu jsou bloky zpracovávány čistě sekvenčně. Jestliže se vyskytuje např. skok z opakování části programu, bude se program tak dlouho zpracovávat, dokud nebude nalezen naprogramovaný konec úseku programu.

Příklad:

```
Programový kód
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
```

Programový kód
N110 X0 Y0 Z0
N120 Z-10
N130 REPEAT BEGIN1 P=2
N140 Z10
N150 X0 Y0
N160 M30

Poznámka

Příkaz `REPEAT` by se měl nacházet za bloky posuvu.

Zpracování z externího zdroje a opakování částí programu

V případě externích programů, ve kterých je určitý úsek programu opakován, se musí opakované řádky programu (je-li použit příkaz REPEATB), příp. začátek úseku programu, který má být opakován (je-li použit příkaz REPEAT), nacházet ve vyrovnávací paměti. Jinak nebude cíl skoku nalezen a dojde k přerušení programu a aktivování alarmu 14000.

Poznámka

Aby bylo možné zpracovávat externí programy bez omezení, pokud jde o opakování úseků programu, doporučujeme, abyste místo funkce „Zpracování z externího zdroje" používali funkci „Zpracování z externí paměti (EES)".

4.1.7 Řídící struktury

Řídící systém standardně zpracovává NC bloky v naprogramované posloupnosti.

Tato posloupnost může být naprogramováním alternativních programových bloků a programových smyček změněna. Pro programování těchto řídících struktur se používají klíčová slova IF, ELSE, ENDIF, LOOP, FOR, WHILE a REPEAT.

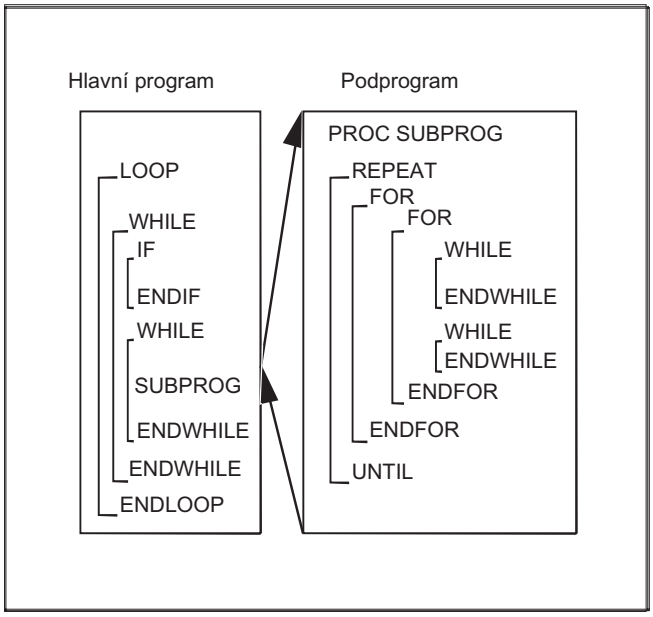
UPOZORNĚNÍ
Chyba v programu
Řídící struktury je možné používat pouze v rámci příkazové části programu. Definice v hlavičce programu nemohou být prováděny ani podmíněně, ani opakovaně.
Klíčová slova pro řídící struktury, stejně jako cíle skoku, nesmí být superponovány makry. Kontrola při definici maker se v této záležitosti neprovádí.

Platnost

Řídící strukturu není možné použít tak, aby přesahovala hranice programu.

Hloubka vnoření

V rámci každé úrovně podprogramu je možná hloubka vnoření až 16 řídících struktur.



Chování doby zpracování

Při standardně aktivním režimu překladače je možné v důsledku použití programových skoků dosáhnout rychlejšího zpracování než s řídícími strukturami.

U předem přeložených cyklů není mezi programovými skoky a řídícími strukturami žádný rozdíl.

Výpis aktuálního bloku u programových smyček

Jestliže jsou v rámci programové smyčky zpracovávány pouze bloky předběžného zpracování, ve výpisu aktuálního bloku se bude zobrazovat poslední blok hlavní větve zpracovávaný **před** programovou smyčkou.

Aby se např. pro potřeby diagnostiky objevovaly ve výpisu aktuálního bloku také bloky předběžného zpracování, musí být aktivován příkaz SBL2 pro dekódování jednotlivých bloků.

Broušení bez bloku v hlavní větvi programu

Jestliže není v rámci programové smyčky naprogramován žádný blok v hlavní větvi programu, bude smyčka zpracovávána v rámci předběžného zpracování tak dlouho, dokud nebude podmínka smyčky splněna.

Tímto způsobem vzniká vysoké zatížení systému a může být nepříznivě ovlivněno zobrazování dat.

Jako řešení se nabízí vložení příkazu STOPRE nebo doby prodlevy G04 o délce 0 sekund **dovnitř** smyčky.

Okrajové podmínky

Vypisování bloku

Bloky s prvky řídících struktur nemohou být přeskakovány.

Řídící struktury a programové skoky

Návěští skoku (label) jsou v blocích s prvky řídících struktur nepřípustné.

Poznámka

V zásadě se doporučuje nepoužívat řídící struktury a programové skoky současně.

Zpracování z externího zdroje

V případě externích programů s řídícími strukturami se musí začátek smyčky nacházet ve vyrovnávací paměti. Jinak nebude cíl skoku nalezen a dojde k přerušení programu a aktivování alarmu 14000.

Poznámka

Aby bylo možné zpracovávat externí programy bez omezení, pokud jde o řídící struktury, doporučujeme, abyste místo funkce „Zpracování z externího zdroje" používali funkci „Zpracování z externí paměti (EES)".

Režim překladače

Řídící struktury jsou zpracovávány podle překladače. Když je rozpoznán konec smyčky, bude vyhledán začátek této smyčky a přitom se budou brát v úvahu další nalezené řídící struktury. Z tohoto důvodu není v režimu překladače zkontrolována bloková struktura programu úplně.

Přípravné zpracování cyklů

Při přípravě cyklů může být správné vnoření řídících struktur zkontrolováno.

4.1.7.1 Podmíněné příkazy a větvení (IF, ELSE, ENDIF)

Podmíněný příkaz: IF - programový blok - ENDIF

V případě podmíněného příkazu je programový blok nacházející se mezi příkazy `IF` a `ENDIF` zpracován jen tehdy, pokud je podmínka splněna.

Větvení: IF - programový blok_1 - ELSE - programový blok_2 - ENDIF

V případě větvení se vždy zpracovává jeden ze dvou programových bloků.

Pokud je podmínka splněna, je zpracován programový blok 1 nacházející se mezi příkazy `IF` a `ELSE`.

Pokud podmínka splněna **není**, je zpracován programový blok 2 nacházející se mezi příkazy ELSE a ENDIF.

Poznámka

Příkaz ELSE v synchronních akcích

Klíčové slovo ELSE může být naprogramováno také v synchronních akcích. Tímto způsobem může být synchronní akce rozšířena o operace, které mají být provedeny, jestliže podmínka není splněna.

Syntaxe

Podmíněný příkaz

```
IF <podmínka>
   Programový blok                    ; zpracovat, pokud: <podmínka> == TRUE
ENDIF
```

Větvení

```
IF <podmínka>
   Programový blok_1                  ; zpracovat, pokud: <podmínka> == TRUE
ELSE
   Programový blok_2                  ; zpracovat, pokud: <podmínka> == FALSE
ENDIF
```

Význam

IF:	Uvádí podmíněný příkaz, příp. větvení.
ELSE:	Uvádí alternativní programový blok.
ENDIF:	Označuje konec podmíněného příkazu, příp. větvení.
<podmínka>:	Logický výraz, z jehož vyhodnocení vyplývá buď TRUE nebo FALSE.

Příklad: Podprogram pro výměnu nástroje

Programový kód	**Komentář**
PROC L6	Rutina pro výměnu nástroje
N500 DEF INT TNR_AKTUELL	Proměnná pro aktivní T-číslo.
N510 DEF INT TNR_VORWAHL	Proměnná pro předem zvolené T-číslo.
	Zjištění aktuálního nástroje.
N520 STOPRE	
N530 IF $P_ISTEST	V režimu testování programu se ...
N540 TNR_AKTUELL = $P_TOOLNO	... z kontextu programu načte "aktuální" nástroj

Programový kód	**Komentář**
N550 ELSE	V opačném případě se ...
N560 TNR_AKTUELL = $TC_MPP6[9998,1]	... načte nástroj ve vřetenu
N570 ENDIF	
N580 GETSELT(TNR_VORWAHL)	Načtení T-čísla předem zvoleného nástroje pro vřeteno.
N590 IF TNR_AKTUELL <> TNR_VORWAHL	Pokud předem zvolený nástroj ještě není aktuálním nástrojem, potom ...
N600 G0 G40 G60 G90 SUPA X450 Y300 Z300 D0	... najíždění na bod pro výměnu nástroje ...
N610 M206	... a provedení výměny nástroje
N620 ENDIF	
N630 M17	

4.1.7.2 Nekonečná programová smyčka (LOOP, ENDLOOP)

Nekonečná smyčka nachází uplatnění v nekonečných programech. Na konci smyčky se vždy provádí skok zpět na její začátek.

Syntaxe

```
LOOP
...
ENDLOOP
```

Význam

LOOP:	Uvádí nekonečnou smyčku.
ENDLOOP:	Označuje konec smyčky a způsobuje návrat zpět na její začátek.

Příklad:

Programový kód
...
LOOP
MSG ("Žádný břit nástroje není aktivní")
M0
STOPRE
ENDLOOP
...

4.1.7.3 Smyčka s počítadlem (FOR ... TO ..., ENDFOR)

Smyčka s počítadlem se používá, jestliže má být opakovaně zpracován úsek programu a počet těchto opakování je předem znám.

Syntaxe

```
FOR <proměnná> = <počáteční hodnota> TO <konečná hodno-
ta>
...
ENDFOR
```

Význam

FOR:	Uvádí smyčku s počítadlem.	
ENDFOR:	Označuje konec smyčky a způsobuje skok zpět na její začátek, a to tak dlouho, dokud není dosaženo koncové hodnoty počítadla.	
<proměnná>:	Počítající proměnná, která načítá nahoru od počáteční až po koncovou hodnotu a s každým průchodem se o hodnotu "1" zvyšuje.	
	Typ	INT nebo REAL **Upozornění** Typ REAL se použije, jestliže je jako počítadlo smyčky naprogramován např. R-parametr. Jestliže je jako počítadlo použita proměnná typu REAL, její hodnota se zaokrouhluje na celé číslo.
<počáteční hodnota>:	Počáteční hodnota pro počítání Podmínka: Počáteční hodnota musí být menší než hodnota konečná.	
<konečná hodnota>:	Konečná hodnota pro počítání	

Příklady

Příklad 1: Proměnná typu INTEGER nebo R-parametr jako proměnná počítadla

Proměnná typu INTEGER jako proměnná počítadla:

Programový kód	**Komentář**
DEF INT iVARIABLE1	
R10=R12-R20*R1 R11=6	
FOR iVARIABLE1= R10 TO R11	; proměnná počítadla = proměnná typu INTEGER
R20=R21*R22+R33	
ENDFOR	
M30	

R-Parametr jako proměnná počítadla:

Programový kód	**Komentář**
R11=6	

Programový kód	Komentář
FOR R10=R12-R20*R1 TO R11	; proměnná počítadla = R-parametr (proměnná typu REAL)
R20=R21*R22+R33	
ENDFOR	
M30	

Příklad 2: Výroba pevně daného počtu kusů

Programový kód	Komentář
DEF INT STUECKZAHL	; definice proměnné typu INT s názvem "STUECKZAHL"
FOR STUECKZAHL = 0 TO 100	; uvádí smyčku s počítadlem Proměnná "STUECKZAHL" se bude postupně zvyšovat od počáteční hodnoty "0" až po konečnou hodnotu "100".
G01 …	
ENDFOR	; konec smyčky s počítadlem
M30	

4.1.7.4 Programová smyčka s podmínkou na začátku smyčky (WHILE, ENDWHILE)

U smyčky WHILE se podmínka nachází na začátku smyčky. Dokud je podmínka splněna, bude se smyčkou WHILE procházet.

Syntaxe

```
WHILE <podmínka>
...
ENDWHILE
```

Význam

WHILE:	Uvádí programovou smyčku.
ENDWHILE:	Označuje konec smyčky a způsobuje návrat zpět na její začátek.
<podmínka>:	Podmínka, která musí být splněna, aby se smyčkou WHILE procházelo.

Příklad:

Programový kód	Komentář
...	
WHILE $AA_IW[BOHRACHSE] > -10	; vyvolání smyčky WHILE za následujících podmínek: Aktuální požadovaná hodnota ve WCS pro osu vrtáku musí být větší než -10.
G1 G91 F250 AX[BOHRACHSE] = -1	
ENDWHILE	

Programový kód	Komentář
...	

4.1.7.5 Programová smyčka s podmínkou na konci smyčky (REPEAT, UNTIL)

U smyčky REPEAT se podmínka nachází na konci smyčky. Smyčka REPEAT je tedy jednou zpracována a potom je opakována tak dlouho, dokud je podmínka splněna.

Syntaxe

```
REPEAT
...
UNTIL <podmínka>
```

Význam

REPEAT:	Uvádí programovou smyčku.
UNTIL:	Označuje konec smyčky a způsobuje návrat zpět na její začátek.
<podmínka>:	Podmínka, která musí být splněna, aby se smyčkou REPEAT už neprocházelo.

Příklad:

Programový kód	Komentář
...	
REPEAT	; vyvolání smyčky REPEAT
...	
UNTIL ...	; kontrola, zda je podmínka splněna
...	

4.1.7.6 Příklad programu se vnořenými řídícími strukturami

Programový kód	Komentář
LOOP	
IF NOT $P_SEARCH	; IF žádné vyhledávání bloku
G1 G90 X0 Z10 F1000	
WHILE $AA_IM[X] <= 100	; WHILE (požadovaná hodnota osy X <= 100)
G1 G91 X10 F500	; Vrtací vzor
Z-5 F100	
Z5	
ENDWHILE	
ELSE	; ELSE vyhledávání bloku
MSG("Im Suchlauf wird nicht gebohrt")	

Programový kód	Komentář
ENDIF	; ENDIF
$A_OUT[1] = 1	; následující vrtaná deska
G4 F2	
ENDLOOP	
M30	

4.1.8 Koordinace programů přesahující hranice kanálu (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM)

V kanálu NC systému může být v principu zpracováván program, který je v něm spuštěn, a to nezávisle na jiných kanálech jeho skupiny provozních režimů (BAG). Pokud se ale na opracování obrobku podílí více programů ve více kanálech jedné BAG současně, musí být zpracování těchto programů v jednotlivých kanálech koordinováno pomocí následujících koordinačních příkazů.

Předpoklady

Všechny kanály, které se na koordinaci programů podílejí, musí patřit do **stejné** skupiny provozních režimů (BAG).

MD10010 $MC_ASSIGN_CHAN_TO_MODE_GROUP[<kanál>] = <číslo BAG>

Název kanálu místo čísla kanálu

Jako parametr v těchto předem definovaných procedurách koordinace programů se mohou namísto čísel kanálů používat také názvy kanálů, které jsou zadány v parametru MD20000 $MC_CHAN_NAME[<index kanálu>]. Používání názvů kanálů v NC programech musí být napřed odblokováno:

MD10280 $MN_PROG_FUNCTION_MASK, Bit 1 = TRUE

Poznámka

Minimální vzdálenost mezi příkazy

Mezi příkazy `INIT`, `START`, `WAITE`, `WAITM`, `SETM`, `CLEARM` a příkazem `WAITMC` musí být zachovávána vzdálenost minimálně dva bloky s příkazy pohybu. Příkaz WIATMC je zpracovatelný blok, který se ale kvůli optimalizaci přesouvá do předcházejícího bloku a který je potom jako blok vymazán. Například příkaz `SETM` zpracovatelným blokem není a přesouvá se do následujícího bloku, takže pokud by vzdálenost byla jen jeden blok, oba příkazy by se ocitly v prostředním bloku. Protože je možný jen jeden blok, v případě, že by pro příkaz `WAITMC` byla vzdálenost jeden blok, optimalizace se neuskuteční.

Program se tím zabrzdí a zpracování se na krátkou chvíli pozastaví.

Syntaxe

```
INIT(<ChanNr>, <Prog>, <AckMode>)
START(<ChanNr>, <ChanNr>, ...)
WAITM(<MarkNr>, <ChanNr>, <ChanNr>, ...)
WAITE(<ChanNr>, <ChanNr>, ...)
```

```
WAITMC(<MarkNr>, <ChanNr>, <ChanNr>, ...)
SETM(<MarkNr>, <MarkNr>, ...)
CLEARM(<MarkNr>, <MarkNr>, ...)
```

Význam

`INIT()`:	Předem definovaná procedura v zvolení NC programu, který má být zpracováván v uvedeném kanálu.		
`START()`:	Předdefinovaná procedura pro spouštění programu vybraného v příslušném kanálu		
`WAITM()`:	Předem definovaná procedura pro čekání na dosažení značky pro čekání v uvedených kanálech Ve vlastním kanálu se uvedená značka pro čekání nastavuje příkazem `WAITM`. Předcházející blok se ukončí s přesným najetím. Po synchronizaci se značka pro čekání vymaže. Současně může být nastaveno max. 10 značek na každý kanál.		
`WAITE()`:	Předem definovaná procedura pro čekání na konec programu v jednom nebo ve více jiných kanálech		
`WAITMC()`:[1)]	Předem definovaná procedura pro čekání na dosažení značky pro čekání v uvedených kanálech Oproti funkci `WAITM` se zabrždění os na přesné najetí uskuteční jen tehdy, pokud v jiném kanálu nebylo značky pro čekání dosud dosaženo.		
`SETM()`:[1)]	Předem definovaná procedura pro nastavení jedné nebo více značek pro čekání pro koordinaci kanálů Zpracování ve vlastním kanálu není touto funkcí nijak ovlivněno. Příkaz `SETM` zůstává v platnosti, i když je aktivován reset kanálu a NC-Start.		
`CLEARM()`:[1)]	Předem definovaná procedura pro vymazání jedné nebo více značek pro čekání pro koordinaci kanálů Zpracování ve vlastním kanálu není touto funkcí nijak ovlivněno. Příkaz `CLEARM()` vymaže všechny značky pro čekání v kanálu. Příkaz `CLEARM(0)` vymaže pouze značku pro čekání „0“. Příkaz `CLEARM` zůstává v platnosti, i když je aktivován reset kanálu a NC-Start.		
`<ChanNr>`:	Číslo kanálu Číslo vlastního kanálu se uvádět nemusí.		
	Typ:	INT	
`<Prog>`:	Absolutní nebo relativní zadání cesty (**není nutné**) + název programu		
	Typ:	STRING	
	Pokud budete potřebovat informace týkající se zadávání cesty, nahlédněte do příručky o programování NC systémů, kapitola „Adresování souborů v paměti pro programy“ (Strana 559).		
`<AckMode>`:	Režim potvrzování (**volitelně**)		
	Typ:	CHAR	
	Hodnoty:	"N"	Bez potvrzení Zpracování programu se uskutečňuje po odeslání příkazu. Odesílatel nebude nijak informován, pokud příkaz nebylo možné úspěšně provést.
		"S"	Synchronní potvrzování Zpracovávání programu bude pozastaveno tak dlouho, dokud komponent příjemce nepotvrdí příkaz. Při pozitivním potvrzení bude zpracován následující příkaz. Při negativním potvrzení se vypíše chybové hlášení.

<MarkNr>:	Číslo značky, na kterou se čeká. **Poznámka** V případě systému s více kanály je k dispozici maximálně 100 značek pro čekání (značka pro čekání 0 ... 99). V systému, který má jen jeden kanál, je k dispozici pouze značka pro čekání 0.
1) Pro potřeby specifické uživatelské komunikace a/nebo koordinace kanálů mohou být značky pro čekání ovládané příkazy SETM / CLEARM používány také bez příkazů podmíněného čekání WAITMC. Značky pro čekání si přitom ponechávají své hodnoty i po resetu kanálu a operaci NC-Start.	

Příklady

Příkaz START prostřednictvím názvů kanálů z parametru MD20000

- Nastavení parametrů
```
MD10280 $MN_PROG_FUNCTION_MASK, Bit 1 = TRUE
$MC_CHAN_NAME[ 0 ] = "BEARBEITUNG" ; název pro kanál 1
$MC_CHAN_NAME[ 1 ] = "ZUFUEHRUNG" ; název pro kanál 2
```
- Programování

Programový kód	Komentář
START(BEARBEITUNG)	; spuštění kanálu 1
START(ZUFUEHRUNG)	; spuštění kanálu 2

Spouštění prostřednictvím lokálních „názvů kanálů" a uživatelské proměnné

Programový kód	Komentář
DEF INT MASCHINE = 1	; definice uživatelské proměnné pro kanál 1
DEF INT LADER = 2	; definice uživatelské proměnné pro kanál 2
...	
START(MASCHINE)	; spuštění kanálu 1
START(LADER)	; spuštění kanálu 2

Spouštění prostřednictvím lokálních „názvů kanálů", uživatelské proměnné a názvů kanálů zadaných v parametrech

Programový kód	Komentář
DEF INT chanNo1	; definice uživatelské proměnné pro kanál 1
DEF INT chanNo2	; definice uživatelské proměnné pro kanál 2
chanNo1 = CHAN_1	; přiřazení názvu kanálu 1 nastaveného v parametrech
chanNo2 = CHAN_2	; přiřazení názvu kanálu 2 nastaveného v parametrech
...	
START(chanNo1)	; spuštění kanálu 1
START(chanNo2)	; spuštění kanálu 2

Příkaz INIT s absolutním udáním cesty

Deaktivování programu /_N_MPF_DIR/_N_ABSPAN1_MPF v kanálu 2.

Programový kód

```
INIT(2,"/_N_WKS_DIR/_N_WELLE1_WPD/_N_ABSPAN1_MPF")
```

Příkaz INIT s názvem programu

Zvolení programu s názvem „MYPROG". Řídící systém vyhledá program na základě cesty pro vyhledávání.

Programový kód

```
INIT(2,"MYPROG")
```

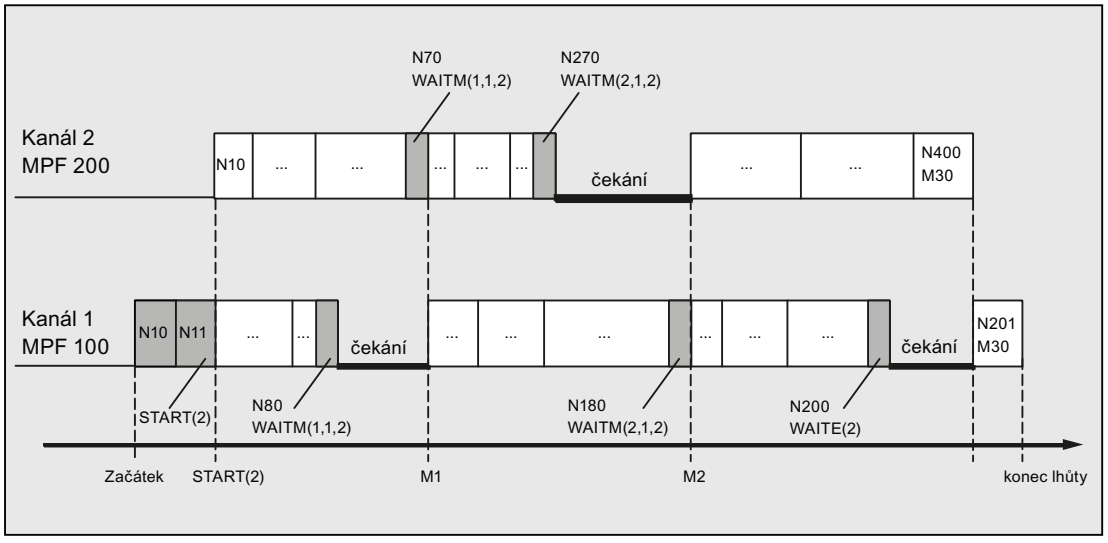
Koordinace programů pomocí funkce WAITM

- Kanál 1: Program /_N_MPF_DIR/_N_MPF100_MPF je již vybrán a je i spuštěn.

Programový kód	**Komentář**
	; program MPF100
N10 INIT(2,"MPF200","N")	; vyvolání programu MPF200, kanál 2
N11 START(2)	; spuštění kanálu 2
...	
N80 WAITM(1,1,2)	; čekání na značku WAIT 1 v kanálu 1 a 2
N81 ...	; kanál 1, N81, a kanál 2, N71, začnou ; synchronizovaně
...	
N180 WAITM(2,1,2)	; čekání na značku WAIT 2 v kanálu 1 a 2
N181 ...	; kanál 1, N181, a kanál 2, N271, začnou ; synchronizovaně
...	
N200 WAITE(2)	; čekání na konec programu v kanálu 2
N201 ...	; zpracování N201 se spustí teprve po skončení programu ; MPF200 v kanálu 2
N201 M30	; konec programu v kanálu 1

- Kanál 2: V kanálu 1 se prostřednictvím bloků N10 a N20 vyvolává a spouští program MPF200_MPF pro kanál 2.

Programový kód	**Komentář**
;$PATH=/_N_MPF_DIR	; Program MPF200
...	
N70 WAITM(1,1,2)	Čekání na značku WAIT 1 v kanálu 1 a 2
N71 ...	; kanál 1, N81, a kanál 2, N71, začnou ; synchronizovaně
...	
N270 WAITM(2,1,2)	Čekání na značku WAIT 2 v kanálu 1 a 2
N271 ...	; kanál 1, N181, a kanál 2, N271, začnou ; synchronizovaně
...	
N400 M30	Konec programu v kanálu 2



Okrajové podmínky

Nesynchronizované zahájení zpracování bloků následujících po značce WAIT

V případě koordinování kanálů pomocí značek WAIT může dojít k nesynchronizovanému spuštění zpracování následujících bloků. Toto chování se vyskytne tehdy, jestliže je bezprostředně před dosažením společné značky WAIT v jednom z kanálů, jež mají být synchronizovány, spuštěna akce, která má v tomto případě za následek vymazání zbytkové dráhy a implicitní nastavení původní polohy (REPOSA).

Předpoklad: Aktuální přiřazení os v kanálech 1 a 2

- Kanál 1: Osy X1 a U
- Kanál 2: Osa X2

Tabulka 4-2 Časový průběh v kanálech 1 a 2

Kanál 1	Kanál 2	Popis
...	...	Libovolné zpracování v kanálu 1 a 2
`N100 WAITM(20,1,2)`		Kanál 1: dosáhne značky `WAIT` a čeká na synchronní akci v kanálu 2
Začátek zpracování příkazu GETD(U): • *Výměna osy* • *Vymazání zbytkové dráhy* • *REPOSA* *Konec*	`N200 GETD(U)`	Kanál 2: vyžádá osu U z kanálu 1 Kanál 1: Zpracování příkazu `GET(U)` na pozadí
	`N210 WAITM(20,1,2)`	Kanál 2: Dosáhne značky `WAIT`. ⇒ Synchronizace kanálů 1 a 2 je tím ukončena.
	`N220 G0 X2=100`	Kanál 2: Zahájení zpracování od bloku N220
`N110 G0 X1=100`		Kanál 1: Časově posunuté zahájení zpracování od bloku N110

4.1.9 Technika maker (DEFINE ... AS)

UPOZORNĚNÍ
Technika maker zvyšuje komplexnost programování. Pomocí techniky maker je možné programovací jazyk řídícího systému silně změnit. Technika maker se smí používat jen s velkou opatrností.

Makro přestavuje posloupnost jednotlivých příkazů, kterou jsou soustředěny do nového celku, který má vlastní název. Při vyvolání makra v průběhu zpracování programu jsou postupně zpracovány příkazy, které byly pod název makra naprogramovány.

V závislosti na oblasti jejich platnosti, tzn. oblasti, ve které definice makra platí, existují následující kategorie maker:

- Lokální makra
  Lokální makra jsou makra, která jsou definována na začátku NC programu, který v okamžiku zpracovávání není hlavním programem. Jsou založeny při vyvolávání NC programu a jsou vymazány s resetem na konci programu, příp. při následujícím spouštění řídícího systému. Přístup k lokálním makrům je možný pouze v rámci NC programu, v němž jsou definovány.
- Makra globální v daném programu
  Programově globální makra jsou makra, která jsou definována na začátku NC programu, který se používá jako hlavní program. Jsou založeny při vyvolávání NC programu a jsou vymazány s resetem na konci programu, příp. při následujícím spouštění řídícího systému. Přístup k programově globálním makrům je k dispozici v hlavním programu a ve všech podprogramech.

  Poznámka

  Dostupnost programových globálních maker

  Programově globální makra definovaná v hlavním programu jsou k dispozici i v podprogramech jen tehdy, pokud je nastaven následující strojní parametr:

  MD11120 $MN_LUD_EXTENDED_SCOPE = 1

  Pokud je MD11120 = 0, jsou programově globální makra definovaná v hlavním programu k dispozici pouze v tomto hlavním programu.

- Globální makra
  Globální makra jsou globálními makry v NC systému, příp. v kanálu, jsou definována v definičním modulu (soubor maker) a zůstávají zachována i po resetu na konci programu, příp. po následujícím spuštění řídícího systému. Globální makra je potom možné vyvolat v libovolném hlavním programu nebo podprogramu a nechat jej tak zpracovat.

  Poznámka

  Aby bylo možné používat makra z **externího** souboru maker v NC-programu, musí být soubor maker načten do NC systému.

Makra musí být před svým použitím definována. Přitom je potřeba dbát na následující pravidla:

- V makru mohou být definovány libovolné identifikátory, G-, M-, H-funkce a L-názvy podprogramů.
- Definice maker se může nacházet na začátku programu nebo ve vlastním definičním souboru (soubor maker).
- Lokální a programově globální makra jsou definována na začátku programu.
- Globální makra musí být definována v souboru maker, např. _N_DEF_DIR/_N_UMAC_DEF.
- Příkazy G-funkcí maker mohou být definovány jen jako globální makra.
- H- a L-funkce mohou být naprogramovány jako 2-místné.
- Příkazy M- a G-funkcí mohou být naprogramovány jako 3-místné.

Poznámka

Definice klíčových slov a rezervované názvy nesmí být pomocí maker přepisovány. To platí také pro všechna návěští skoků v rámci příkazu GOTO a také pro klíčová slova v programových smyčkách, jako jsou struktury tvořené příkazy FOR, WHILE, LOOP, REPEAT.

Syntaxe

Definice makra:
```
DEFINE <název makra> AS <operace_1> <operace_2> ...
```

Volání v NC programu:
```
<název makra>
```

Význam

`DEFINE ... AS`:	Kombinace klíčových slov pro definici makra
`<název makra>`:	Název makra Jako názvy maker jsou přípustné pouze identifikátory. Pomocí názvu makra se potom v NC programu makro vyvolává.
`<Operace_1>`:	První programový příkaz v makru
`<Operace_2>`:	Druhý programový příkaz v makru

Příklady

Příklad 1: Definice makra na začátku programu

Programový kód	**Komentář**
`DEFINE LINIE AS G1 G94 F300`	`; Definice makra`
`...`	
`N70 LINIE X10 Y20`	`; Volání makra`
`...`	

Příklad 2: Definice maker v souboru makra

Programový kód	Komentář
DEFINE M6 AS L6	; Při výměně nástroje se bude vyvolávat podprogram, který převezme přenos nezbytných dat. V podprogramu se spouští M-funkce pro vlastní výměna nástroje (např. M106).
DEFINE G81 AS DRILL(81)	; kopírování příkazů G-funkcí podle normy DIN
DEFINE G33 AS M333 G333	; Při řezání závitu je požadována synchronizace s PLC. Předcházející příkaz G-funkce G33 byl pomocí strojního parametru přejmenován na G333, programování ale zůstává pro uživatele stejné.

Příklad 3: Externí soubor maker

Po načtení externího souboru maker do řídícího systému musí být soubor maker načten do NC systému. Teprve potom mohou být makra v NC-programech používána.

Programový kód	Komentář
%_N_UMAC_DEF	
;$PATH=/_N_DEF_DIR	; Specifická uživatelská makra
DEFINE PI AS 3.14	
DEFINE TC1 AS M3 S1000	
DEFINE M13 AS M3 M7	; Vřeteno se otáčí vpravo, zapnutí chladicí kapaliny
DEFINE M14 AS M4 M7	; Vřeteno se otáčí vlevo, zapnutí chladicí kapaliny
DEFINE M15 AS M5 M9	; Zastavení vřetena, vypnutí chladicí kapaliny
DEFINE M6 AS L6	; Vyvolání programu pro výměnu nástroje
DEFINE G80 AS MCALL	; Deaktivování cyklu pro vrtání
M30	

4.2 Technika podprogramů

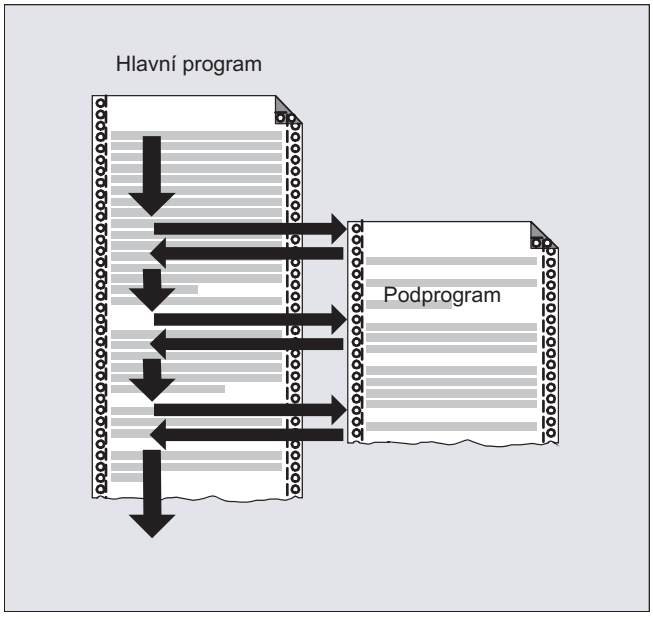
4.2.1 Všeobecně

4.2.1.1 Podprogram

Označení podprogram pochází ještě z dob, kdy byly výrobní programy pevně rozděleny na hlavní programy a podprogramy. Hlavní programy přitom byly výrobní programy, které se v řídícím systému vybraly ke zpracování a které s potom spouštěly. Podprogramy bývaly výrobní programy, které bývaly vyvolávány z hlavního programu.

Toto pevné rozdělení v dnešním jazyku systému SINUMERIK už neexistuje. V principu může být každý výrobní program vybrán a spuštěn jako program hlavní nebo může být vyvolán z jiného výrobního programu jako podprogram.

Proto jsou v dalším popisu jako podprogramy označovány ty výrobní programy, které jsou vyvolávány z jiného výrobního programu.



Aplikace

Stejně jako u všech vyšších programovacích jazyků se i v jazyku NC systému používají podprogramy k tomu, aby se části programu, které se mají použít vícekrát, uložily do samostatného uzavřeného programu.

Podprogram nabízí následující výhody:

- Výrazně vyšší přehlednost a čitelnost programu
- Zlepšení jakosti díky opětovnému použití testovaných částí programu
- Nabízí možnost vytvořit si specifické knihovny obráběcích programů
- Úspora místa v paměti

4.2.1.2 Názvy podprogramů

Pravidla pro sestavování názvů

Název podprogramu může být libovolný, musí však být dodržena následující pravidla:

- Přípustné znaky:
  - Písmena: A ... Z, a ... z
  - Číslice: 0 ... 9
  - Znak podtržení: _
- První dva znaky by měly být dvě písmena nebo znak podtržení, za kterým následuje písmeno.

  Poznámka

  Jestliže jsou tyto podmínky splněny, potom může být tento NC program samotný vyvoláván jako podprogram z nějakého jiného programu zadáním tohoto názvu programu. Jestliže název programu proti tomu začíná číslicemi, potom je volání podprogramu možné jedině pomocí příkazu CALL.

- Maximální délka: 24 znaků

Poznámka

Velká/malá písmena

V jazyku NC systému SINUMERIK **nejsou** rozlišována velká a malá písmena.

Poznámka

Nepřípustné názvy programů

Aby se zabránilo problémům s aplikacemi Windows, následující názvy programů se **nesmějí** používat:

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

Interní přípony řídícího systému

Název podprogramu zadaný při jeho sestavování je uvnitř řídícího systému rozšířen o předponu a o příponu:

- Počáteční řetězec: `_N_`
- Koncový řetězec: `_SPF`

Použití názvu programu

Když je název programu použit např. při volání podprogramu, jsou možné všechny kombinace předpony, názvu programu a přípony.

Příklad:

Podprogram s názvem programu `SUB_PROG` může být spuštěn pomocí následujících identifikátorů:

1. `SUB_PROG`
2. `_N_SUB_PROG`
3. `SUB_PROG_SPF`
4. `_N_SUB_PROG_SPF`

Hlavní program a podprogram stejného názvu

Jestliže existuje hlavní program (.MPF) a podprogram (.SPF) se stejným názvem programu, musí se při používání názvů těchto programů v NC programu zadávat také odpovídající přípona souboru, aby byla zajištěna jejich jednoznačná identifikace. Jinak se použije program, který bude při procházení podle cesty pro vyhledávání nalezen jako první objekt s uvedeným názvem programu.

Název souboru a název programu

Název souboru a název programu, který je v tomto souboru obsažen, musí být vždy identické.

Soubor proto vždy obsahuje jeden program.

4.2.1.3 Vnoření podprogramů

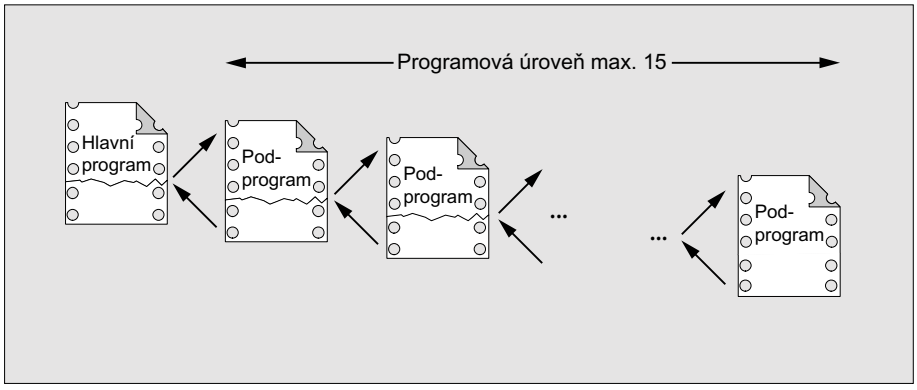
Hlavní program může vyvolávat podprogramy, které mohou opět vyvolávat další podprogramy. Zpracování programů takto může být postupně do sebe vnořeno. Každý program přitom běží na své vlastní programové úrovni.

Hloubka vnoření

Jazyk NC systému má v současnosti k dispozici 16 programových úrovní. Hlavní program vždy běží na nejvyšší programové úrovni 0. Podprogram vždy běží na nejbližší nižší programové úrovni, která následuje za úrovní, z níž byl vyvolán. Programová úroveň 1 je tedy první úrovní pro podprogramy.

Rozdělení programových úrovní:

- Programová úroveň 0: Úroveň hlavního programu
- Programová úroveň 1 - 15: Úroveň podprogramu 1 - 15



Rutiny přerušení (ASUP)

Jestliže je v rámci rutiny přerušení vyvoláván podprogram, nebude zpracováván na programové úrovni, která je momentálně v kanálu aktivní (n), nýbrž rovněž na nejbližší nižší programové úrovni (n+1). By byla tato operace možná i na nejnižší programové úrovni, jsou v souvislosti s rutinami přerušení k dispozici ještě další 2 programové úrovně (16 a 17).

Jestliže jsou zapotřebí více než 2 programové úrovně, je nutné na to explicitně brát zřetel při návrhu struktury výrobního programu, který má být v kanálu zpracováván. To znamená, že se smí počítat jen s maximálně tolika programovými úrovněmi, aby byl pro zpracování přerušení k dispozici dostatečný počet programových úrovní.

Jestliže tedy zpracování přerušení potřebuje např. 4 programové úrovně, musí být výrobní program strukturován tak, aby se obsazovala maximálně 13. programová úroveň. Pokud se potom vyskytne přerušení, budou pro ně potřebné 4 programové úrovně (14 až 17) k dispozici.

Cykly firmy Siemens

Cykly firmy Siemens potřebují 3 programové úrovně. Volání cyklů firmy Siemens se proto musí uskutečňovat maximálně:

- zpracovávání výrobního programu: Programová úroveň 12
- Rutina přerušení: Programová úroveň 14

4.2.1.4 Cesta pro vyhledávání

Při volání podprogramu bez udání cesty řídící systém prohledává paměti pro programy, které jsou k dispozici, podle předem definované posloupnosti při vyhledávání (viz "Cesta pro vyhledávání při volání podprogramu (Strana 564)").

4.2.1.5 Formální a skutečný parametr

O formálních a skutečných parametrech se mluví v souvislosti s definicí a s vyvoláváním podprogramů s předávanými parametry.

Formální parametr

Při definici podprogramu musí být definovány také parametry, které mají být podprogramu předány, což jsou tak zvané formální parametry, spolu s jejich typem a s názvy.

Formální parametry tak definují rozhraní podprogramu.

Příklad:

Programový kód	Komentář
`PROC KONTUR (REAL X, REAL Y)`	`; formální parametry: X a Y, oba typu REAL`
`N20 X1=X Y1=Y`	`; najíždění osou X1 na pozici X a osou Y1 na pozici Y`
`...`	
`N100 RET`	

Skutečné parametry

Podprogramu musí být při jeho volání předány absolutní hodnoty nebo proměnné, což jsou tak zvané skutečné parametry.

Skutečný parametr takto při volání doplní do rozhraní podprogramu skutečné hodnoty.

Příklad:

Programový kód	Komentář
`N10 DEF REAL BREITE`	`; definice proměnných`
`N20 BREITE=20.0`	`; přiřazení hodnoty proměnné`
`N30 KONTUR(5.5, BREITE)`	`; volání podprogramu se skutečnými parametry: 5.5 a BREITE`
`...`	
`N100 M30`	

4.2.1.6 Předávání parametrů

Definice podprogramu s předáváním parametrů

Definice podprogramu s předáváním parametrů se uskutečňuje pomocí klíčového slova `PROC` a výpisem úplného seznamu všech podprogramem očekávaných parametrů.

Neúplné předávání parametrů

Při volání podprogramu nemusí být explicitně předávány vždy všechny parametry, které jsou v rozhraní podprogramu definovány. Pokud je nějaký parametr vypuštěn, bude se pro tento parametr předávat standardní hodnota "0".

Aby ale byla posloupnost parametrů jednoznačně identifikována, musí být vždy zapsány čárky, které slouží jako oddělovací znak pro parametry. Výjimku tvoří poslední parametry. Pokud jsou při volání vypuštěny, mohou odpadnout i poslední čárky.

Příklad:

Podprogram:

Programový kód	**Komentář**
`PROC SUB_PROG (REAL X, REAL Y, REAL Z)`	`; formální parametry: X, Y a Z`
`...`	
`N100 RET`	

Hlavní program:

Programový kód	**Komentář**
`PROC MAIN_PROG`	
`...`	
`N30 SUB_PROG(1.0,2.0,3.0)`	`; volání podprogramu s úplným předáváním parametrů:`
	`X=1.0, Y=2.0, Z=3.0`
`…`	
`N100 M30`	

Příklady pro volání podprogramu v bloku `N30` s neúplným předáváním parametrů:

```
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
```

UPOZORNĚNÍ

Předávání parametrů typu Call-By-Reference

Parametry, které jsou předávány prostřednictvím Call-by-Reference, nesmí být při volání podprogramu vypuštěny.

UPOZORNĚNÍ

Datový typ AXIS

Parametry datového typu AXIS nesmí být při volání podprogramu vypuštěny.

Kontrola předávaných parametrů

Prostřednictvím systémové proměnné $P_SUBPAR [n], kde n = 1, 2, ..., může být v podprogramu zkontrolováno, zda se parametr explicitně předává nebo zda byl vypuštěn. Index n se vztahuje na posloupnost formálních parametrů. Index n = 1 se vztahuje na 1. formální parametr, index n = 2 na 2. formální parametr atd.

Následující úsek programu ukazuje na příkladu 1. formálního parametru, jak může být toto přezkoumání realizováno:

Programování	**Komentář**
PROC SUB_PROG (REAL X, REAL Y, REAL Z)	; formální parametry: X, Y a Z
N20 IF $P_SUBPAR[1]==TRUE	; přezkoumání 1. formálního parametru X
...	; tyto operace se uskuteční, jestliže byl formální parametr X explicitně předán
N40 ELSE	
...	; tyto operace se uskuteční, jestliže formální parametr X nebyl předán
N60 ENDIF	
...	; všeobecné operace
N100 RET	

4.2.2 Definice podprogramu

4.2.2.1 Podprogram bez předávání parametrů

Při definici podprogramů bez předávání parametrů může definiční řádek na začátku programu odpadnout.

Syntaxe

```
[PROC <název programu>]
...
```

Význam

PROC:	Definiční příkaz na začátku programu
<název programu>:	Název programu

Příklad:

Příklad 1: Podprogram s příkazem PROC

Programový kód	**Komentář**
PROC SUB_PROG	; definiční řádek
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	; skok zpátky z podprogramu

Příklad 2: Podprogram bez příkazu PROC

Programový kód	**Komentář**
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	; skok zpátky z podprogramu

Viz také

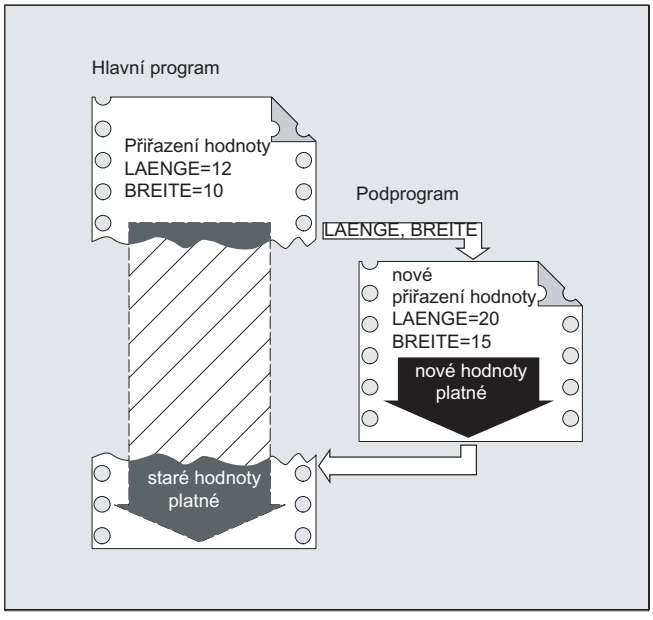
Volání podprogramu bez předávání parametrů (Strana 526)

4.2.2.2 Podprogram s předáváním parametrů Call-by-Value (PROC)

Definice podprogramu s předáváním parametrů Call-by-Value se uskutečňuje pomocí klíčového slova PROC, za kterým následuje název programu a výpis úplného seznamu všech parametrů spolu s jejich typy a názvy. Definiční příkaz se musí nacházet na prvním řádku programu.

Call-by-Value (Volání hodnotou)

Při předávání parametrů typu Call-By-Value volající program přenáší do podprogramu pouze hodnotu proměnné. Podprogram v důsledku toho nemá k proměnné žádný přímý přístup. Jestliže se hodnota parametru změní, změní se jen hodnota, která je viditelná v podprogramu. Hodnota proměnné definovaná ve volajícím programu zůstává nezměněna. Předávání parametrů typu Call-by-Value proto nemá žádný zpětný vliv na volající program.



Syntaxe

```
PROC <název programu> (<typ parametru> <název parametru>=<počáteční
hodnota>, ...)
```

Poznámka

Může být předáváno až 127 parametrů.

Význam

PROC:	Definiční příkaz na začátku programu
<název programu>:	Název programu
<typ parametru>:	Datový typ parametru (např. REAL, INT, BOOL)
<název parametru>:	Název parametru
<počáteční hodnota>:	Volitelná hodnota pro inicializaci parametru (**nepovinné**) Jestliže při volání podprogramu není uveden žádný parametr, obsahuje tento parametr inicializační hodnotu, která je mu přiřazena.

Příklady

Příklad 1

Definice podprogramu SUB_PROG se třemi parametry typu REAL s předdefinovanými hodnotami:

Programový kód

```
PROC SUB_PROG(REAL LENGTH=10.0, REAL WIDTH=20.0, REAL HIGHT=30.0)
```

Příklad 2

Různé varianty volání

Programový kód

```
PROC MAIN_PROG
  REAL PAR_1 = 100
  REAL PAR_2 = 200
  REAL PAR_3 = 300
  ; varianty volání
  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
```

Viz také

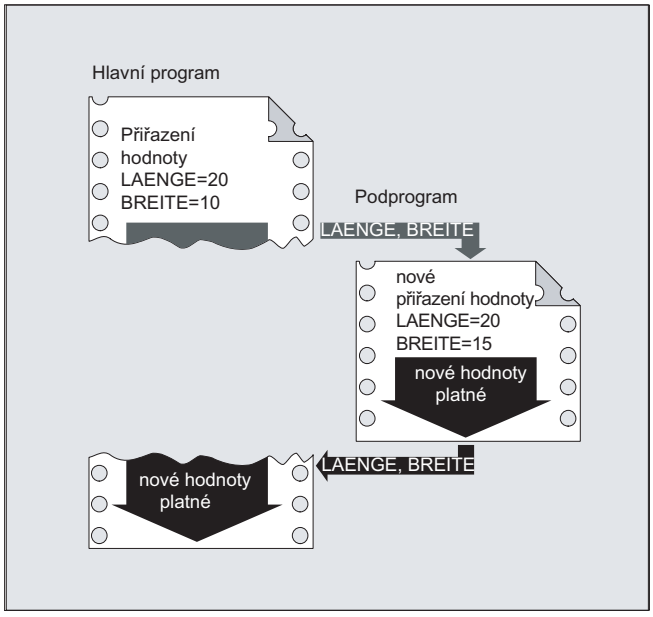
Volání podprogramu s předáváním parametrů (EXTERN) (Strana 528)

4.2.2.3 Podprogram s předáváním parametrů Call-by-Reference (PROC, VAR)

Definice podprogramu s předáváním parametrů Call-by-Reference se uskutečňuje pomocí klíčového slova `PROC`, za kterým následuje název programu a výpis úplného seznamu všech parametrů s klíčovým slovem `VAR` a jejich typy a názvy. Definiční příkaz se musí nacházet na prvním řádku programu. Jako parametr mohou být předávány také odkazy na pole.

Call-by-Reference (Volání odkazem)

Při předávání parametrů typu Call-By-Reference volající program přenáší do podprogramu ne hodnotu proměnné, ale odkaz (ukazatele) na příslušnou proměnnou. Podprogram v důsledku toho nemá k proměnné žádný přímý přístup. Jestliže se hodnota parametru změní, nezmění se jen hodnota, která je viditelná v podprogramu, ale změní se i hodnota proměnné definovaná ve volajícím programu. Předávání parametrů typu Call-by-Reference má tedy i po skončení podprogramu zpětný vliv na volající program.



Poznámka

Předávání parametrů typu Call-by-Reference je zapotřebí jen tehdy, pokud je předávaná proměnná ve volajícím programu lokálně definována (LUD). Proměnné, které jsou globální v kanálu nebo v NC systému, nemusí být předávány, protože podprogram k nim může mít přímý přístup.

Syntaxe

```
PROC <název programu> (VAR <typ parametru> <název parametru>, ...)
PROC <název programu> (VAR <typ pole> <název pole>
[<m>,<n>,<o>], ...)
```

Poznámka

Může být předáváno až 127 parametrů.

Význam

PROC:	Definiční příkaz na začátku programu	
VAR:	Klíčové slovo pro předávání parametru typu Call-by-Reference	
<název programu>:	Název programu	
<typ parametru>:	Datový typ parametru (např. REAL, INT, BOOL)	
<název parametru>:	Název parametru	
<typ pole>:	Datový typ prvku pole (např. REAL, INT, BOOL)	
<název pole>:	Název pole	
[<m>,<n>,<o>]:	Velikost pole V současnosti je možné pracovat s maximálně 3-rozměrnými poli:	
	<m>:	Velikost pole pro 1. rozměr
	<n>:	Velikost pole pro 2. rozměr
	<o>:	Velikost pole pro 3. rozměr

Poznámka

- Název programu zadaný po klíčovém slově PROC se musí shodovat s názvem programu, který je uveden na uživatelském rozhraní.
- Pomocí polí s neurčenou délkou fungujícími jako předávané parametry mohou být podprogramem zpracovávána pole proměnné délky. Za tím účelem se při definici např. dvourozměrného pole jako formální parametr délka v 1. rozměru nezadává. Čárka ale musí být zapsána.
  Příklad: PROC <název programu> (VAR REAL FELD[,5])

Příklad:

Definice podprogramu se dvěma parametry Call-by-Reference typu REAL:

Programový kód

```
; parametr 1: odkaz na typ: REAL, název: LAENGE
; parametr 2: odkaz na typ: REAL, název: BREITE
PROC SUB_PROG(VAR REAL LAENGE, VAR REAL BREITE)
```

Viz také

Volání podprogramu s předáváním parametrů (EXTERN) (Strana 528)

4.2.2.4 Ukládání modálních G-funkcí (SAVE)

Atribut SAVE způsobuje, že všechny příkazy modálních G-funkcí, které byly před voláním podprogramu aktivní, se uloží a po ukončení podprogramu se znovu aktivují.

UPOZORNĚNÍ
Přerušení režimu řízení pohybu po dráze Jestliže je v době aktivního režimu řízení pohybu po dráze vyvolán podprogram s atributem SAVE, režim řízení pohybu po dráze se po skončení podprogramu (skok zpět) přeruší.

Syntaxe

```
PROC <název podprogramu> SAVE
```

Význam

SAVE:	Uložení stavu příkazů modálních G-funkcí před voláním podprogramu a jejich opětovné nastavení po skončení podprogramu.

Příklad:

V podprogramu KONTUR je aktivní příkaz modální G-funkce G91 (inkrementální rozměry). V hlavním programu je aktivní příkaz modální G-funkce G90 (absolutní rozměry). Když bude v definici podprogramu zadán příznak SAVE, po skončení podprogramu bude v hlavním programu znovu v platnosti G90.

Definice podprogramu:

Programový kód	**Komentář**
PROC KONTUR (REAL WERT1) SAVE	; definice podprogramu s parametrem SAVE
N10 G91 ...	; příkaz modální G-funkce G91: Řetězové kótování
N100 M17	; konec podprogramu

Hlavní program:

Programový kód	**Komentář**
N10 G0 X... Y... G90	; příkaz modální G-funkce G90: Absolutní rozměry
N20 ...	
...	
N50 KONTUR (12.4)	; volání podprogramu
N60 X... Y...	; příkaz modální G-funkce G90 je díky příkazu SAVE znovu aktivován

Okrajové podmínky

Framy

Chování framů v souvislosti s podprogramy s atributem SAVE závisí na typu framu a může být nastaveno pomocí strojních parametrů.

4.2.2.5 Potlačení zpracovávání blok po bloku (SBLOF, SBLON)

Když je aktivován režim zpracování blok po bloku, má uživatel možnost zpracovat NC program úplně nebo i částečně bez zastavování. Režim zpracovávání blok po bloku je možné potlačit příkazem SBLOF, pro opětovné aktivování slouží příkaz SBLON.

Potlačení zpracování blok po bloku pro celý NC program

Jestliže je deaktivování zpracování blok po bloku (SBLOF) naprogramováno na prvním řádku (PROC ...) **hlavního programu**, platí až do konce nebo přerušení daného NC programu. NC program potom při zpracování blok po bloku proběhne bez pozastavení.

Jestliže je deaktivování zpracování blok po bloku (SBLOF) naprogramováno na prvním řádku (PROC ...) **podprogramu**, platí až do konce nebo přerušení daného NC programu. Pomocí naprogramovaného příkazu pro návrat se rozhone, zda se má nebo nemá na konci podprogramu zpracování pozastavit:

- Návrat s příkazem M17: Zastavení na konci podprogramu
- Návrat s příkazem RET: **Žádné** zastavení na konci podprogramu

Potlačení zpracování blok po bloku v rámci programu

Jestliže je deaktivování zpracování blok po bloku (SBLOF) naprogramováno v nějakém bloku v rámci NC programu, bude zpracovávání blok po bloku deaktivováno od tohoto bloku až po následující naprogramované aktivování režimu zpracovávání blok po bloku (SBLON) nebo do skončení aktivní úrovně podprogramu.

Syntaxe

Potlačení zpracování blok po bloku pro celý NC program:

```
PROC ... SBLOF
...
```

Potlačení zpracování blok po bloku v rámci programu:

```
...
SBLOF
...
SBLON
...
```

Význam

PROC:	První příkaz programu	
SBLOF:	Předem definovaná procedura pro deaktivování zpracování blok po bloku	
	Samostatný blok:	ano, v bloku PROC možné
	Platnost:	modální
SBLON:	Předem definovaná procedura pro aktivování zpracování blok po bloku	
	Samostatný blok:	ano
	Platnost:	modální

Zvláštnosti

- **Vypisování právě zpracovávaného bloku v případě potlačeného zpracování blok po bloku**
  Zobrazování právě zpracovávaného bloku může být v podprogramech potlačeno pomocí příkazu DISPLOF. Pokud je příkaz DISPLOF naprogramován spolu s příkazem SBLOF, pak se při zastavení během zpracovávání blok po bloku uvnitř podprogramu vypisuje volání podprogramu.
- **Potlačení zpracování blok po bloku při asynchronních podprogramech (ASUP)**
  Aby se program ASUP zpracoval v jednom kroku, když je aktivní zpracování blok po bloku, musí být v programu ASUP naprogramován příkaz PROC s příkazem SBLOF. To platí také pro funkci "Editovatelný systémový ASUP" (MD11610 $MN_ASUP_EDITABLE).
  Jestliže je zastavování při zpracovávání blok po bloku v systémových nebo uživatelských asynchronních podprogramech (ASUP) potlačeno naprogramováním příkazu SBLOF na řádku PROC nebo prostřednictvím nastavení ve strojním parametru MD10702 $MN_IGNORE_SINGLEBLOCK_MASK (bit0 = 1 příp. bit1 = 1), potom může být zastavování během zpracovávání blok po bloku opět aktivováno naprogramováním příkazu SBLON v programu ASUP.
  Pokud je zastavování při zpracovávání blok po bloku v uživatelských asynchronních podprogramech (ASUP) prostřednictvím nastavení ve strojním parametru MD20117 $MC_IGNORE_SINGLEBLOCK_ASUP potlačeno, potom naprogramováním příkazu SBLON v programu ASUP **nemůže** být zastavování během zpracovávání blok po bloku znovu aktivováno.
- **Zvláštnosti u různých typů zpracování blok po bloku**
  - „SB2: Matematický blok“ A MD10702 $MN_IGNORE_SINGLEBLOCK_MASK, bit 12 = 1:
    → V bloku s příkazem SBLON k pozastavení nedochází.
  - „SB3: Blok po bloku jemně“:
    → Příkaz SBLOF je potlačen**.**
- **Potlačení zpracování blok po bloku v případě většího počtu programových úrovní**
  Pokud bylo v podprogramu v příkazu PROC naprogramováno SBLOF, při návratu z podprogramu pomocí příkazu M17 bude zpracování pozastaveno. Tím se zabrání, aby se ve volajícím programu ihned zpracoval následující blok. Jestliže je v podprogramu, který nemá příkaz SBLOF v příkazu PROC, příkazem SBLOF aktivováno potlačení zpracování blok po bloku, bude zpracování pozastaveno teprve až po následujícím bloku s funkcemi stroje ve volajícím programu. Je-li toto chování nežádoucí, musí být v podprogramu ještě před návratem (M17) znovu naprogramován příkaz SBLON. Při návratu pomocí příkazu RET se zpracování v nadřazeném programu nezastaví.

Příklady

Příklad 1: Potlačení zpracování blok po bloku v rámci programu

Výchozí situace: Zpracování blok po bloku je aktivní.

Programový kód	Komentář
N10 G1 X100 F1000	
N20 **SBLOF**	; Deaktivování zpracování v režimu blok po bloku.
N30 Y20	
N40 M100	
N50 R10=90	
N60 **SBLON**	; opětovné zapnutí zpracování blok po bloku
N70 M110	
N80 ...	

Oblast mezi bloky N20 a N60 se v režimu zpracování blok po bloku zpracuje jako jeden krok.

Příklad 2: Cyklus se má pro uživatele chovat jako jeden příkaz

Výchozí situace: Zpracování blok po bloku je aktivní.

Hlavní program:

Programový kód
...
N100 G1 X10 G90 F200
N120 X-4 Y6
N130 **CYCLE1**
N140 G1 X0
N150 M30

Cyklus CYCLE1:

Programový kód	Komentář
N100 **PROC** CYCLE1 DISPLOF **SBLOF**	; Potlačení zpracování blok po bloku pro celý program.
N110 R10=3*SIN(R20)+5	
N120 IF (R11 <= 0)	
N130 SETAL(61000)	
N140 ENDIF	
N150 G1 G91 Z=R10 F=R11	
N160 M17	

Cyklus CYCLE1 se při aktivním zpracovávání blok po bloku zpracuje najednou. Tzn. aby se cyklus CYCLE1 zpracoval, stačí jen jednou stisknout tlačítko Start.

Příklad 3: ASUP spouštěný z PLC pro aktivování změněného posunutí počátku a korekčních parametrů nástroje se nemá vypisovat.

Programový kód	Komentář
N100 **PROC** NV **SBLOF DISPLOF**	; Potlačení zpracování blok po bloku a vypisování právě zpracovávaného bloku.
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=$P_TOOLNO	
N170 RET	
N180 _G57: G57 D=$P_TOOL T=$P_TOOLNO	
N190 RET	
N200 END: D=$P_TOOL T=$P_TOOLNO	
N210 RET	

Příklad 4: Cílené pozastavení v podprogramu

Výchozí situace:

- Zpracování blok po bloku je aktivní.
- MD10702 $MN_IGNORE_SINGLEBLOCK_MASK, bit12 = 1

Hlavní program:

Programový kód	Komentář
N10 G0 X0	; Zastavení zpracování blok po bloku
N20 X10	; Zastavení zpracování blok po bloku
N30 **CYCLE**	; pohybový blok generovaný cyklem
N50 G90 X20	; Zastavení zpracování blok po bloku
M30	

Cyklus CYCLE:

Programový kód	Komentář
PROC CYCLE **SBLOF**	; Potlačení zpracování blok po bloku
N100 R0=1	
N110 **SBLON**	; Žádné zastavení zpracování blok po bloku kvůli MD10702 bit12=1
N120 X1	; Zastavení zpracování blok po bloku
N140 **SBLOF**	
N150 R0=2	

Programový kód	Komentář
RET	

Příklad 5: Potlačení zpracování blok po bloku v případě většího počtu programových úrovní

Výchozí situace:

- Je aktivováno zpracování blok po bloku typu 2.
- V bloku s příkazem SBLON nemá docházet k pozastavení (MD10702 $MN_IGNORE_SINGLEBLOCK_MASK, bit12 = 1)

```
Programový kód                          Komentář
N10 X0 F1000                            ; Zastavení zpracování blok po bloku
N20 UP1(0)
      PROC UP1(INT _NR) SBLOF           ; Deaktivování zpracování blok po bloku
                                        pro UP1
      N100 X10
      N110 UP2(0)
            PROC UP2(INT _NR)
            N200 X20
            N210 SBLON                  ; Aktivování zpracování v režimu blok po
                                        bloku
            N220 X22                    ; Zastavení zpracování blok po bloku
            N230 UP3(0)
                  PROC UP3(INT _NR)
                  N300 SBLOF            ; Deaktivování zpracování v režimu blok
                                        po bloku.
                  N305 X30
                  N310 SBLON            ; Aktivování zpracování v režimu blok po
                                        bloku
                  N320 X32              ; Zastavení zpracování blok po bloku
                  N330 SBLOF            ; Deaktivování zpracování v režimu blok
                                        po bloku.
                  N340 X34
                  N350 M17              ; Zastavení zpracování blok po bloku
                                        (M17)
            N240 X24                    ; Zastavení zpracování blok po bloku
                                        (N210)
            N250 M17                    ; Zastavení zpracování blok po bloku
                                        (M17)
      N120 X12
      N130 M17                          ; Zastavení zpracování blok po bloku
                                        (M17)
N30 X0                                  ; Zastavení zpracování blok po bloku
N40 M30                                 ; Zastavení zpracování blok po bloku
```

4.2.2.6 Potlačení vypisování aktuálního bloku (DISPLOF, DISPLON, ACTBLOCNO)

V příslušném okně se standardně vypisuje aktuální programový blok. V cyklech, příp. v podprogramech, může být vypisování aktuálního bloku pomocí příkazu DISPLOF potlačeno. Namísto aktuálního bloku se potom vypisuje volání cyklu, příp. podprogramu. Pomocí příkazu DISPLON může být potlačení vypisování bloku opět odstraněno.

Příkazy DISPLOF, příp. DISPLON se programují na řádek programu s příkazem PROC, takže platí pro celý podprogram a implicitně také pro všechny podprogramy volané tímto podprogramem, které neobsahují žádný z příkazů DISPLON, příp. DISPLOF. Toto chování platí také pro programy ASUP.

Syntaxe

```
PROC … DISPLOF
PROC … DISPLOF ACTBLOCNO
PROC … DISPLON
```

Význam

DISPLOF:	Příkaz pro potlačování vypisování aktuálního bloku.	
	Umístění:	Na konci programového řádku s příkazem PROC.
	Platnost:	Až do návratu z podprogramu nebo do konce programu.
	Upozornění: Jestliže jsou z podprogramu s příkazem DISPLOF vyvolávány další podprogramy, potom je vypisování aktuálního bloku potlačeno také v těchto podprogramech, pokud v nich není explicitně naprogramován příkaz DISPLON.	
DISPLON:	Příkaz pro odstranění potlačování vypisování aktuálního bloku	
	Umístění:	Na konci programového řádku s příkazem PROC.
	Platnost:	Až do návratu z podprogramu nebo do konce programu.
	Upozornění: Jestliže jsou z podprogramu s příkazem DISPLON vyvolávány další podprogramy, potom se i v těchto podprogramech aktuální programový blok, pokud v nich není explicitně naprogramován příkaz DISPLOF.	
ACTBLOCNO:	Příkaz DISPLOF ve spojení s atributem ACTBLOCNO způsobuje, že v případě alarmu se bude vypisovat číslo aktuálního bloku, ve kterém se alarm vyskytl. To platí i tehdy, jestliže je na některé z nižších programových úrovní naprogramován příkaz DISPLOF. V případě příkazu DISPLOF bez atributu ACTBLOCNO se oproti tomu vypisuje číslo bloku s voláním cyklu, příp. podprogramu z poslední programové úrovně, která není opatřena příkazem DISPLOF.	

Příklady

Příklad 1: Potlačení vypisování aktuálního bloku v cyklu

Programový kód	**Komentář**
PROC CYCLE(AXIS TOMOV, REAL POSITION) SAVE DISPLOF	; potlačení vypisování aktuálního bloku Místo toho se má vypisovat volání cyklu, např.: CYCLE(X,100.0)

Programový kód	Komentář
DEF REAL DIFF	; obsah cyklu
G01 ...	
...	
RET	; návrat zpět z podprogramu V okně pro vypisování bloku se vypisuje blok, který následuje za voláním cyklu.

Příklad 2: Vypisování bloku v případě aktivování alarmu

Podprogram SUBPROG1 (s atributem ACTBLOCNO):

Programový kód	Komentář
PROC SUBPROG1 DISPLOF ACTBLOCNO	
N8000 R10 = R33 + R44	
...	
N9040 R10 = 66 X100	; aktivování alarmu 12080
...	
N10000 M17	

Podprogram SUBPROG2 (bez atributu ACTBLOCNO):

Programový kód	Komentář
PROC SUBPROG2 DISPLOF	
N5000 R10 = R33 + R44	
...	
N6040 R10 = 66 X100	; aktivování alarmu 12080
...	
N7000 M17	

Hlavní program:

Programový kód	Komentář
N1000 G0 X0 Y0 Z0	
N1010 ...	
...	
N2050 SUBPROG1	; hlášení alarmu = "12080 kanál K1 blok N9040 Syntaktická chyba u textu R10="
N2060 ...	
N2350 SUBPROG2	; hlášení alarmu = "12080 kanál K1 blok N2350 Syntaktická chyba u textu R10="
...	
N3000 M30	

Příklad 3: Odstranění potlačení vypisování aktuálního bloku

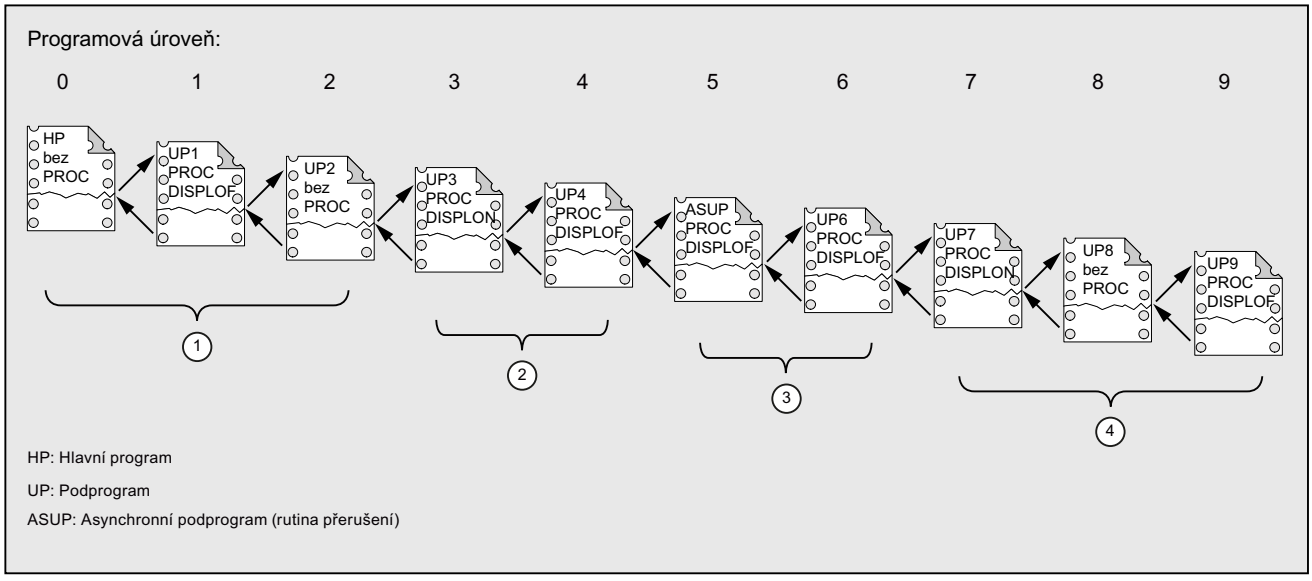
Podprogram SUB1 s potlačením vypisování:

Programový kód	**Komentář**
PROC SUB1 DISPLOF	; potlačení vypisování aktuálního bloku v podprogramu SUB1 Místo toho se má vypisovat blok s voláním podprogramu SUB1.
...	
N300 SUB2	; volání podprogramu SUB2
...	
N500 M17	

Podprogram SUB2 bez potlačení vypisování:

Programový kód	**Komentář**
PROC SUB2 DISPLON	; odstranění potlačení vypisování aktuálního bloku v podprogramu SUB2
...	
N200 M17	; návrat zpět do podprogramu SUB1 V podprogramu SUB1 je vypisování aktuálního bloku opět potlačeno.

Příklad 4: Chování vypisování při různých kombinacích příkazů DISPLON/DISPLOF



① V okně výpisu aktuálního bloku se vypisují řádky výrobního programu z programové úrovně 0.
② V okně výpisu aktuálního bloku se vypisují řádky výrobního programu z programové úrovně 3.
③ V okně výpisu aktuálního bloku se vypisují řádky výrobního programu z programové úrovně 3.
④ V okně výpisu aktuálního bloku se vypisují řádky výrobního programu z programové úrovně 7/8.

4.2.2.7 Označení podprogramů s přípravou (PREPRO)

Klíčovým slovem PREPRO na konci řádku s příkazem PROC mohou být označeny všechny soubory, které mají být při náběhu systému předběžně zpracovány.

Poznámka

Tento způsob přípravy programu závisí na nastavení strojního parametru MD10700 $MN_PREPROCESSING_LEVEL. Věnujte prosím pozornost informacím od výrobce stroje.

Syntaxe

```
PROC ... PREPRO
```

Význam

PREPRO:	Klíčové slovo pro označení všech souborů, které mají být v rámci náběhu systému předběžně zpracovány a které patří k NC programům uloženým v adresáři cyklů.

Načítání a vyvolávání podprogramů s přípravou

Jak podprogramy s parametry připravené při náběhu systému, tak také při volání podprogramu, se adresáři cyklů prochází ve stejné posloupnosti:

1. _N_CUS_DIR - cykly uživatele
2. _N_CMA_DIR - cykly výrobce
3. _N_CST_DIR - standardní cykly

V případě NC programů stejného názvu a s odlišnými významy se aktivuje první nalezený příkaz PROC a ostatní příkazy PROC jsou přeskakovány, aniž by bylo generováno alarmové hlášení.

4.2.2.8 Návrat zpět z podprogramu - M17

N konci podprogramu se nachází příkaz pro návrat M17 (příp. příkaz konce výrobního programu M30). Způsobuje návrat zpátky do volajícího výrobního programu na blok, který následuje za voláním podprogramu.

Poznámka

S příkazy M17 a M30 mají v NC jazyce zcela stejné postavení.

Syntaxe

```
PROC <název programu>
...
M17/M30
```

Okrajové podmínky

Vliv návratu z podprogramu na režim řízení pohybu po dráze

Jestliže se příkaz M17 (příp. M30) nachází v bloku výrobního programu samotný, aktivní režim řízení pohybu po dráze v kanálu se tím přeruší.

Aby se přerušení režimu řízení pohybu po dráze zabránilo, je zapotřebí příkaz M17 (příp. M30) napsat v posledním bloku posuvu. Kromě toho musí být nastaven na "0" následující strojní parametr:

MD20800 $MC_SPF_END_TO_VDI = 0 (žádný výstup příkazu M30/M17 na rozhraní NC/PLC)

Příklad:

1. Podprogram s příkazem M17 v samostatném bloku

Programový kód	Komentář
N10 G64 F2000 G91 X10 Y10	
N20 X10 Z10	
N30 M17	; návrat s přerušením režimu řízení pohybu po dráze

2. Podprogram s příkazem M17 v posledním bloku posuvu

Programový kód	Komentář
N10 G64 F2000 G91 X10 Y10	
N20 X10 Z10 M17	; návrat bez přerušení režimu řízení pohybu po dráze

4.2.2.9 Návrat z podprogramu pomocí příkazu RET

Jako náhrada příkazu pro návrat z podprogramu M17 je možné v podprogramu používat také příkaz RET. Příkaz RET musí být naprogramován v samostatném bloku výrobního programu. Stejně jako příkaz M17 i příkaz RET způsobuje návrat zpátky do volajícího výrobního programu na blok, který následuje za voláním podprogramu.

Poznámka

Naprogramováním parametrů může být chování při návratu pomocí příkazu RET změněno (viz "Návrat z podprogramu s nastavitelnými parametry (RET ...) (Strana 515)").

Aplikace

Příkaz RET je potřeba používat tehdy, pokud režim řízení pohybu po dráze (G64) (G641 ... G645) nemá být návratem z podprogramu přerušen.

Předpoklady

Příkaz RET se může používat pouze v podprogramech, které nejsou definovány s atributem SAVE.

Syntaxe

```
PROC <název programu>
...
RET
```

Příklad:

Hlavní program:

Programový kód	Komentář
PROC MAIN_PROGRAM	; začátek programu
...	
N50 SUB_PROG	; volání podprogramu: SUB_PROG
N60 ...	
...	
N100 M30	; konec programu

Podprogram:

Programový kód	Komentář
PROC SUB_PROG	
...	
N100 RET	; návrat se uskuteční na blok N60 v hlavním programu

4.2.2.10 Návrat z podprogramu s nastavitelnými parametry (RET ...)

Obecně platí, že pomocí příkazu RET naprogramujete skok z podprogramu zpět do volajícího programu. Zpracování bude potom pokračovat řádkem programu, který následuje za voláním podprogramu. Jestliže má zpracování programu pokračovat na jiném místě, máte k dispozici následující možnosti:

- Pokračování zpracováním programu po vyvolání cyklu pro oddělování třísky v režimu dialektu ISO (po popisu kontury).
- Návrat zpět do hlavního programu z libovolné úrovně podprogramu (i po ASUP) při zpracování chybových stavů.
- Návrat zpět přes více programových úrovní pro speciální aplikace v cyklech překladače a v režimu dialektu ISO.

Za tím účelem je potřeba naprogramovat příkaz RET s dalšími parametry.

Směr vyhledávání

Jestliže je zadán parametr <cílový blok>, uskuteční se skok zpět napřed na blok, který následuje za blokem volání. Potom se vyhledá cílový blok ve směru **konce** programu, na který se uskuteční návratový skok. Pokud je vyhledávání neúspěšné, spustí se další hledání, tentokrát směrem k začátku programu.

Syntaxe

```
RET("<cílový blok>")
RET("<cílový blok>",<blok za cílovým blokem>)
RET("<cílový blok>",<blok za cílovým blokem>,<počet úrovní při
návratu>)
RET("<cílový blok>", ,<počet úrovní při návratu>)
RET("<cílový blok>",<blok za cílovým blokem>,<počet úrovní při
návratu>,
<návrat na začátek programu>)
RET( , ,<počet úrovní při návratu>,<návrat na začátek programu>)
```

Význam

RET:	Konec podprogramu	
<cílový blok>:	V parametru je jako cíl skoku uveden blok, na kterém má zpracování programu pokračovat. Jestliže není naprogramován parametr <počet úrovní při návratu>, potom se cíl skoku nachází v programu, z něhož byl aktuální podprogram vyvolán. Možné údaje jsou následující:	
	<číslo bloku>	Číslo cílového bloku. Vyhledávání čísla bloku v programu, do kterého se zpracování vrací, probíhá napřed směrem k jeho konci.
	<návěští skoku>	Návěští skoku, na které se v programu uskutečňuje skok zpět, musí existovat. Vyhledávání návěští skoku v programu, do kterého se zpracování vrací, probíhá napřed směrem k jeho konci.
	<řetězec znaků>	Řetězec znaků, na který se v programu uskutečňuje skok zpět, musí existovat (např. jako název programu nebo název proměnné). Vyhledávání řetězce znaků v programu, do kterého se zpracování vrací, probíhá napřed směrem k jeho konci. Pro programování řetězce znaků v cílovém bloku platí následující pravidla: • **Mezera na konci** (na rozdíl od návěští skoku, které je na konci označeno ":"). • **Před řetězcem znaků** smí být uvedeny pouze číslo bloku a/nebo návěští skoku, **žádné programové příkazy**.

`<blok za cílovým blokem>`:	Pomocí tohoto parametru se zadává, zda má zpracování programu pokračovat blokem, který je zadán v parametru <cílový blok>, nebo blokem, který za ním následuje.		
	Typ:	INT	
	Hodnota:	0	Skok zpět se uskuteční na blok, který je zadán v parametru <cílový blok>.
		> 0	Skok zpět se uskuteční na blok, který následuje za blokem uvedeným v parametru <cílový blok>.
`<počet úrovní při návratu>`:	V tomto parametru se zadává počet programových úrovní, o které se má skočit zpět, aby se zde našel cílový blok a aby zpracování programu pokračovalo.		
	Typ:	INT	
	Hodnota:	1	Program bude pokračovat na "aktuální programové úrovni -1" (tedy stejně jako v případě příkazu `RET` bez parametrů).
		2	Program bude pokračovat na "aktuální programové úrovni -2", tzn. jedna úroveň bude přeskočena.
		3	Program bude pokračovat na "aktuální programové úrovni -3", tzn. dvě úrovně budou přeskočeny.
		...	
	Rozsah hodnot:	1 ... 15	
`<Skok zpátky na začátek programu>`:	V tomto parametru se zadává, zda se při návratu do hlavního programu, ve kterém je aktivní **režim dialektu ISO**, má pokračovat ve zpracování tohoto programu od jeho začátku.		
	Typ:	BOOL	
	Hodnota:	1	Jestliže se uskutečňuje návrat zpět do hlavního programu a jestliže je v něm aktivní **režim dialektu ISO**, skočí se na začátek programu.

Poznámka

Při návratu z podprogramu, při němž je pro vyhledání cílového bloku zadán řetězec znaků, se ve volaném programu vždy napřed hledá návěští skoku.

Pokud má být cíl skoku jednoznačně definován prostřednictvím řetězce znaků, nesmí proto řetězec znaků odpovídat názvu návěští skoku, neboť potom by se při návratu z podprogramu vždy skákalo na návěští skoku a nikoli na řetězec znaků (viz příklad 2).

Okrajové podmínky

Návrat zpět přes více programových úrovní

Při návratu zpět přes několik programových úrovní jsou vyhodnocovány příkazy SAVE jednotlivých programových úrovní.

Jestliže je při návratu zpět přes několik programových úrovní aktivní modální volání podprogramu a pokud je v některém z přeskakovaných podprogramů naprogramován příkaz pro deaktivování modálního podprogramu MCALL, zůstává modální podprogram i nadále aktivní.

UPOZORNĚNÍ
Chyba v programu Při návratu přes více programových úrovní je odpovědností uživatele, aby se postaral, že potřebná modální nastavení zůstávají i nadále v platnosti. Toho je možno dosáhnout např. naprogramováním odpovídajícího hlavního bloku.

Zpracování z externího zdroje

Jestliže mají být pomocí funkce „Zpracování z externího zdroje" zpracovány hlavní programy a podprogramy, které se nacházejí v externí paměti programů, musí se cíle návratových skoků naprogramované pomocí příkazu RET(...) nacházet ve vyrovnávací paměti. Jinak nebude cíl skoku nalezen a dojde k přerušení programu a aktivování alarmu 14000.

Poznámka

Aby bylo možné zpracovávat externí programy bez omezení, pokud jde o naprogramované příkazy skoků, doporučujeme, abyste místo funkce „Zpracování z externího zdroje" používali funkci „Zpracování z externí paměti (EES)".

Příklady

Příklad 1: Opětovné pokračování v hlavním programu po zpracování programu ASUP

```
Programování                          Komentář
N10010 CALL "UP1"                     ; Programová úroveň 0 (hlavní program)
  N11000 PROC UP1                     ; Programová úroveň 1
  N11010 CALL "UP2"
  N12000 PROC UP2                     ; Programová úroveň 2
  ...
  N19000 PROC ASUP                    ; Programová úroveň 3 (zpracování programu ASUP)
  ...
  N19100 RET("N10900", ,$P_STACK)     ; návrat z podprogramu do hlavního programu
                                      ; $P_STACK: aktuální programová úroveň
N10900                                ; cílový blok v hlavním programu
N10910 MCALL                          ; deaktivování modálního volání podprogramu
N10920 G0 G60 G40 M5                  ; inicializace dalších modálních nastavení
```

Příklad 2: Řetězec znaků (<STRING>) jako zadání pro vyhledávání cílového bloku

Hlavní program:

Programový kód	Komentář
PROC MAIN_PROGRAM	
N1000 DEF INT iVar1=1, iVar2=4	
N1010 ...	
N1200 subProg1	; Volání podprogramu "subProg1"
N1210 M2 S1000 X10 F1000	
N1220	
N1400 subProg2	; Volání podprogramu "subProg2"
N1410 M3 S500 Y20	
N1420 ..	
N1500 lab1: iVar1=R10*44	
N1510 F500 X5	
N1520 ...	
N1550 subprog1: G1 X30	; "subProg1" je zde definováno jako návěští skoku.
N1560 ...	
N1600 subProg3	; Volání podprogramu "subProg3"
N1610 ...	
N1900 M30	

Podprogram subProg1:

Programový kód	Komentář
PROC subProg1	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg2")	; Návrat zpět do hlavního programu na blok N1400

Podprogram subProg2:

Programový kód	Komentář
PROC subProg2	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("iVar1")	; Návrat zpět do hlavního programu na blok N1500

Podprogram subProg3:

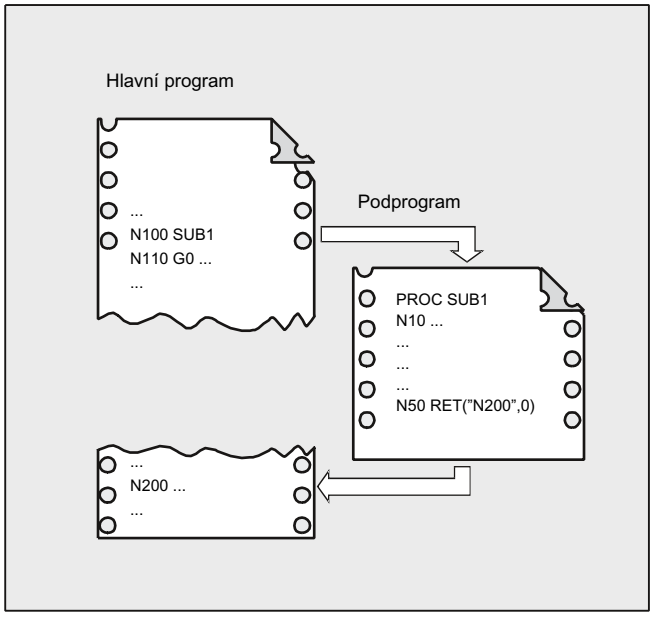
Programový kód	Komentář
PROC subProg3	
N2000 R10=R20+100	

Programový kód	**Komentář**
N2010 ...	
N2200 RET("subProg1")	; Návrat zpět do hlavního programu na blok N1550

Další informace

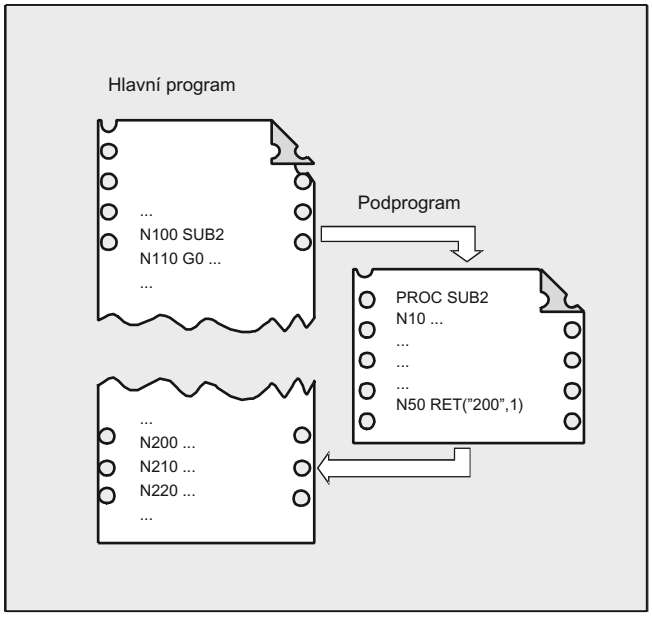
Následující obrázky ukazují výsledky různých nastavení návratových parametrů

1. <cílový blok> = "N200", <blok za cílovým blokem> = 0



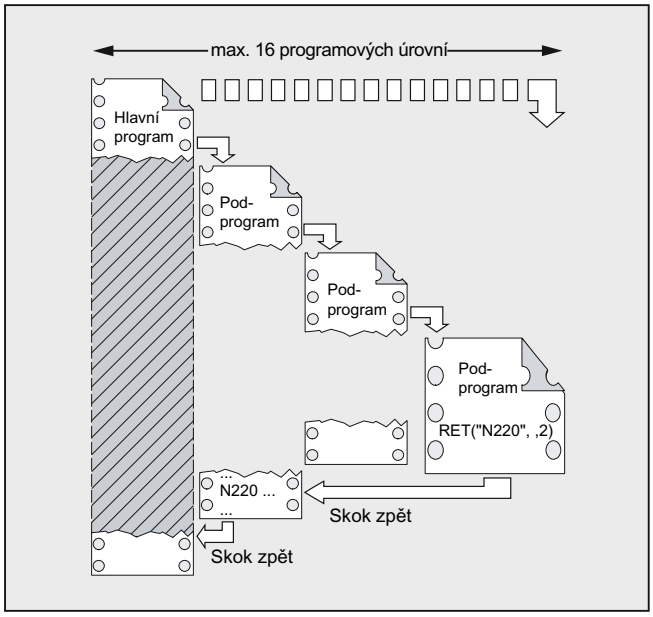
Po příkazu `RET` bude zpracovávání programu pokračovat blokem `N200` v hlavním programu.

2. <cílový blok> = "N200", <blok za cílovým blokem> = 1



Po příkazu `RET` bude zpracovávání programu pokračovat blokem (`N210`), který v hlavním programu následuje za blokem `N200`.

3. <cílový blok> = "N220", <počet úrovní při návratu> = 2



Po příkazu `RET` budou při návratu zpět přeskočeny dvě programové úrovně a zpracovávání programu bude pokračovat blokem `N220`.

4.2.2.11 Návrat z podprogramu s nastavitelnými parametry (RETB ...):

Obecně platí, že pomocí příkazu `RETB` naprogramujete skok z podprogramu zpět do volajícího programu. Zpracování bude potom pokračovat řádkem programu, který následuje za voláním podprogramu. Jestliže má zpracování programu pokračovat na jiném místě, máte k dispozici následující možnosti:

- Pokračování zpracováním programu po vyvolání cyklu pro oddělování třísky v režimu dialektu ISO (po popisu kontury).
- Návrat zpět do hlavního programu z libovolné úrovně podprogramu (i po ASUP) při zpracování chybových stavů.
- Návrat zpět přes více programových úrovní pro speciální aplikace v cyklech překladače a v režimu dialektu ISO.

Za tím účelem je potřeba naprogramovat příkaz `RETB` s dalšími parametry.

Směr vyhledávání

Jestliže je zadán parametr `<cílový blok>`, uskuteční se skok zpět napřed na blok, který následuje za blokem volání. Potom se vyhledá cílový blok ve směru **začátku** programu, na který se uskuteční návratový skok. Pokud je vyhledávání neúspěšné, spustí se další hledání, tentokrát směrem ke konci programu.

Syntaxe

```
RETB(„<cílový blok>")
RETB("<cílový blok>",<blok za cílovým blokem>)
RETB("<cílový blok>",<blok za cílovým blokem>,<počet úrovní při
návratu>)
RETB("<cílový blok>", ,<počet úrovní při návratu>)
RETB("<cílový blok>",<blok za cílovým blokem>,<počet úrovní při
návratu>,
<skok zpátky na začátek programu>)
RETB( , ,<počet úrovní při návratu>,<návrat na začátek programu>)
```

Význam

RETB:	Konec podprogramu		
<cílový blok>:	V parametru je jako cíl skoku uveden blok, na kterém má zpracování programu pokračovat. Jestliže není naprogramován parametr <počet úrovní při návratu>, potom se cíl skoku nachází v programu, z něhož byl aktuální podprogram vyvolán. Možné údaje jsou následující:		
	<číslo bloku>		Číslo cílového bloku. Vyhledávání čísla bloku v programu, do kterého se zpracování vrací, probíhá napřed směrem k jeho začátku.
	<návěští skoku>		Návěští skoku, na které se v programu uskutečňuje skok zpět, musí existovat. Vyhledávání návěští skoku v programu, do kterého se zpracování vrací, probíhá napřed směrem k jeho začátku.
	<řetězec znaků>		Řetězec znaků, na který se v programu uskutečňuje skok zpět, musí existovat (např. jako název programu nebo název proměnné). Vyhledávání řetězce znaků v programu, do kterého se zpracování vrací, probíhá napřed směrem k jeho začátku. Pro programování řetězce znaků v cílovém bloku platí následující pravidla: • **Mezera na konci** (na rozdíl od návěští skoku, které je na konci označeno ":"). • **Před řetězcem znaků** smí být uvedeny pouze číslo bloku a/nebo návěští skoku, **žádné programové příkazy**.
<blok za cílovým blokem>:	Pomocí tohoto parametru se zadává, zda má zpracování programu pokračovat blokem, který je zadán v parametru <cílový blok>, nebo blokem, který za ním následuje.		
	Typ:	INT	
	Hodnota:	0	Skok zpět se uskuteční na blok, který je zadán v parametru <cílový blok>.
		> 0	Skok zpět se uskuteční na blok, který následuje za blokem uvedeným v parametru <cílový blok>.

<počet úrovní při návratu>:	V tomto parametru se zadává počet programových úrovní, o které se má skočit zpět, aby se zde našel cílový blok a aby zpracování programu pokračovalo.		
	Typ:	INT	
	Hodnota:	1	Program bude pokračovat na "aktuální programové úrovni -1" (tedy stejně jako v případě příkazu `RET` bez parametrů).
		2	Program bude pokračovat na "aktuální programové úrovni -2", tzn. jedna úroveň bude přeskočena.
		3	Program bude pokračovat na "aktuální programové úrovni -3", tzn. dvě úrovně budou přeskočeny.
		...	
	Rozsah hodnot:	1 ... 15	
<Skok zpátky na začátek programu>:	V tomto parametru se zadává, zda se při návratu do hlavního programu, ve kterém je aktivní **režim dialektu ISO**, má pokračovat ve zpracování tohoto programu od jeho začátku.		
	Typ:	BOOL	
	Hodnota:	1	Jestliže se uskutečňuje návrat zpět do hlavního programu a jestliže je v něm aktivní **režim dialektu ISO**, skočí se na začátek programu.

Poznámka

Při návratu z podprogramu, při němž je pro vyhledání cílového bloku zadán řetězec znaků, se ve volaném programu vždy napřed hledá návěští skoku.

Pokud má být cíl skoku jednoznačně definován prostřednictvím řetězce znaků, nesmí proto řetězec znaků odpovídat názvu návěští skoku, neboť potom by se při návratu z podprogramu vždy skákalo na návěští skoku a nikoli na řetězec znaků (viz příklad 2).

Okrajové podmínky

Návrat zpět přes více programových úrovní

Při návratu zpět přes několik programových úrovní jsou vyhodnocovány příkazy SAVE jednotlivých programových úrovní.

Jestliže je při návratu zpět přes několik programových úrovní aktivní modální volání podprogramu a pokud je v některém z přeskakovaných podprogramů naprogramován příkaz pro deaktivování modálního podprogramu MCALL, zůstává modální podprogram i nadále aktivní.

UPOZORNĚNÍ
Chyba v programu
Při návratu přes více programových úrovní je odpovědností uživatele, aby se postaral, že potřebná modální nastavení zůstávají i nadále v platnosti. Toho je možno dosáhnout např. naprogramováním odpovídajícího hlavního bloku.

Zpracování z externího zdroje

Jestliže mají být pomocí funkce „Zpracování z externího zdroje“ zpracovány hlavní programy a podprogramy, které se nacházejí v externí paměti programů, musí se cíle návratových skoků naprogramované pomocí příkazu RETB(...) nacházet ve vyrovnávací paměti. Jinak nebude cíl skoku nalezen a dojde k přerušení programu a aktivování alarmu 14000.

Poznámka

Aby bylo možné zpracovávat externí programy bez omezení, pokud jde o naprogramované příkazy skoků, doporučujeme, abyste místo funkce „Zpracování z externího zdroje“ používali funkci „Zpracování z externí paměti (EES)“.

Příklad

```
Programový kód                                  Komentář
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
```

```
Programový kód                      Komentář
PROC END_CYC(…)                     ; volání v hlavním programu, řádek N3040
   N10000 …
   N15000 if status == 1
      N15010 RETB("START_CYC")      ; skok zpět do volajícího programu BEISPIEL.MPF
                                    ; vyhledat řetězec znaků "START_CYC"
                                    ; směr vyhledávání: dozadu směrem k začátku pro-
                                    gramu
                                    ; pokračování zpracování programu na řádku N3000
   N15020 endif
   N15030 if status == 0
      N15040 RET                    ; skok zpět do volajícího programu BEISPIEL.MPF
                                    ; pokračování zpracování programu na řádku N3050
   N15050 endif
```

Programový kód	Komentář
N16000 RET("START_CYC")	; skok zpět do volajícího programu BEISPIEL.MPF
	; vyhledat řetězec znaků "START_CYC"
	; směr vyhledávání: dopředu směrem ke konci programu
	; pokračování zpracování programu na řádku N4500
N17060 RETB	; skok zpět do volajícího programu BEISPIEL.MPF
	; pokračování zpracování programu na řádku N3050
	; příkaz RETB bez zadání parametru je identický s příkazem RET

4.2.3 Volání podprogramu

4.2.3.1 Volání podprogramu bez předávání parametrů

Volání podprogramu se uskutečňuje buď pomocí adresy L a číslem podprogramu nebo zadáním názvu podprogramu.

Jako podprogram je možné volat i hlavní program. Konec programu `M2` nebo `M30` uvedený v hlavním programu bude v tomto případě vyhodnocován jako `M17` (konec programu s návratem zpátky do volajícího programu).

Poznámka

Analogicky je možné spustit podprogram jako program hlavní.

Strategie řídícího systému při vyhledávání:

Existuje *_MPF ?

Existuje *_SPF ?

Z toho vyplývá: Jestliže jsou název volaného podprogramu a název hlavního programu identické, potom bude znovu volán program, který spustil volání. Tento zpravidla nežádoucí efekt je nutno eliminovat jednoznačnou volnou názvu hlavního programu a podprogramu.

Poznámka

Podprogramy, které nevyžadují žádné předávání parametrů, mohou být volány také z inicializačního souboru.

Syntaxe

```
L<číslo>/<název programu>
```

Poznámka

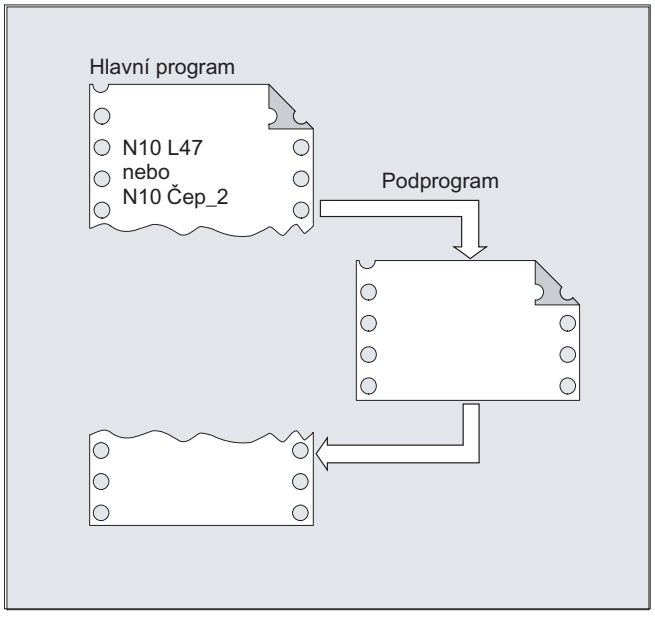
Volání podprogramu musí být vždy naprogramováno v samostatném NC-bloku.

Význam

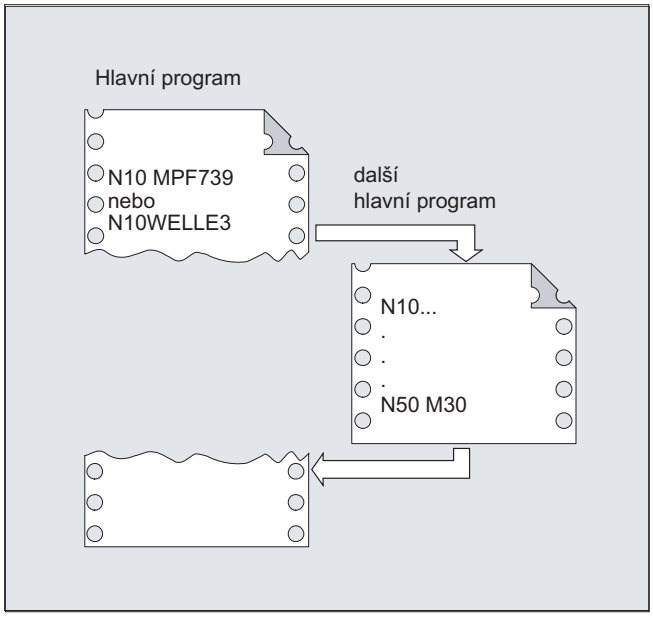
L:	Adresa pro volání podprogramu	
<číslo>:	Číslo podprogramu	
	Typ:	INT
	Hodnota:	Maximálně 7 míst v desítkové soustavě **Pozor:** Nuly na začátku mají při vyhodnocování názvů svůj význam (--> L123, L0123 a L00123 jsou tři různé podprogramy).
<název programu>:	Název podprogramu (nebo hlavního programu)	

Příklady

Příklad 1: Volání podprogramu bez předávání parametrů



Příklad 2: Volání hlavního programu jako podprogramu



Viz také

Podprogram bez předávání parametrů (Strana 499)

4.2.3.2 Volání podprogramu s předáváním parametrů (EXTERN)

Při volání podprogramu s předáváním parametrů mohou být proměnné nebo hodnoty předávány přímo (nikoli u parametrů typu `VAR`).

Podprogramy s předáváním parametrů musí být před voláním v hlavním programu s příkazem `EXTERN` deklarovány (např. na začátku programu). Uvádí se přitom název podprogramu a typy proměnných v posloupnosti, v jaké jsou předávány.

UPOZORNĚNÍ
Nebezpečí záměny Jak typy proměnných, tak i posloupnost při předávání se musí shodovat s definicemi, které byly v podprogramu zadány v příkazu `PROC`. Názvy parametrů se mohou v hlavním programu a v podprogramu lišit.

Syntaxe

```
EXTERN <název programu>(<typ_par1>,<typ_par2>,<typ_par3>)
...
<název programu>(<hodnota_par1>,<hodnota_par2>,<hodnota_par3>)
```

Poznámka

Volání podprogramu musí být vždy naprogramováno v samostatném NC-bloku.

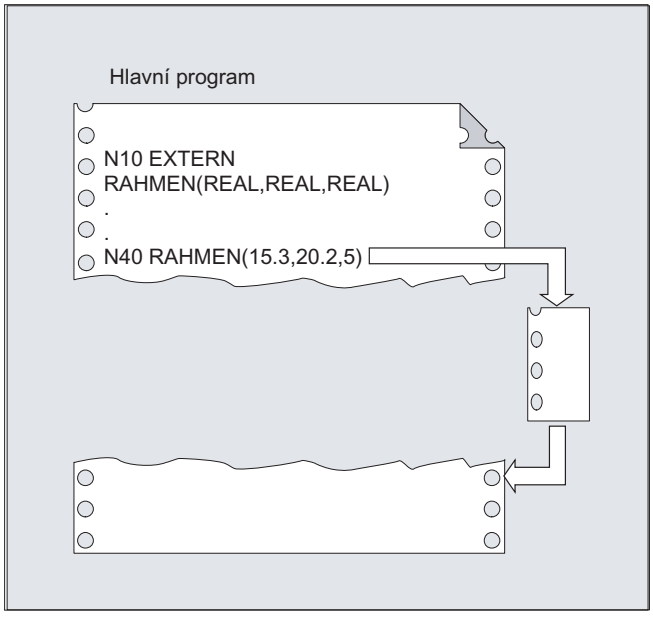
Význam

<název programu>:	Název podprogramu
EXTERN:	Klíčové slovo pro identifikaci podprogramu s předáváním parametrů **Upozornění:** Příkaz EXTERN musí být uváděn jen tehdy, pokud se podprogram nachází v adresáři obrobku nebo v globálním adresáři podprogramů. Při volání cyklů se příkaz EXTERN používat nemusí.
<typ_par1>,<typ_par2>,<typ_par3>:	Typy proměnných předávaných parametrů v pořadí jejich předávání
<hodnota_par1>,<hodnota_par2>,<hodnota_par3>:	Hodnota proměnné pro předávaný parametr

Příklady

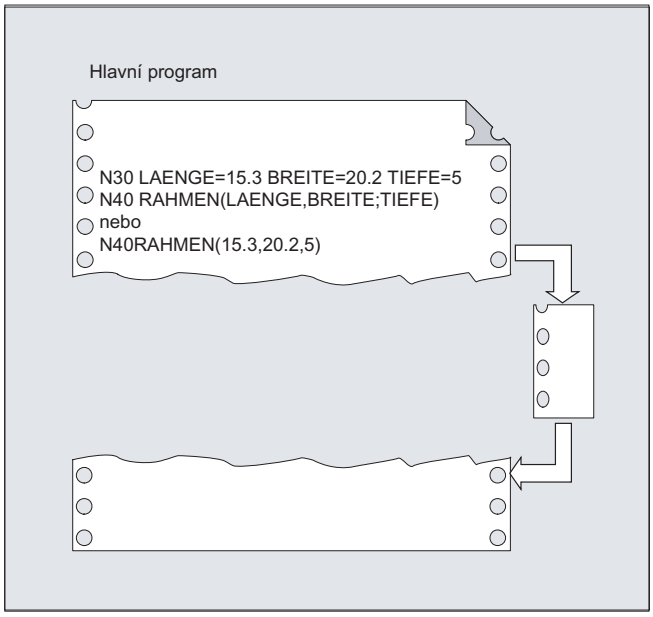
Příklad 1: Volání podprogramu s předcházející deklarací

Programový kód	**Komentář**
N10 EXTERN RAHMEN(REAL,REAL,REAL)	; Zadání podprogramu.
...	
N40 RAHMEN(15.3,20.2,5)	; Volání podprogramu s předáváním parametrů.



Příklad 2: Volání podprogramu bez deklarace

Programový kód	**Komentář**
`N10 DEF REAL LAENGE, BREITE, TIEFE`	
`N20 …`	
`N30 LAENGE=15.3 BREITE=20.2 TIEFE=5`	
`N40 RAHMEN(LAENGE,BREITE,TIEFE)`	`; nebo: N40 RAHMEN(15.3,20.2,5)`



Viz také

Podprogram s předáváním parametrů Call-by-Value (PROC) (Strana 500)

Podprogram s předáváním parametrů Call-by-Reference (PROC, VAR) (Strana 502)

4.2.3.3 Počet opakování programu (P)

Jestliže má být podprogram zpracován několikrát po sobě, je možné v bloku s voláním podprogramu naprogramovat pomocí adresy `P` požadovaný počet opakování tohoto programu.

⚠ POZOR

Volání podprogramu s opakováním programu a s předáváním parametrů

Parametry se předávají pouze při volání programu, resp. při prvním průchodu. Pro další opakování zůstávají tyto parametry nezměněné. Jestliže si přejete, aby se při opakováních programu tyto parametry měnily, musíte v podprogramu definovat odpovídající opatření.

Syntaxe

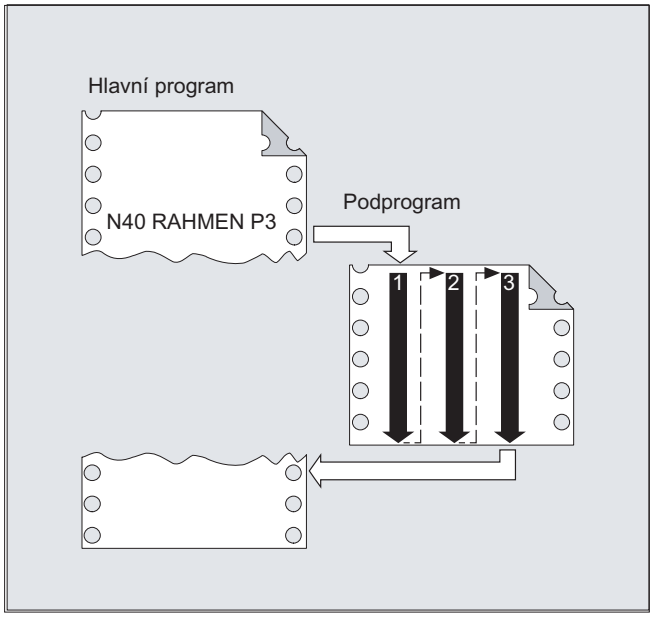
```
<název programu> P<hodnota>
```

Význam

`<název programu>`:	Volání podprogramu	
`P`:	Adresa pro naprogramování opakování programu	
`<hodnota>`:	Počet opakování programu	
	Typ:	INT
	Rozsah hodnot:	1 ... 9999 (bez znaménka)

Příklad:

Programový kód	**Komentář**
...	
N40 RAHMEN P3	; podprogram RAHMEN má být zpracován třikrát po sobě
...	



4.2.3.4 Modální volání podprogramu (MCALL)

Prostřednictvím modálního volání podprogramu `MCALL <název programu>` není zadaný podprogram vyvolán okamžitě. Místo toho se jeho volání uskutečňuje od tohoto okamžiku ve výrobním programu automaticky po každém bloku s příkazem pohybu po dráze. A to i na různých programových úrovních.

Poznámka

Při zpracování programu se vždy uplatňuje jen poslední modální volání podprogramu `MCALL <název programu>`. Aktuální modální volání podprogramu nahrazuje to, které bylo v platnosti předtím.

Pokud jsou do podprogramu předávány nějaké parametry, uskutečňuje se předávání parametrů jen při volání:

```
MCALL(<název programu>(Par1, Par2, ...))
```

Poznámka

Definiční datová oblast na začátku podprogramu se zpracovává pouze jedenkrát, a to při zpracování bloku, v němž je naprogramováno volání s příkazem MCALL. Při následujících voláních podprogramu po pohybových blocích se již definiční datová oblast nezpracovává. To má za následek, že hodnota přiřazená při definici lokální uživatelské proměnné (LUD) již není po prvním volání k dispozici a místo ní se v této proměnné nachází hodnota, která byla v cyklu naposled do proměnné uložena. Proměnná není po prvním volání nově definována, místo toho zůstává uložena poslední hodnota z předcházejícího volání. Aby se tomuto chování zabránilo, doporučuje se definice LUD a přiřazení hodnot od sebe oddělit a přiřazení hodnot naprogramovat v samostatném bloku až za definiční datovou oblastí.

UPOZORNĚNÍ
Modální volání podprogramu bez pohybu po dráze Modální podprogram se vyvolává v následujících situacích i bez toho, že by byl naprogramován nějaký pohyb po dráze: • Naprogramování adres S nebo F, pokud je aktivní příkaz G0 nebo G1. • Pokud se v bloku nachází samotný příkaz G0 nebo G1 nebo pokud byl naprogramován spolu s dalšími příkazy G-funkcí.

Syntaxe

```
MCALL <Název programu>
...
MCALL
```

Význam

`MCALL <Název programu>`:	Aktivování funkce „Modální volání podprogramu“
`<Název programu>`:	Název podprogramu
`MCALL`:	Prostřednictvím příkazu MCALL bez zadání názvu programu je funkce "Modální volání podprogramu" deaktivována.

Okrajové podmínky

ASUP

Jestliže je zpracování výrobního programu přerušeno příkazem ASUP (viz kapitola "Rutiny přerušení (ASUP) (Strana 544)„), v tomto stavu ASUP se žádná modální volání podprogramů neprovádějí.

Pokud je příkaz ASUP spuštěn, když je kanál ve stavu „Reset“, chová se vzhledem k modálním voláním podprogramů jako normální výrobní program.

Cyklus pro výměnu nástroje

Jestliže je v cyklu pro výměnu nástroje funkce "Modální volání podprogramu“ deaktivována, je potřeba zajistit, aby byl cyklus pro výměnu nástroje vyvoláván případně také po vyhledávání bloku implicitně prostřednictvím vyhledávání bloku s příkazem ASUP nebo manuálně pomocí přepsání v paměti. V této situaci nesmí být funkce „Modální volání podprogramu“ deaktivována, protože jinak by byl výsledek vyhledávání bloku nesprávný. Z tohoto důvodu se doporučuje deaktivování funkce "Modální volání podprogramu" v cyklu pro výměnu nástroje naprogramovat následujícím způsobem:

```
Programový kód          Komentář
...
```

Programový kód	Komentář
IF $AC_ASUP == 0	; volání se neuskutečňuje prostřednictvím vyhledávání bloku s příkazem ASUP nebo přepsáním v paměti
MCALL	; deaktivování funkce modálního volání podprogramu
ENDIF	
...	

Příklady

Příklad 1

Programový kód	Komentář
N10 G0 X0 Y0	
N20 MCALL L70	; aktivování modálního volání podprogramu L70
N30 X10 Y10	; najíždění na bod X10 Y10, potom se vyvolá program L70
N40 X20 Y20	; najíždění na bod X20 Y20, potom se vyvolá program L70
...	
N100 MCALL	; deaktivování funkce modálního volání podprogramu
N110 X0 Y0	; najíždění na bod X0 Y0, podprogram L70 se **nevyvolává**

Příklad 2

Programový kód
N10 G0 X0 Y0
N20 MCALL L70
N30 L80

V tomto příkladu se nacházejí následující NC bloky s naprogramovanými dráhovými osami v podprogramu L80. Podprogram L70 se vyvolává prostřednictvím podprogramu L80.

4.2.3.5 Nepřímé volání podprogramu (CALL)

V závislosti na daných podmínkách mohou být na jednom místě vyvolávány různé podprogramy. Za tím účelem se název podprogramu ukládá do proměnné typu STRING. Volání podprogramu se pak uskutečňuje příkazem `CALL` a názvem proměnné.

Poznámka

Nepřímé volání podprogramu je možné pouze pro podprogramy bez předávání parametrů. Pro účely přímého volání podprogramu uložte jeho název do konstanty typu STRING.

Syntaxe

```
CALL <název programu>
```

Význam

CALL:	Příkaz pro nepřímé volání podprogramu	
<název programu>:	Název podprogramu (proměnná nebo konstanta)	
	Typ:	STRING

Příklad:

Přímé volání s konstantou typu STRING:

Programový kód	**Komentář**
…	
CALL "/_N_WKS_DIR/_N_SUBPROG_WPD/_N_TEIL1_SPF"	; Podprogram TEIL1 s přímým voláním typu CALL.
…	

Nepřímé volání pomocí proměnné:

Programový kód	**Komentář**
…	
DEF STRING[100] PROGNAME	; Definice proměnné.
PROGNAME="/_N_WKS_DIR/_N_SUBPROG_WPD/_N_TEIL1_SPF"	; Přiřazení podprogramu TEIL1 proměnné PROGNAME.
CALL PROGNAME	; Nepřímé volání podprogramu TEIL1 pomocí příkazu CALL a proměnné PROGNAME.
…	

4.2.3.6 Nepřímé volání podprogramu se zadáním úseku programu, který á být zpracován (CALL BLOCK ... TO ...)

Pomocí příkazu CALL a kombinace klíčových slov BLOCK ... TO se nepřímo vyvolává podprogram a v něm se zpracovává úsek, který začíná počátečním a končí koncovým návěštím.

Syntaxe

```
CALL <název programu> BLOCK <počáteční návěští> TO <koncové návěští>
CALL BLOCK <počáteční návěští> TO <koncové návěští>
```

Význam

CALL:	Příkaz pro nepřímé volání podprogramu	
<název programu>:	Název podprogramu (proměnná nebo konstanta), který obsahuje úsek programu, který má být zpracován (**údaj je nepovinný**).	
	Typ:	STRING
	Upozornění: Jestliže parametr <název programu> není naprogramován, úsek programu označený návěštími <počáteční návěští> a <koncové návěští> bude vyhledán v aktuálním programu a bude zpracován.	
BLOCK ... TO ... :	Kombinace klíčových slov pro nepřímé zpracování úseku programu	
<počáteční návěští:	Proměnná, která odkazuje na začátek úseku programu, který má být zpracován.	
	Typ:	STRING
<koncové návěští:	Proměnná, která odkazuje na konec úseku programu, který má být zpracován.	
	Typ:	STRING

Příklad:

Hlavní program:

Programový kód	**Komentář**
...	
DEF STRING[20] STARTLABEL, ENDLABEL	; definice proměnných pro počáteční a koncové návěští
STARTLABEL="LABEL_1"	
ENDLABEL="LABEL_2"	
...	
CALL "CONTUR_1" BLOCK STARTLABEL TO ENDLABEL	; nepřímé volání podprogramu a označení úseku programu, který má být zpracován
...	

Podprogram:

Programový kód	**Komentář**
PROC CONTUR_1 ...	
LABEL_1	; počáteční návěští: Začátek zpracování úseku programu
N1000 G1 ...	
...	
LABEL_2	; koncové návěští: Konec zpracování úseku programu
...	

4.2.3.7 Nepřímé volání programu naprogramovaného v jazyce ISO (ISOCALL)

Pomocí nepřímého volání programu pomocí příkazu ISOCALL je možné vyvolat program naprogramovaný v jazyce ISO. Přitom se aktivuje režim ISO nastavený ve strojních parametrech. Na konci programu se znovu stane platným původní režim zpracování. Jestliže není ve strojních parametrech nastaven žádný režim ISO, uskuteční se volání podprogramu v režimu Siemens.

Pokud budete potřebovat další informace k režimu ISO: Příručka k funkcím, Dialekty ISO

Syntaxe

```
ISOCALL <název programu>
```

Význam

ISOCALL:	Klíčové slovo pro nepřímé volání podprogramu, při kterém se aktivuje režim ISO nastavený ve strojních parametrech.
<název programu>:	Název programu naprogramovaného v jazyce ISO (proměnná nebo konstanta typu STRING)

Příklad: Vyvolání kontury s programováním cyklu v režimu ISO

Programový kód	**Komentář**
0122_SPF	; popis kontury v režimu ISO
N1010 G1 X10 Z20	
N1020 X30 R5	
N1030 Z50 C10	
N1040 X50	
N1050 M99	
N0010 DEF STRING[5] PROGNAME = "0122"	; výrobní program (cyklus) od firmy Siemens
...	
N2000 R11 = $AA_IW[X]	
N2010 ISOCALL PROGNAME	
N2020 R10 = R10+1	; zpracování programu 0122 v režimu ISO
...	
N2400 M30	

4.2.3.8 Volání podprogramu s udáním cesty a parametrů (PCALL)

Pomocí příkazu PCALL mohou být vyvolávány podprogramy s absolutním zadáním cesty a s předáváním parametrů.

Syntaxe

```
PCALL <cesta/název programu>(<parametr 1>,…,<parametr n>)
```

Význam

PCALL:	Klíčové slovo pro volání podprogramu s absolutním zadáním cesty.
<cesta/název programu>:	Absolutní zadání cesty včetně názvu podprogramu. Pokud budete potřebovat informace o pravidlech pro zadávání cesty, viz "Adresování souborů v paměti pro programy (Strana 559)". Jestliže žádná absolutní cesta nebyla zadána, chová se příkaz PCALL stejně jako standardní volání s identifikátorem podprogramu. Název programu se zadává bez předpony a bez přípony souboru. Jestliže má být název programu naprogramován s úvodním řetězcem a s příponou souboru, potom musí být explicitně vysvětlen i s předponou a s příponou pomocí příkazu EXTERN.
<parametr 1>, ...:	Aktuální parametry v souladu s příkazem PROC v podprogramu.

Příklad:

Programový kód
PCALL/_N_WKS_DIR/_N_WELLE_WPD/WELLE(parametr1,parametr2,…)

4.2.3.9 Rozšíření cesty pro vyhledávání při volání podprogramu (CALLPATH)

Pomocí příkazu CALLPATH je možné rozšířit cestu pro vyhledávání při volání podprogramu. Tímto způsobem mohou být vyvolávány také podprogramy z adresáře obrobku, který není vybrán, aniž by bylo nutno udávat úplný absolutní název cesty tohoto podprogramu.

Další možnost použití se nabízí v provozním režimu EES „EES bez GDIR", jestliže se pro ukládání globálních podprogramů používá adresář v externí paměti pro programy. V tomto případě je možné pomocí příkazu CALLPATH rozšířit cestu pro vyhledávání o tento adresář pro podprogramy.

Rozšíření cesty pro vyhledávání se provádí před zadáním pro uživatelské cykly (_N_CUS_DIR).

Následující události způsobí, že se rozšíření cesty pro vyhledávání opět deaktivuje:

- Příkaz CALLPATH s mezerami
- Příkaz CALLPATH bez parametrů
- Konec výrobního programu
- Reset

Syntaxe

```
CALLPATH("<název cesty>")
```

Význam

CALLPATH:	Klíčové slovo pro programovatelné rozšíření cesty pro vyhledávání Musí být naprogramováno na samostatném řádku výrobního programu.
<název cesty>:	Konstanta / proměnná typu STRING. Obsahuje absolutní údaj cesty do adresáře, o který má být cesta pro vyhledávání rozšířena. Pokud budete potřebovat informace o pravidlech pro zadávání cesty, viz "Adresování souborů v paměti pro programy (Strana 559)".

Příklad:

Cesta pro vyhledávání by měla být rozšířena o adresář příslušného obrobku:

Programový kód

```
...
CALLPATH("/_N_WKS_DIR/_N_MYWPD_WPD")
...
```

Tímto příkazem se vytvoří následující cesta pro vyhledávání (položka 5 je nová):

1. Aktuální adresář/*název*
2. Aktuální adresář/*název_SPF*
3. Aktuální adresář/*název_MPF*
4. //NC:/_N_SPF_DIR/*název_SPF*
5. **/_N_WKS_DIR/_N_MYWPD_WPD/název_SPF**
6. /N_CUS_DIR/*název_SPF*
7. /_N_CMA_DIR/*název_SPF*
8. /_N_CST_DIR/*název_SPF*

Okrajové podmínky

- Příkaz CALLPATH kontroluje, zda naprogramovaný název cesty skutečně existuje. Vyskytne-li se chyba, zpracování výrobního programu se přeruší a aktivuje se alarm korekčního bloku 14009.
- Příkaz CALLPATH může být naprogramován i v souborech INI. Potom je v platnosti po celou dobu zpracovávání souboru INI (soubor WPD-INI nebo inicializační program pro data aktivní v NC systému, např. framy v 1. kanálu _N_CH1_UFR_INI). Pak se znovu obnoví původní cesta pro vyhledávání.

4.2.3.10 Zpracovávání externího podprogramu (EXTCALL)

Pomocí příkazu EXTCALL je možné výrobní program načíst a zpracovat z externí paměti.

Jako externí paměť jsou k dispozici:

- Lokální jednotka
- Síťová jednotka
- Jednotka připojená přes USB

Poznámka

Jako rozhraní pro zpracování externího programu nacházejícího se na jednotce připojené přes USB se smí používat pouze rozhraní USB na ovládacím panelu, příp. na TCU.

UPOZORNĚNÍ
Poškození nástroje/obrobku kvůli flash-disku s USB Pro zpracovávání externích podprogramů nedoporučujeme používat USB flash disky. Přerušení komunikace s USB Flash diskem v průběhu zpracovávání výrobního programu kvůli komplikacím s kontakty, výpadkům, přerušením v důsledku nárazu nebo kvůli vytažení v důsledku neopatrnosti má za následek okamžité přerušení zpracovávání. Nástroj a/nebo obrobek by se přitom mohly poškodit.

Předdefinované nastavení cesty k externímu programu

Cesta do externího programového adresáře může být předem určena pomocí nastavovaného parametru:

SD42700 $SC_EXT_PROG_PATH

Spolu s cestou a s identifikátorem programu, které jsou zadány při volání pomocí příkazu EXTCALL, tak vzniká celková cesta k volanému výrobnímu programu.

Poznámka

Jestliže má být adresář programu (cesta) udáván pouze prostřednictvím volání příkazem EXTCALL, musí být parametr SD42700 prázdný!

Poznámka

Parametr

Při vyvolávání externího programu mu není možné předávat žádné parametry.

Syntaxe

```
EXTCALL("<cesta/><název programu>")
```

Význam

EXTCALL:	Příkaz pro vyvolání externího podprogramu	
"<cesta/><název programu>":	Konstanta / proměnná typu STRING	
	<cesta/>:	Absolutní nebo relativní zadání cesty (**není nutné**)
	<název programu>:	Název programu se zadává bez předpony "_N_". Příponu souboru ("MPF", "SPF") je možné spolu se znaky "_" nebo "." v názvu programu uvést (**není nutné**). Příklad: "WELLE" "WELLE_SPF" "WELLE.SPF"

Zadání cesty: Zkrácené popisy

Při zadávání cesty se mohou používat následující zkrácené popisy:

- Lokální jednotka: "**LOCAL_DRIVE:**"
- Paměťová karta: "**CF_CARD:**"
- Jednotka připojená přes USB (ovládací panel): "**USB:**"

Jako alternativu můžete používat také zkrácená označení „CF_CARD:“ a „LOCAL_DRIVE:“.

Příklad

Zpracovávání z lokální jednotky

Hlavní program "MAIN.MPF" se nachází v paměti NC systému a je vybrán pro zpracování:

Podprogram "SP_1"

Externí podprogram "SP_1.SPF", příp. "SP_1.MPF", se nachází na lokální jednotce v adresáři "/user/sinumerik/data/prog/WKS.DIR/WST1.WPD".

Cestu do externího adresáře programů je zapotřebí nastavit pomocí:

SD42700 $SC_EXT_PROG_PATH = LOCAL_DRIVE:WKS.DIR/WST1.WPD

Poznámka

Údaj cesty pro volání externích podprogramů:

- Bez použití předdefinovaného nastavení: "LOCAL_DRIVE:WKS.DIR/WST1.WPD/SP_1"
- S použitím předdefinovaného nastavení: "SP_1"

Podprogram "SP_2"

Externí podprogram "SP_2.SPF", příp. "SP_2.MPF" se nalézá v adresáři WKS.DIR/WST1.WPD na jednotce připojené přes USB. Pro cestu do podprogramu "SP_1" se použije předdefinované nastavení cesty do externího adresáře programů a také v hlavním programu se neuvádí. Z tohoto důvodu musí být při volání podprogramu "SP_2" uvedena plná cesta.

Hlavní program "MAIN"

Programový kód
N010 PROC MAIN
N020 ...
N030 EXTCALL("SP_1")
N030 EXTCALL("USB:WKS.DIR/WST1.WPD/SP_2")
N050 ...
N060 M30

Další informace

Volání příkazem EXTCALL s absolutním udáním cesty

Pokud je podprogram k dispozici v adresáři podle zadané cesty, voláním příkazem EXTCALL se spustí jeho zpracování. Pokud podprogram v adresáři podle zadané cesty k dispozici není, zpracování programu se voláním pomocí příkazu EXTCALL zastaví.

Volání příkazem EXTCALL s relativním udáním cesty / bez udání cesty

V případě volání pomocí příkazu EXTCALL s relativním zadáním cesty, příp. bez udání cesty, jsou stávající programové paměti prohledávány podle následujícího vzoru:

1. Jestliže je v nastavovaném parametru SD42700 $SC_EXT_PROG_PATH předem stanovena nějaká cesta, bude po zadání volání pomocí příkazu EXTCALL (název programu, příp. s relativním udáním cesty) vyhledávání začínat v rámci této cesty. Absolutní cesta potom vznikne zřetězením následujících posloupností znaků:
   - Předem definované udání cesty v parametru SD42700 $SC_EXT_PROG_PATH
   - Oddělovací znak "/"
   - Zadání cesty a název podprogramu v příkazu EXTCALL
2. Pokud nebyl pomocí postupu uvedeného v bodě 1. nalezen, jsou prohledány adresáře v uživatelské paměti.

Vyhledávání skončí, jakmile je podprogram poprvé nalezen. Pokud podprogram není nalezen, zpracování programu se voláním pomocí příkazu EXTCALL zastaví.

Nastavitelná paměť pro načítání (vyrovnávací paměť FIFO)

Pro zpracovávání externích podprogramů je zapotřebí vyrovnávací paměť pro načítání. Velikost této paměti je předem nastavena na 30 kbytů a tato velikost může být změněna pouze výrobcem stroje (prostřednictvím MD18360 MM_EXT_PROG_BUFFER_SIZE).

Poznámka

Podprogramy s příkazy skoků

U externích podprogramů, které obsahují příkazy skoku (GOTOF, GOTOB, CASE, FOR, LOOP, WHILE, REPEAT, IF, ELSE, ENDIF atd.), se musí cíle skoků nacházet uvnitř vyrovnávací paměti pro načítání tohoto programu.

Tato podmínka představuje problém především u příkazů skoku na začátek programu (GOTOS), protože tyto programy jsou zpravidla příliš velké, takže se pro ně ve vyrovnávací paměti nenajde dost místa. Při prvním načtení se začátek programu z vyrovnávací paměti odstraní. Jestliže při následném zpracovávání programu dojde na provedení příkazu skoku na začátek programu, potom funkce není schopna cíl skoku najít. Dojde k přerušení programu a aktivování alarmu 14000.

Aby bylo možné zpracovávat externí podprogramy bez omezení, pokud jde o naprogramované příkazy skoků, doporučujeme, abyste místo funkce „Zpracování externích podprogramů (EXTCALL)“ používali funkci „Zpracování z externí paměti (EES)“.

Poznámka

Programy ShopMill/ShopTurn

Programy ShopMill a ShopTurn musí být kvůli popisům kontur, které jsou vkládány až na konec souboru, uloženy v paměti pro načítání programů celé.

Pro paralelně zpracovávané externí podprogramy je rovněž zapotřebí vlastní vyrovnávací paměť pro načítání.

Reset / Konec programu / POWER ON

Operace RESET a POWER ON způsobují přerušení externího volání podprogramu a vymazání obsahu vyrovnávací paměti, kam se program načítá.

Podprogram vybraný pro "zpracování z externího zdroje" zůstává pro tuto funkci vybrán, i když je proveden RESET, POWER ON nebo nastane konec programu. Toto chování se neliší vůči interně vybraným programům, neboť externí paměť programů zůstává i nadále k dispozici.

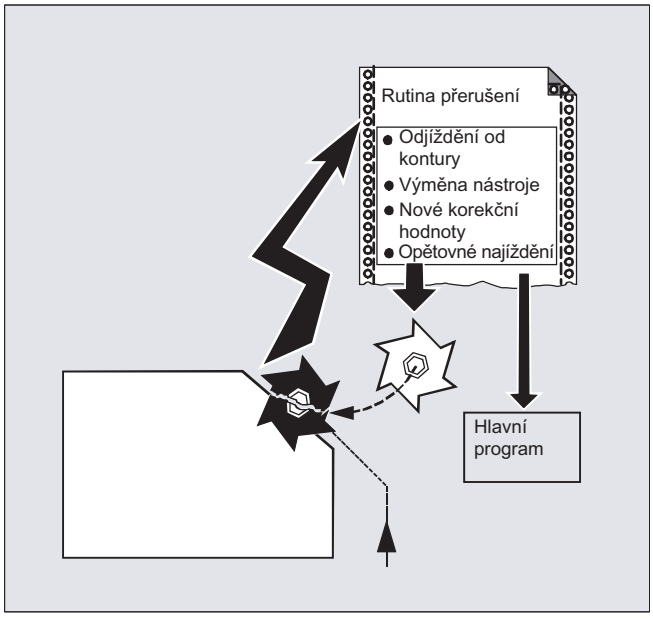
4.3 Rutiny přerušení (ASUP)

4.3.1 Funkce rutiny přerušení

Poznámka

Pojmy "asynchronní podprogram (ASUP)" a "Rutina přerušení", které se střídavě vyskytují v následujícím popisu, označují stejnou funkci.

Funkce rutiny přerušení bude objasněna na základě typického příkladu:



V průběhu zpracování se zlomí nástroj. V důsledku toho se aktivuje signál, který momentálně probíhající obráběcí operaci zastaví a současně spustí podprogram - tak zvanou rutinu přerušení. V tomto podprogramu se nacházejí všechny příkazy, které mají být v tomto případě provedeny.

Jakmile je podprogram zpracován (a v důsledku toho je obnovena připravenost k další práci), skočí řídící systém zpátky do hlavního programu a obrábění bude pokračovat - po aktivování příkazu REPOS - na místě, kde došlo k přerušení (viz "Opětovné najíždění na konturu (Strana 866)").

⚠ POZOR

Nebezpečí kolize

Jestliže v podprogramu není žádný příkaz REPOS naprogramován, potom se najede do polohy odpovídající koncovému bodu bloku, který odpovídá přerušenému bloku.

4.3.2 Sestavování rutin přerušení

Sestavení rutiny přerušení jako podprogramu

Při definici je rutina přerušení označena jako podprogram.

Příklad:

Programový kód	Komentář
`PROC ABHEB_Z`	`; název programu "ABHEB_Z"`
`N10 ...`	`; potom následují NC bloky`
`...`	
`N50 M17`	`; v závěru konec programu a návrat do hlavního programu`

Modální ukládání příkazu G-funkce (SAVE)

Při definici může být rutina přerušení označena atributem SAVE.

Atribut SAVE způsobuje, že modální příkazy G-funkcí, které byly před voláním rutiny přerušení aktivní, se uloží a po skončení rutiny přerušení se znovu aktivují (viz "Podprogramy s mechanismem SAVE (SAVE) (Strana 504)").

Díky tomu je možné, aby obrábění po zpracování rutiny přerušení pokračovalo od místa, kde k přerušení došlo.

Příklad:

Programový kód
`PROC ABHEB_Z SAVE`
`N10 ...`
`...`
`N50 M17`

Další přiřazení rutin přerušení (SETINT)

V rámci rutiny přerušení mohou být naprogramovány příkazy SETINT (viz "Přiřazování a spouštění rutin přerušení (SETINT)" (Strana 545)), takže je možné spouštět další vnořené rutiny přerušení. Spouštění se uskutečňuje až prostřednictvím vstupu.

4.3.3 Přiřazování a spouštění rutin přerušení (SETINT, PRIO, BLSYNC)

Řídící systém disponuje několika rychlými vstupy (vstup 1 ... 8), které spouštění příslušné přerušení (1 ... 8). Každému přerušení je možno pomocí příkazu `SETINT` přiřadit prioritu a příslušnou rutinu přerušení. Jestliže je spuštěno přerušení aktivováním určitého rychlého vstupu, aktuální zpracování v kanálu se přeruší a spustí se odpovídající rutina přerušení.

Priorita přerušení

Pokud jsou v jednom výrobním programu přiřazena přerušení několika vstupům, musí být jednotlivým přerušením přiřazeny různé priority.

Jednomu přerušení může být přiřazena hodnota priority 1 ... 128. Hodnota 1 odpovídá nejvyšší, 128 nejnižší prioritě.

Syntaxe

```
SETINT(<n>) <NAME>
SETINT(<n>) PRIO=<hodnota> <NAME>
SETINT(<n>) PRIO=<hodnota> <NAME> BLSYNC
SETINT(<n>) PRIO=<hodnota> <NAME> LIFTFAST
```

Význam

SETINT(<n>):	Signál přerušení <n> je přiřazen NC programu (ASUP) < název>. Přiřazená rutina přerušení se spouští, jakmile je rozpoznán signál přerušení <n> == 1. **Poznámka:** Jestliže je signálu přerušení <n> přiřazena jiná rutina přerušení, stává se předcházející přiřazení neplatným.	
<n>:	Číslo signálu přerušení	
	Typ:	INT
	Rozsah hodnot:	1 ... 32
PRIO=:	Priorita přerušení (nepovinné)	
<hodnota>:	(**nepovinné**) Hodnota priority	
	Typ:	INT
	Rozsah hodnot:	1 ... 128 (1 ⇒ nejvyšší priorita)
<NAME>:	Název NC programu (ASUP)	
BLSYNC:	(**nepovinné**) Příkaz BLSYNC způsobuje, že po spuštění přerušení se napřed počká, dokud nebude zpracován aktuální blok. Teprve potom se spustí rutina přerušení.	
LIFTFAST:	(**nepovinné**) Příkaz LIFTFAST způsobuje, že po spuštění přerušení se napřed uskuteční rychlé pozvednutí nástroje (viz kapitola "Rychlé pozvednutí od kontury (SETINT LIFTFAST, ALF) (Strana 549)"). Teprve potom se spustí rutina přerušení.	

Okrajové podmínky

Pravidla pro přerušení

1. Pro každé přerušení, které nemůže být okamžitě zpracováno nebo které je v daném okamžiku už zpracováváno, se do paměti uloží další požadavek na přerušení. Další požadavky na toto stejné přerušení jsou ztraceny.
2. Pokud je momentálně zpracováváno nějaké přerušení a pokud se aktivuje další přerušení s vyšší prioritou, přeruší toto přerušení s nižší prioritou. Po ukončení přerušení s vyšší prioritou pokračuje zpracování přerušení s nižší prioritou. Jestliže se během zpracování přerušení s vyšší prioritou objeví další požadavky pro přerušení s nižší prioritou, ukládají se tyto požadavky do paměti. Další se ztratí.
3. Pokud je momentálně zpracováváno nějaké přerušení a pokud se aktivuje další přerušení s vyšší prioritou, přeruší toto přerušení s nižší prioritou. Zpracuje se přerušení s vyšší prioritou. Pokud se znovu aktivuje přerušení s vyšší prioritou, zpracování aktuálního přerušení se pozastaví a zpracuje se přerušení s vyšší prioritou. Může existovat maximálně šest aktivních úrovní přerušení. Jedna momentálně zpracovávaná úroveň přerušení a pět čekajících úrovní přerušení. Pro každou aktivní úroveň přerušení se ukládá maximálně jeden další požadavek na přerušení. Všechny další požadavky na přerušení jsou ztraceny. Ztraceny jsou také požadavky na přerušení, jestliže by pro ně byly požadovány další úrovně přerušení (úroveň přerušení ≥ 7).

Příklady

Příklad 1: Přiřazení rutin přerušení a definice priorit

Programový kód	**Komentář**
`N20 SETINT(3) PRIO=1 ABHEB_Z`	`; IF vstup 3 == 1`
	`; THEN spustit rutinu přerušení "ABHEB_Z"`
`N30 SETINT(2) PRIO=2 ABHEB_X`	`; IF vstup 2 == 1`
	`; THEN spustit rutinu přerušení „ABHEB_X"`

Jestliže se vstupy aktivují současně, jsou rutiny přerušení zpracovávány postupně po sobě v posloupnosti podle svých hodnot priority: napřed "ABHEB_Z", pak "ABHEB_X".

Příklad 2: Nové přiřazení rutiny přerušení

Programový kód	**Komentář**
`N20 SETINT(3) PRIO=2 ABHEB_Z`	`; IF vstup 3 == 1`
	`; THEN spustit rutinu přerušení` **`"ABHEB_Z"`**
`...`	
`N80 SETINT(3) PRIO=1 ABHEB_X`	`; IF vstup 3 == 1`
	`; THEN spustit rutinu přerušení „`**`ABHEB_X"`**

4.3.4 Deaktivování/opětovné aktivování přiřazení rutiny přerušení (DISABLE, ENABLE)

Příkaz `SETINT` může být pomocí příkazu `DISABLE` deaktivován a pomocí příkazu `ENABLE` opět aktivován, aniž by se přiřazení vstup --> rutina přerušení ztratilo.

Syntaxe

```
DISABLE(<n>)
ENABLE(<n>)
```

Význam

`DISABLE(<n>)`:	Příkaz: **Deaktivování** přiřazení rutiny přerušení pro vstup <n>.	
`ENABLE(<n>)`:	Příkaz: **Opětovné aktivování** přiřazení rutiny přerušení pro vstup <n>.	
<n>:	Parametr: Číslo signálu přerušení	
	Typ:	INT
	Rozsah hodnot:	1 ... 32

Příklad:

Programový kód	**Komentář**
`N20 SETINT(3) PRIO=1 ABHEB_Z`	`; jestliže je aktivován vstup 3, pak by se`
	`; měla spustit rutina přerušení "ABHEB_Z"`
`...`	
`N90 DISABLE(3)`	`; příkaz SETINT z bloku N20 se deaktivuje`
`...`	
`N130 ENABLE(3)`	`; příkaz SETINT z bloku N20 se znovu aktivuje aktivuje`
`...`	

4.3.5 Vymazání přiřazení rutiny přerušení (CLRINT)

Přiřazení signálu přerušení k nějakému NC programu (ASUP), které bylo definováno příkazem `SETINT`, může být vymazáno příkazem `CLRINT`.

Syntaxe

```
CLRINT(<n>)
```

Význam

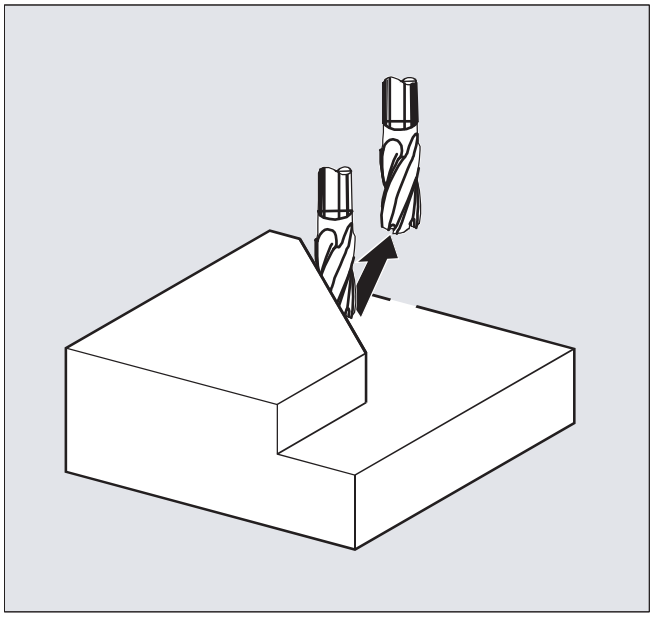
CLRINT(<n>):	Příkaz: Vymazání přiřazení signálu přerušení <n> NC programu (ASUP) <n> definovanému pomocí příkazu SETINT.	
<n>:	Parametr: Číslo signálu přerušení	
	Typ:	INT
	Rozsah hodnot:	1 ... 32

Příklad:

Programový kód	**Komentář**
N20 SETINT(3) PRIO=2 ABHEB_Z	
...	
N50 CLRINT(3)	; přiřazení mezi vstupem "3" a rutinou přerušení "ABHEB_Z" se vymaže

4.3.6 Rychlé pozvednutí od kontury (SETINT LIFTFAST, ALF)

Pokud je příkaz SETINT naprogramován s parametrem LIFTFAST, pak když se aktivuje vstup, nástroj se rychlým pozvednutím odsune od kontury obrobku.



Další postup závisí na tom, zda příkaz SETINT obsahuje vedle parametru LIFTFAST ještě i rutinu přerušení.

S rutinou přerušení: **Po** rychlém pozvednutí se spustí rutina přerušení.

Bez rutiny přerušení: Po rychlém pozvednutí se zpracování zastaví a aktivuje se alarm.

Syntaxe

```
SETINT(<n>) PRIO=1 LIFTFAST
SETINT(<n>) PRIO=1 <NAME> LIFTFAST
```

Význam

SETINT(<n>):	Příkaz: Přiřazení vstupu <n> rutině přerušení. Jakmile se vstup <n> aktivuje, spustí se přiřazená rutina přerušení.	
<n>:	Parametr: Číslo vstupu	
	Typ:	INT
	Rozsah hodnot:	1 ... 8
PRIO= :	Definice priority	
<hodnota>:	Hodnota priority	
	Rozsah hodnot:	1 ... 128
	Hodnota 1 odpovídá nejvyšší prioritě.	
<NAME>:	Název podprogramu (rutiny přerušení), který má být zpracován.	
LIFTFAST:	Příkaz: Rychlé pozvednutí od kontury	
ALF=... :	Příkaz: Programovatelný směr pohybu (nachází se v pohybovém bloku) Pokud budete potřebovat informace o možnostech programování s příkazem ALF, viz kapitola " Směr pohybu při rychlém pozvednutí od kontury (Strana 551) ".	

Okrajové podmínky

Chování v případě aktivního framu se zrcadlovým převrácením

Při určování směru pozvednutí se kontroluje, zda není aktivní nějaký frame se zrcadlovým převrácením. V takovém případě se u směru pozvedávání vztaženého na směr tečny prohodí směry vlevo a vpravo. Směrové složky v nasměrování nástroje se zrcadlově nepřevracejí. Toto chování se aktivuje nastavením strojního parametru:

MD21202 $MC_LIFTFAST_WITH_MIRROR = TRUE

Příklad:

Zlomený nástroj má být automaticky nahrazen náhradním nástrojem. Opracovávání bude potom pokračovat s novým nástrojem.

Hlavní program:

Hlavní program	**Komentář**
N10 SETINT(1) PRIO=1 W_WECHS LIFTFAST	; když se aktivuje vstup 1, bude nástroj rychlým pozvednutím okamžitě odsunut od kontury (kód č. 7 pro korekci rádiusu nástroje G41) Potom se zpracuje rutina přerušení "W_WECHS".
N20 G0 Z100 G17 T1 ALF=7 D1	
N30 G0 X-5 Y-22 Z2 M3 S300	
N40 Z-7	

Hlavní program	**Komentář**
N50 G41 G1 X16 Y16 F200	
N60 Y35	
N70 X53 Y65	
N90 X71.5 Y16	
N100 X16	
N110 G40 G0 Z100 M30	

Podprogram:

Podprogram	**Komentář**
PROC W_WECHS SAVE	; podprogram s uložením aktuálního provozního stavu
N10 G0 Z100 M5	; poloha pro výměnu nástroje, zastavení vřetena
N20 T11 M6 D1 G41	; výměna nástroje
N30 REPOSL RMBBL M3	; opětovné najíždění na konturu a skok zpátky do hlavního programu (naprogramuje se v jednom bloku)

4.3.7 Směr pohybu při rychlém pozvednutí od kontury

Zpětný pohyb

Rovina zpětného pohybu je určena prostřednictvím následujících příkazů G-funkcí:

- LFTXT
  Rovina zpětného pohybu se určuje z tečny ke dráze a ze směru nástroje (standardní nastavení).
- LFWP
  Rovina zpětného pohybu je momentálně aktivní pracovní rovina, která je zvolena jedním z příkazů G-funkcí G17, G18 nebo G19. Směr zpětného pohybu je nezávislý na tečně ke dráze. Tímto způsobem může být naprogramováno rychlé pozvednutí rovnoběžně s osou.
- LFPOS
  Zpětný pohyb osy stanovené příkazem POLFMASK / POLFLIN na absolutní pozici naprogramovanou pomocí příkazu POLF.
  Příkaz ALF nemá žádný vliv na směr pozvednutí pro větší počet os, jakož i pro větší počet os v lineárním vztahu.

Programovatelný směr pohybu (ALF=...)

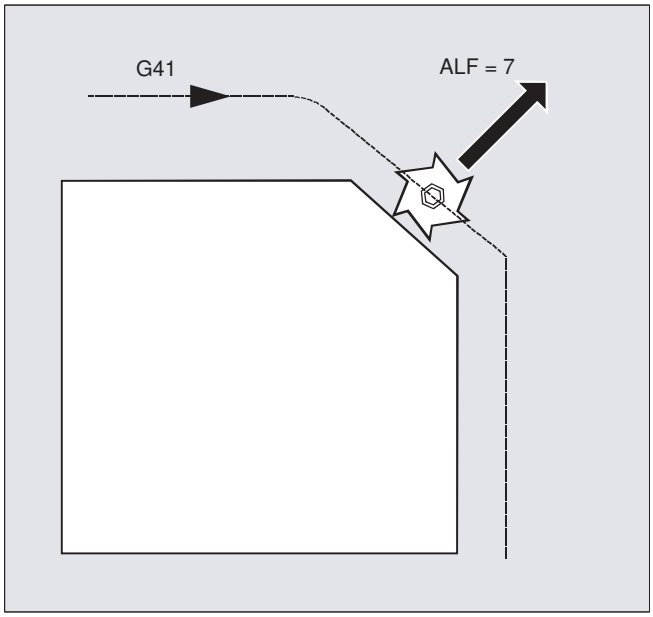
Směr se programuje v diskrétních úhlových krocích 45 stupňů pomocí příkazu ALF v rovině zpětného pohybu.

Možné směry pohybů jsou uloženy v řídícím systému pod speciálními kódovými čísly a pomocí těchto čísel je možné je vyvolat.

Příklad:

Programový kód
N10 SETINT(2) PRIO=1 ABHEB_Z LIFTFAST ALF=7

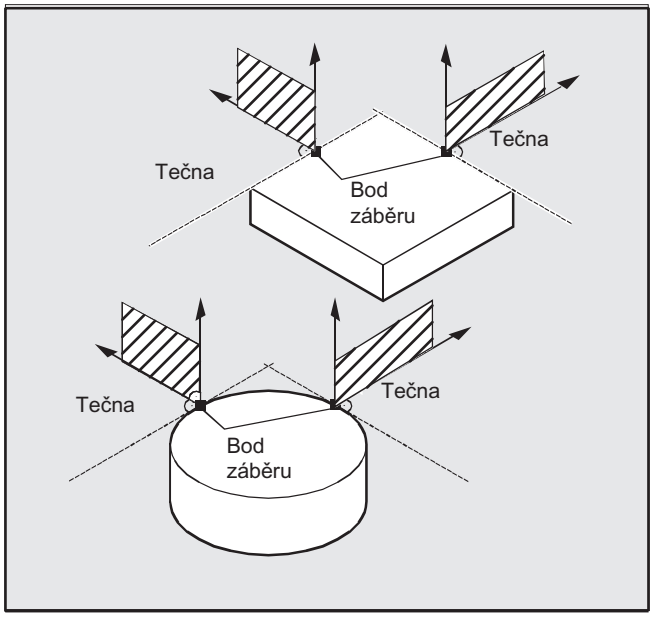
Když je aktivován příkaz `G41` (směr opracování vlevo od kontury), pohybuje se nástroj kolmo od kontury.



Vztažná rovina pro popis směru pohybu u příkazu LFTXT

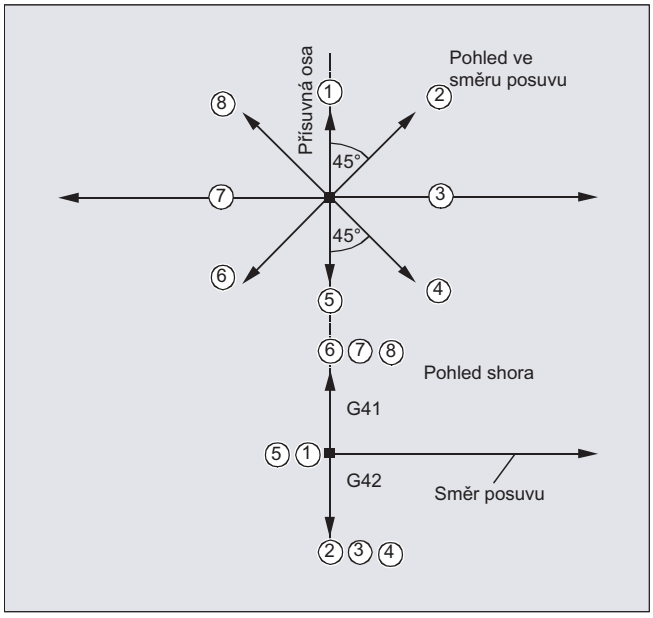
V bodě záběru nástroje na naprogramované kontuře je definována rovina, která slouží jako vztažná rovina pro udávání směru při pozvedávání pomocí odpovídajících kódových čísel.

Vztažná rovina je definována na základě osy přísuvu nástroje (směr přísuvu) a vektoru, který je kolmý jednak na tuto osu a jednak na tečnu v bodě záběru nástroje na kontuře.



Kódová čísla se směry pohybů u příkazu LFTXT

Na následujícím obrázku naleznete kódová čísla se směry pohybů, které se určují na základě vztažné roviny.



Pro `ALF=1` definován návrat ve směru nástroje.

Když je nastaveno ALF=0, je funkce "Rychlé pozvednutí" vypnuta.

⚠ POZOR

Nebezpečí kolize

Když je aktivována korekce rádiusu nástroje:

- v případě příkazu G41 by se kódy 2, 3, 4
- v případě příkazu G42 by se kódy 6, 7, 8

neměly používat, protože v těchto případech nástroj přijíždí ke kontuře a dostal by se do kolize s obrobkem.

Kódová čísla se směry pohybů u příkazu LFWP

V případě LFWP je směr v pracovní rovině přiřazen podle následujícího uspořádání:

- G17: Rovina X/Y
  ALF=1: Zpětný pohyb ve směru X
  ALF=3: Zpětný pohyb ve směru Y
- G18: Rovina Z/X
  ALF=1: Zpětný pohyb ve směru Z
  ALF=3: Zpětný pohyb ve směru X
- G19: Rovina Y/Z
  ALF=1: Zpětný pohyb ve směru Y
  ALF=3: Zpětný pohyb ve směru Z

Viz také

Rychlý zpětný pohyb během řezání závitu (LFON, LFOF, DILF, ALF, LFTXT, LFWP, LFPOS, POLF, POLFMASK, POLFMLIN) (Strana 233)

4.3.8 Posloupnost pohybů při rutinách přerušení

Rutina přerušení bez příkazu LIFTFAST

Pohyby os na dráze se zabrzdí až do zastavení. Potom se spustí rutina přerušení.

Poloha, kde došlo k zabrždění, se uloží jako místo, kde došlo k přerušení, a při aktivování příkazu REPOS s RMIBL na konci rutiny přerušení se na ni najede.

Rutina přerušení s příkazem LIFTFAST

Pohyby os na dráze se zabrzdí. Současně se jako superponovaný pohyb uskuteční pohyb LIFTFAST. Jakmile se pohyb po dráze a pohyb LIFTFAST úplně zastaví, spustí se rutina přerušení.

Jako místo, kde došlo k přerušení, se uloží pozice na kontuře, na které byl zahájen pohyb LIFTFAST a na které byla tedy dráha opuštěna.

Rutina přerušení s parametry `LIFTFAST` a `ALF=0` se chová identicky jako rutina přerušení bez příkazu `LIFTFAST`.

Poznámka

Vzdálenost, o kterou geometrické osy odjedou při rychlém pozvednutí od kontury, může být nastavena pomocí strojního parametru.

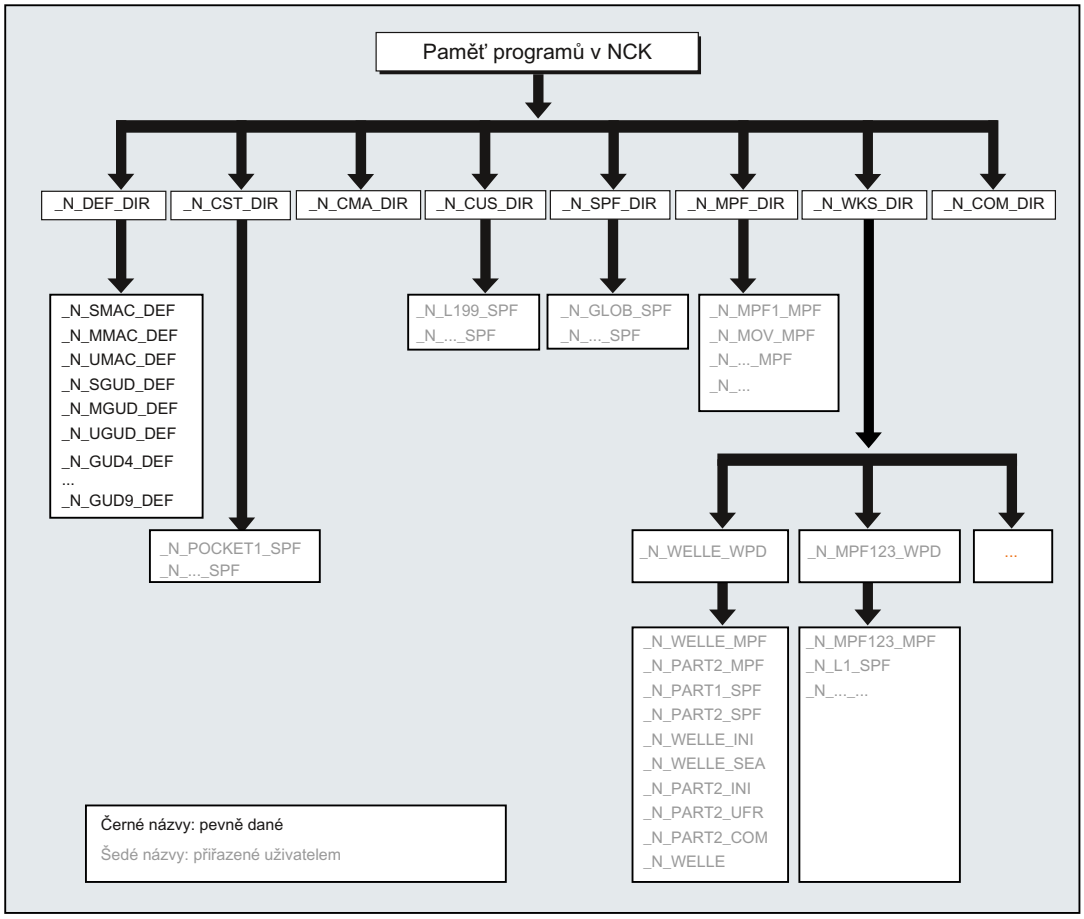
4.4 Správa souborů a programů

4.4.1 Programová paměť

4.4.1.1 Paměť programů v NCK

V paměti programů jsou trvale uloženy soubory a programy (např. hlavní programy a podprogramy, definice maker) (--> pasivní systém souborů).

Kromě toho existuje určitý počet typů souborů, které zde mohou být dočasně uloženy a v případě potřeby (např. při opracovávání určitých obrobků) jsou přeneseny do pracovní paměti (např. pro účely inicializace).



Standardní adresáře

Standardně jsou k dispozici následující adresáře:

Adresář	Obsah
_N_DEF_DIR	Datové moduly a moduly maker
_N_CST_DIR	Standardní cykly
_N_CMA_DIR	Cykly výrobce
_N_CUS_DIR	Uživatelské cykly
_N_WKS_DIR	Obrobky
_N_SPF_DIR	Globální podprogramy
_N_MPF_DIR	Hlavní programy
_N_COM_DIR	Komentáře

Typy souborů

Do paměti programů se mohou přenášet následující typy souborů:

Typ souboru	Popis
<název>_MPF	Hlavní program
<název>_SPF	Podprogram
<název>_TEA	Strojní parametry
<název>_SEA	Nastavované parametry
<název>_TOA	Korekční parametry nástroje
<název>_UFR	Posunutí počátku / frame
<název>_INI	Inicializační soubory
<název>_GUD	Globální uživatelská data
<název>_RPA	R-Parametry
<název>_COM	Komentář
<název>_DEF	Definice pro globální uživatelská data a makra

Hlavní adresář obrobků (_N_WKS_DIR)

Hlavní adresář obrobků je standardně vytvořen v paměti programů pod označením _N_WKS_DIR. Hlavní adresář obrobků obsahuje všechny obrobky, které máte naprogramovány, a odpovídající adresáře obrobků.

Adresáře obrobků (..._WPD)

Adresář obrobku obsahuje soubory, které jsou nezbytné pro opracovávání obrobku. Může se jednat o hlavní programy, podprogramy, libovolné inicializační programy a soubory komentářů.

Inicializační programy se po zvolení příslušného programu při prvním spuštění výrobního programu jedenkrát zpracují (v závislosti na strojním parametru MD11280 $MN_WPD_INI_MODE).

Příklad:

Adresář obrobku _N_WELLE_WPD, který byl založen pro obrobek WELLE, obsahuje následující soubory:

Soubor	Popis
_N_WELLE_MPF	Hlavní program
_N_PART2_MPF	Hlavní program
_N_PART1_SPF	Podprogram
_N_PART2_SPF	Podprogram
_N_WELLE_INI	Všeobecný inicializační program pro data daného obrobku
_N_WELLE_SEA	Inicializační program nastavovaných parametrů
_N_PART2_INI	Všeobecný inicializační program pro data programu PART2
_N_PART2_UFR	Inicializační program pro data framu pro program PART2
_N_WELLE_COM	Soubor komentářů

Kromě toho mohou být v adresáři obrobku ukládána také data, která nejsou bezprostředně zapotřebí pro zpracování programu prostřednictvím NC systému. Vedle souborů ASCII to mohou být také binární soubory, jako např. obrázky ve formátu JPG nebo popisy ve formátu PDF. Protože NC systém by mohl tyto objekty interpretovat jako binární soubory, musí být přípony souborů v NC systému známy (nastavení v rámci uvádění do provozu pomocí parametru MD17000 $MN_ EXTENSIONS_OF_BIN_FILES; v základním nastavení jsou předem definovány následující přípony souborů: JPG, GIF, PNG, BMP, PDF, ICO, HTM).

Volba obrobku pro zpracování

Adresář obrobku může být vybrán pro zpracování v určitém kanálu, Pokud se v tomto adresáři nachází hlavní program **stejného názvu** nebo jen jediný hlavní program (_NPF), bude tento program automaticky vybrán pro zpracování

4.4.1.2 Externí programová paměť

Kromě pasivního systému souborů v NC mohou být na stroji k dispozici také externí programové paměti (např. na lokálním pevném disku nebo na síťové jednotce).

Prostřednictvím funkce "Zpracování z externího zdroje" nebo "EES (Execution from External Storage)" mohou být výrobní programy zpracovávány **přímo** z externí paměti programů.

Další informace: Příručka k funkcím, Základní funkce

Globální paměť výrobních programů (GDIR)

Při konfiguraci diskových jednotek může být jedna jednotka vybrána jako globální paměť výrobních programů (GDIR).

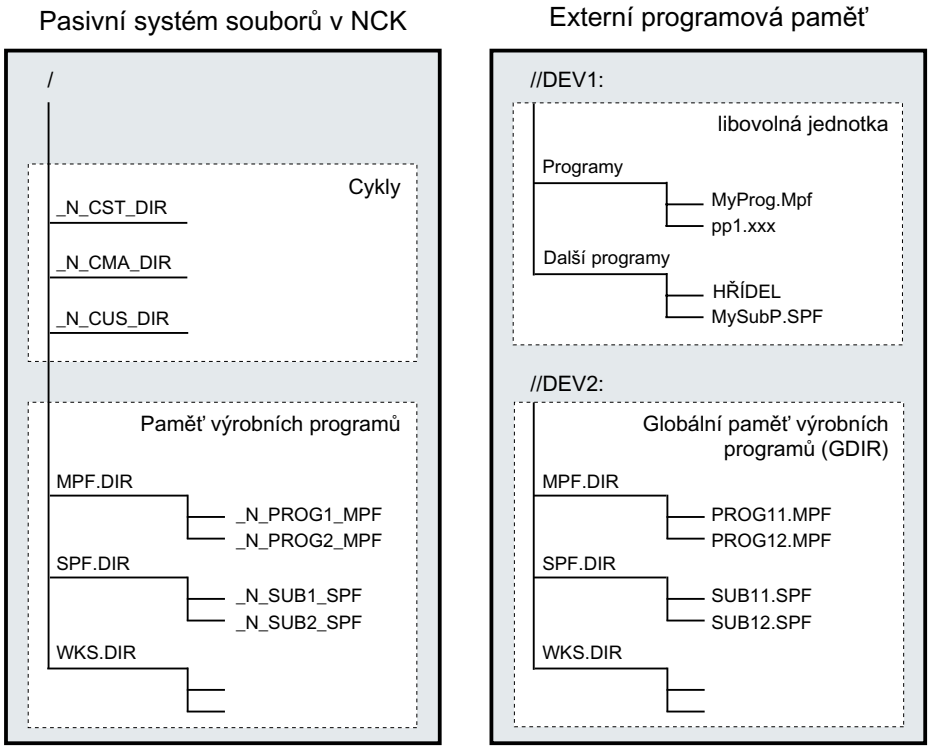
Systém potom na diskové jednotce automaticky založí adresáře MPF.DIR, SPF.DIR a WKS.DIR. Tyto tři adresáře tvoří paměť GDIR.

GDIR hraje roli výlučně pro fungování EES. V závislosti na konfiguraci jednotek GDIR nahrazuje nebo rozšiřuje GDIR paměti výrobních programů NC systému. Zřízení paměti GDIR není ale pro fungování EES bezpodmínečně nutné.

Adresáře a soubory v paměti GDIR mohu být ve výrobním programu adresovány stejným způsobem, jako v pasivním systému souborů. Díky tomu je možné kompatibilní přenášení NC programů s údajem cesty z pasivního systému souborů do GDIR. Adresář SPF.DIR v paměti GDIR je obsažen v cestě pro vyhledávání podprogramů.

Organizace programů

Organizaci programů v externí paměti programů má ozřejmit následující obrázek:



Systém souborů nerozlišuje velká a malá písmena

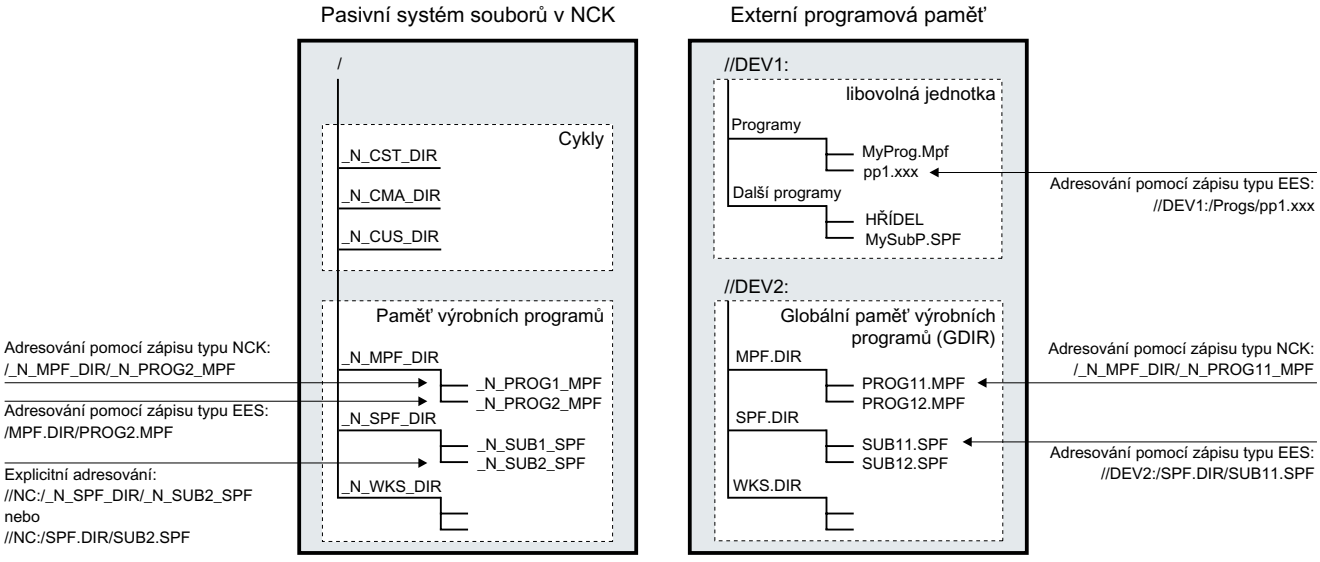
Poznámka

Aby se při adresování souborů zabránilo problémům v důsledku zápisu velkých a malých písmen (viz „Adresování souborů v paměti pro programy (Strana 559)“), měl by se jako externí programová paměť používat systém souborů, který **nerozlišuje velká a malá písmena**.

4.4.1.3 Adresování souborů v paměti pro programy

Soubor v paměti pro programy, na který je směrován příkaz pro manipulaci se soubory (např. WRITE, DELETE, READ, ISFILE, FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO), je popsán buď prostřednictvím absolutní cesty plus názvu souboru nebo jen prostřednictvím samotného názvu souboru. Ve druhém případě se jako cesta k souboru použije cesta vybraného programu.

Adresování pomocí zápisu typu NC/EES



Adresování souborů v pasivním systému souborů

Adresování souborů v pasivním systému souborů se uskutečňuje obvyklým způsobem v **zápisu ve formátu NC systému** (názvy adresářů a názvy souborů začínají identifikací domény "_N_", oddělovací znak pro příponu adresáře/souboru je "_") bez udání názvu jednotky. Adresování pomocí **způsobu zápisu EES** (bez identifikace domény "_N_", oddělovací znak pro příponu názvu adresáře/souboru je ".") je ale přípustné také.

Příklad:

- Způsob zápisu typu NC: "/_N_SPF_DIR/_N_SUB1_SPF"
- Způsob zápisu typu EES: "/SPF.DIR/SUB1.SPF"

Poznámka

Adresování souborů v pasivním systému souborů při způsobu zápisu typu EES se interně převádí do způsobu zápisu typu NC podle následujících pravidel:

- Názvy adresářů a názvy souborů jsou rozšířeny o identifikaci domény "_N_".
- Jestliže je čtvrtý znak od konce v názvu adresáře nebo v názvu souboru tečka ("."), je převedena na znak podtržení ("_").

Prostřednictvím předdefinovaných názvů jednotek "//NC:" je možno pasivní systém souborů adresovat také cíleně.

Příklad:

- Způsob zápisu typu NC: "//NC:/_N_SPF_DIR/_N_SUB1_SPF"
- Způsob zápisu typu EES: "//NC:/SPF.DIR/SUB1.SPF"

Adresování souborů v externí paměti pro programy

Adresování souborů v externí paměti pro programy, která není konfigurována jako paměť GDIR, se musí uskutečňovat pomocí způsobu typu EES. Na začátku adresovací cesty musí být uveden název jednotky (např. "//DEV1:“). Přípustné jsou všechny symbolické názvy jednotek, které jsou konfigurovány v souboru /user/sinumerik/hmi/cfg/logdrive.ini.

Příklad:

- Způsob zápisu typu EES: "//DEV1:/MyProgDir/pp1.xxx"
- Způsob zápisu typu NC: nepřípustný

Adresování souborů v globální paměti pro výrobní programy (GDIR)

Při adresování souborů v paměti GDIR je kromě udání cesty pomocí způsobu zápisu typu EES přípustné také zadání cesty pomocí způsobu zápisu typu NC.

Příklad:

- Způsob zápisu typu EES: "//DEV2:/MPF.DIR/PROG11.MPF"
- Způsob zápisu typu NC: "/_N_MPF_DIR/_N_PROG11_MPF"

Poznámka

Adresování souborů v paměti GDIR pomocí způsobu zápisu typu NC se interně převádí do způsobu zápisu typu EES podle následujících pravidel:

- Identifikace domény "_N_" v názvech adresářů a názvech souborů se odstraní.
- Jestliže je čtvrtý znak od konce v názvu adresáře nebo v názvu souboru znak podtržení („_"), je převeden na tečku („.“).

Pravidla pro zadávání cesty

Úplné udání cesty se skládá z názvu jednotky, cesty popisující adresář a z názvu souboru.

Název jednotky

Pro zadávání názvu jednotky platí následující pravidla:

- Přípustné jsou všechny symbolické názvy jednotek, které jsou konfigurovány v souboru /user/sinumerik/hmi/cfg/logdrive.ini.
- N začátku se nachází řetězec znaků "//", za kterým následuje minimálně jedno písmeno nebo číslice.
- Následující znaky mohou být libovolnou kombinací písmen, číslic, znaků podtržení ("_“) a mezer.
- Název musí být ukončen písmenem nebo číslicí, za nimiž následuje „:“.
- Další speciální znaky jsou nepřípustné.

Poznámka

Pro pasivní systém souborů je předem definován název jednotky "//NC:“.

Příklady:

- Externí programová paměť:
  - //Drive1:
  - //Drive_1:
  - //Drive 1:
  - //A B:
  - //1 B C 2:

Cesta do adresáře

Pro zadávání cesty do adresáře platí následující pravidla:

- Na začátku a na konci cesty adresáře a jako oddělovací znak pro jednotlivé složky cesty se používá znak "/".

  Poznámka

  Dvojitý znak podtržení ("//“) v rámci cesty adresáře je **nepřípustný**!

- Názvy adresářů:
  - Název adresáře musí začínat písmenem nebo číslicí. Jen v případě adresování pomocí zápisu typu NC začínají názvy adresářů identifikací domény "_N_".
  - Následující znaky mohou být libovolnou kombinací písmen, číslic a znaků podtržení ("_").

    Poznámka

    V případě externích programových pamětí jsou v názvech adresářů přípustné také mezery. To však neplatí, pokud je externí programová paměť konfigurována jako globální paměť výrobních programů (GDIR).

  - Další speciální znaky jsou nepřípustné.
- Přípony názvů adresářů:
  - Přípony názvů adresářů se musí skládat přesně ze třech písmen/číslic.
  - Jsou oddělovány buď znakem "_" (zápis typu NC) nebo "." (zápis typu EES) od názvu adresáře.

  Poznámka

  V pasivním systému souborů existují pouze přípony názvů adresářů _DIR a _WPD.

Příklady:

- Pasivní systém souborů, příp. GDIR:
  - Způsob zápisu typu NC: _N_WKS_DIR/_N_MYNCPROGS_WPD/...
  - Způsob zápisu typu EES: WKS.DIR/MYPROGS.WPD/...
- Externí programová paměť:
  - /abc
  - /ab_c.def
  - /ab c1.def
  - /a b c .d11
  - /abc.def/ghi.klm

Název souboru

Pro názvy souborů platí následující pravidla:

- Jen v případě adresování pomocí zápisu typu NC začínají názvy souborů identifikací domény "_N_".
- Následující dva znaky by měly být dvě písmena nebo znak podtržení, za kterým následuje písmeno.

  Poznámka

  Jestliže jsou tyto podmínky splněny, potom může být tento NC program samotný vyvoláván jako podprogram z nějakého jiného programu zadáním tohoto názvu programu. Jestliže název programu proti tomu začíná číslicemi, potom je volání podprogramu možné jedině pomocí příkazu CALL.

- Následující znaky mohou být libovolnou kombinací písmen, číslic a znaků podtržení ("_").
- Přípona názvu souboru:
  - Přípona názvu souboru se musí skládat přesně ze třech písmen/číslic.

    Poznámka

    Pokud budete potřebovat informace o příponách souborů v pasivním systému souborů, viz "Paměť programů v NCK (Strana 555)".

  - Jsou oddělovány bud znakem "_" (zápis typu NC) nebo "." (zápis typu EES) od názvu souboru.

Příklady:

- Pasivní systém souborů, příp. GDIR:
  - Způsob zápisu typu NC: _N_SUB1_SPF
  - Způsob zápisu typu EES: SUB1.SPF
- Externí programová paměť:
  - Teil1
  - _Teil1
  - Teil_1.spf
  - Teil1.mpf

Podprogramy podle normy DIN

Pro podprogramy podle normy DIN platí následující pravidla:

- Prvním znakem je písmeno „L“.
- Následující znaky jsou číslice (minimálně jedna).
- Přípona názvu souboru:
  - Přípona názvu souboru se musí skládat přesně ze třech písmen.
  - Jsou oddělovány bud znakem "_" (zápis typu NC) nebo "." (zápis typu EES) od názvu souboru.

Příklady:

- L123
- L1_SPF (způsob zápisu typu NC) příp. L1.SPF (způsob zápisu typu EES)

Maximální délka cesty

Pro zadání názvu jednotky a cesty do adresáře je k dispozici maximálně 128 bytů, délka řetězce udávajícího název souboru smí činit maximálně 31 bytů. Z toho vyplývá, že celá cesta může mít maximální délku 159 bytů.

4.4.1.4 Cesta pro vyhledávání při volání podprogramu

Při volání podprogramu bez udání cesty se absolutní cesta zjišťuje prostřednictvím zpracování pevně dané cesty pro vyhledávání.

Paměť pro programy je přitom prohledávána v následující posloupnosti:

		Adresář	Popis
1		aktuální adresář / *název*	Aktuálním adresářem je adresář, ve kterém je vybírán program. Může se jednat o následující: • adresář obrobku nebo standardní adresář _N_MPF_DIR v paměti výrobních programů NC systému, příp. globální paměť pro programy. nebo • libovolný adresář v externí paměti pro programy
2		aktuální adresář / *název_SPF*	
3		aktuální adresář / *název_MPF*	
4	a	//NC:/_N_SPF_DIR / *name_SPF*	Adresář podprogramu v paměti výrobních programů v NC systému
	b	//DEV2:/_N_SPF_DIR / název_SPF [1]	Adresář podprogramu v globální paměti výrobních programů **Upozornění:** Tento krok vyhledávání odpadá, pokud součástí konfigurace není žádná globální paměť výrobních programů nebo pokud vyhledávání probíhá v paměti pro programy v NC systému.
5		Rozšíření cesty pro vyhledávání naprogramované pomocí příkazu `CALLPATH` (viz "Rozšíření cesty pro vyhledávání při volání podprogramu (CALLPATH) (Strana 538)"). **Poznámka:** Jestliže není naprogramován žádný příkaz `CALLPATH`, tento krok vyhledávání odpadá.	
6		/_N_CUS_DIR / *název_SPF*	Adresář uživatelských cyklů
7		/_N_CMA_DIR / *název_SPF*	Adresář cyklů výrobce
8		/_N_CST_DIR / *název_SPF*	Adresář standardních cyklů

[1] //DEV2:" představuje například jednotku, na které je konfigurována globální paměť pro výrobní programy.

Pro vyhledávání platí následující pravidla:

- Cesta pro vyhledávání se prochází pro každé jednotlivé volání podprogramu, tzn. nezáleží na tom, kde se nachází nadřazený program.
- V závislosti na adresáři je potřeba mít na zřeteli různé typy souborů.
- V zásadě se v adresáři nehledá v podřízených podadresářích.

4.4.1.5 Zjištění cesty a názvu souboru

Pokud potřebujete zjistit cestu a název souboru NC programu, máte ve výrobním programu k dispozici následující systémové proměnné, které můžete načíst:

Systémová proměnná	Typ	Význam
$P_STACK	INT	Obsahuje programovou úroveň, na které je momentální NC program zpracováván.
$P_PATH[<n>]	STRING	Obsahuje cestu NC programu, který je zpracováván na programové úrovni zvolené indexem pole <n>. Příklady: $P_PATH[0] poskytuje cestu pro hlavní program, např. "/_N_WKS_DIR/_N_WELLE_WPD/". $P_PATH[$P_STACK - 1] obsahuje cestu volaného programu. Jestliže se cesta vztahuje na NC program, který je uložen v pasivním systému souborů NC nebo v globální paměti pro výrobní programy (GDIR), je pro cestu použit způsob zápisu typu NC systému. Jestliže se cesta vztahuje na NC program, který je zpracováván z nějaké jiné externí paměti pro programy, než je globální paměť pro výrobní programy, poskytuje proměnná $P_PATH cestu ve formátu EES.
$P_PROG[<n>]	STRING	Obsahuje název NC programu, který je zpracováván na programové úrovni zvolené indexem pole <n>. Jestliže je NC program uložen v pasivním systému souborů NC nebo v globální paměti pro výrobní programy, je pro název programu použit způsob zápisu typu NC. Jestliže je NC program zpracováván z nějaké jiné externí jednotky, než je globální paměť výrobních programů, poskytuje proměnná $P_PROG název ve způsobu zápisu typu EES.
$P_PROGPATH	STRING	Obsahuje cestu NC programu, který je právě zpracováván. Volání proměnné $P_PROGPATH je stejné jako proměnné $P_PATH[$P_STACK].

Systémová proměnná	Typ	Význam	
$P_IS_EES_PATH[<n>]	BOOL	Umožňuje zjistit, zda cesta poskytovaná proměnnou $P_PATH[<n>], příp. název programu v proměnné $P_PROG[<n>] odpovídá způsobu zápisu NC systému nebo EES.	
		= FALSE	$P_PATH[<n>] a $P_PROG[<n>] poskytují zápis typu NC. Tzn. každý identifikátor obsahuje počáteční řetězec „_N_“. Oddělovací znak pro identifikaci souborů je „_". Příklady: • Cesta s typem zápisu NC: "/_N_WKS_DIR/_N_MYWPD_WPD/" • Název programu s typem zápisu NC: "_N_MYPROG_MPF" Cesta ve způsobu zápisu typu NC se může vztahovat jak na pasivní systém souborů v NC, tak i na globální paměť pro výrobní programy.
		= TRUE	$P_PATH[<n>] a $P_PROG[<n>] poskytují zápis typu EES. Tzn. identifikátory neobsahují počáteční řetězec „_N_“. Oddělovací znak pro identifikaci souborů je „.". Příklady: • Cesta s typem zápisu EES: "//DEV1:/WKS.DIR/MYWPD.WPD/" • Název programu s typem zápisu EES: "MYPROG.MPF"

<n>: Index <n> definuje programovou úroveň, ze které se mají informace o cestě načítat (rozsah hodnot: 0 ... 17)

Poznámka

V režimu EES jsou kromě globální paměti pro výrobní programy (GDIR) systémovými proměnnými $P_PROG, $P_PATH, a $P_PROGPATH poskytovány názvy cest ve způsobu zápisu typu EES. Uživatelské programy, které tyto názvy cest vyhodnocují a dále zpracovávají, musí proto být pro režim EES rozšířeny tak, aby se i názvy cest daly zpracovávat ve způsobu zápisu typu EES.

4.4.2 Pracovní paměť (CHANDATA, COMPLETE, INITIAL)

Funkce

Pracovní paměť obsahuje aktuální systémová a uživatelská data, se kterými systém pracuje (aktivní systém souborů), např.:

- Aktivní strojní parametry
- Korekční parametry nástroje

- Posunutí počátku
- ...

Inicializační programy

V této souvislosti se jedná o programy, které zajišťují dosazování předdefinovaných hodnot pro data v pracovní paměti (inicializace). Za tím účelem se používají následující typy dat:

Typ souboru	Popis
název_TEA	Strojní parametry
název_SEA	Nastavované parametry
název_TOA	Korekční parametry nástroje
název_UFR	Posunutí počátku / frame
název_INI	Inicializační soubory
název_GUD	Globální uživatelská data
název_RPA	R-Parametry

Datové oblasti

Data mohou být rozčleněna do různých oblastí, v nichž mají platit. Řídící systém může například disponovat větším počtem kanálů nebo může být k dispozici více os, což je obvyklé.

Existují:

Označení	Datové oblasti
NC	Specifická data NC systému
CH<n>	Data specifického kanálu (<n> udává číslo kanálu)
AX<n>	Data specifické osy (<n> udává číslo osy stroje)
TO	Parametry nástroje
COMPLETE	Všechna data

Založení inicializačního programu na externím PC

Prostřednictvím identifikace datové oblasti a identifikace datového typu mohou být definovány oblasti, které jsou při ukládání dat považovány za jednu jednotku.

_N_AX5_TEA_INI	Strojní parametry pro osu 5
_N_CH2_UFR_INI	Framy kanálu 2
_N_COMPLETE_TEA_INI	Všechny strojní parametry

Po uvedení řídícího systému do provozu je v pracovní paměti k dispozici datový blok, který zaručuje, že řídící systém pracuje správně.

Postup v případě řídících systémů s více kanály (CHANDATA)

Příkaz CHANDATA(<číslo kanálu>) pro větší počet kanálů je přípustný pouze v souboru _N_INITIAL_INI. Jedná se o inicializační soubor, pomocí kterého se uskutečňuje inicializace všech dat v řídícím systému.

Programový kód	**Komentář**
%_N_INITIAL_INI	
CHANDATA(1)	
	; Přiřazení os stroje kanálu 1:
$MC_AXCONF_MACHAX_USED[0]=1	
$MC_AXCONF_MACHAX_USED[1]=2	
$MC_AXCONF_MACHAX_USED[2]=3	
CHANDATA(2)	
	; Přiřazení os stroje kanálu 2:
$MC_AXCONF_MACHAX_USED[0]=4	
$MC_AXCONF_MACHAX_USED[1]=5	
CHANDATA(1)	
	; Osové strojní parametry:
	; Okno přesného najetí hrubé:
$MA_STOP_LIMIT_COARSE[AX1]=0.2	; Osa 1
$MA_STOP_LIMIT_COARSE[AX2]=0.2	; Osa 2
	; Okno přesného najetí jemné:
$MA_STOP_LIMIT_FINE[AX1]=0.01	; Osa 1
$MA_STOP_LIMIT_FINE[AX1]=0.01	; Osa 2

UPOZORNĚNÍ

Příkaz CHANDATA

Ve výrobním programu smí být příkaz CHANDATA použit pouze v kanálu, v němž se daný NC program zpracovává, tzn. tento příkaz se může používat také pro ochranu NC programů, aby nemohly být zpracovávány v kanálu, pro který nejsou určeny.

Pokud se vyskytne chyba, zpracování programu se přeruší.

Poznámka

Soubory INI v seznamu úloh neobsahují žádné příkazy CHANDATA.

Ukládání inicializačních programů (COMPLETE, INITIAL)

Soubory z pracovní paměti se mohou ukládat také na externí PC, odkud je možné je opět načíst.

- Soubory se ukládají pomocí příkazu COMPLETE.
- Pomocí příkazu INITIAL se vytvoří soubor INI ((_N_INITIAL_INI) zahrnující všechny oblasti.

Načtení inicializačního programu

UPOZORNĚNÍ
Ztráta dat Pokud je načten soubor s názvem "INITIAL_INI", potom jsou všechna data, do nichž tento soubor nedosazuje žádné hodnoty, inicializována standardními hodnotami. Výjimku z tohoto pravidla představují pouze strojní parametry. Standardní hodnoty (za normálních okolností "NULA" jsou dosazovány také do **nastavovaných parametrů, parametrů nástrojů, posunutí počátku, hodnot GUD, ...**.

Pro načítání jednotlivých strojních parametrů se hodí např. soubor COMPLETE_TEA_INI. V tomto souboru řídící systém očekává strojní parametry. V tomto případě díky tomu zůstávají ostatní datové oblasti nedotčeny.

Načítání inicializačního programu

Pokud programy INI využívají pouze data určitého kanálu, mohou být zvoleny a vyvolány stejně jako výrobní programy. Takto je tedy možné inicializovat také data řízená programem.

4.5 Manipulace se soubory

4.5.1 Zápis do souboru (WRITE)

Pomocí příkazu `WRITE` je možné zapisovat bloky/data z NC programu na konec souboru, který se nachází v pasivním systému souborů nebo v externí paměti programů (protokolový soubor). Může se jednat také o program, který je momentálně zpracováván.

Poznámka

Jestliže soubor, do něhož se má pomocí příkazu `WRITE` zapisovat, ještě v paměti programů neexistuje, bude nově založen.

Předpoklady

Momentálně nastavená úroveň ochrany musí být stejná nebo vyšší, než je úroveň oprávnění pro zápis do souboru pomocí příkazu WRITE. Pokud tomu tak není, přístup je odmítnut a vypíše se chybové hlášení (výsledná hodnota v chybové proměnné = 13).

Syntaxe

```
DEF INT <chyba>
...
WRITE(<Chyba>,"<Název souboru>"/"<ExtG>","<Blok/Data>")
```

Význam

WRITE:	Příkaz pro vložení bloku, příp. dat na konec uvedeného souboru		
<chyba>:	**Parametr 1:** Proměnná pro výslednou chybovou hodnotu		
	Typ:	INT	
	Hodnota:	0	žádná chyba
		1	Cesta není povolena
		2	Cesta není nalezena
		3	Soubor není nalezen
		4	nesprávný datový typ
		10	Soubor je plný
		11	Soubor je využíván
		12	žádné volné kapacity
		13	žádná přístupová oprávnění
		14	Chybějící příkaz EXTOPEN pro výstupní zařízení nebo zpracování tohoto příkazu skončilo neúspěchem
		15	Chyba při zápisu na externí zařízení
		16	naprogramována neplatná externí cesta
<název souboru>:	**Parametr 2:** Název souboru, do kterého se má uvedený blok, příp. uvedená data, vložit.		
	Typ:	STRING	
	Před vlastním názvem souboru může být uvedena absolutní cesta. Pokud cesta není udána, bude soubor vyhledán v aktuálním adresáři (= adresář zvoleného programu). Pokud budete potřebovat informace o pravidlech pro zadávání cesty, viz "Adresování souborů v paměti pro programy (Strana 559)".		
<ExtG>:	Jestliže mají být data odesílána na externí zařízení/do externího souboru pomocí funkce „Process DataShare“, musí být místo názvu souboru uveden symbolický identifikátor pro externí zařízení/soubor, které se mají otevřít.		
	Typ:	STRING	
	Pokud budete potřebovat další informace, viz "Process DataShare - Výstup do externího zařízení/souboru (EXTOPEN, WRITE, EXTCLOSE) (Strana 1010)". **Upozornění:** Tento identifikátor musí být identický s identifikátorem použitým v příkazu EXTOPEN.		
<blok/data>:	**Parametr 3:** Blok, příp. data, která mají být do uvedeného souboru vložena.		
	Typ:	STRING	

Poznámka

Při zápisu do pasivního systému souborů nebo do externí paměti programů implicitně vkládá příkaz WRITE na konec výstupního řetězce implicitně znak "LF" (LINE-FEED = konec řádku).

Při odesílání do externího zařízení/souboru prostřednictvím funkce „Process DataShare“ toto chování neplatí. Jestliže má být znak "LF" odeslán, musí být explicitně uveden ve výstupním řetězci.

--> K tomu viz příklad 3: Implicitní/explicitní znak "LF"!

Okrajové podmínky

- **Maximální velikost souboru (--> výrobce stroje!)**
  Maximální možná velikost protokolového souboru v pasivním systému souborů se nastavuje pomocí strojního parametru:
  MD11420 $MN_LEN_PROTOCOL_FILE
  Maximální velikost souboru platí pro všechny soubory, které byly založeny příkazem `WRITE` v pasivním systému souborů. V případě překročení této velikosti se aktivuje chybové hlášení a blok, příp. data se do paměti neuloží. Pokud postačuje kapacita paměti, může být založen nový soubor.

Příklady

Příklad 1: Příkaz WRITE do pasivního systému souborů bez absolutního udání cesty

Programový kód	Komentář
`N10 DEF INT ERROR`	`; definice chybové proměnné`
`N20 WRITE(ERROR,"PROT","PROTOKOLL VOM 7.2.97")`	`; zapíše text "PROTOKOLL VOM 7.2.97" do souboru _N_PROT_MPF`
`N30 IF ERROR`	`; vyhodnocování chyby`
`N40 MSG ("Fehler bei WRITE-Befehl:" << ERROR)`	
`N50 M0`	
`N60 ENDIF`	
`...`	

Příklad 2: Příkaz WRITE do pasivního systému souborů s absolutním udáním cesty

Programový kód
`...`
`WRITE(ERROR,"/_N_WKS_DIR/_N_PROT_WPD/_N_PROT_MPF","PROTOKOLL VOM 7.2.97")`
`...`

Příklad 3: Implicitní/explicitní znak "LF"

a, Zápis do pasivního systému souborů s implicitně vkládaným znakem "LF"

Programový kód
`...`
`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`

Výsledný výstup:

MY_STRING

MY_STRING

b, Zápis do externího souboru bez implicitně vkládaného znaku "LF"

Programový kód

```
...
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
```

Výsledný výstup:

MY_STRINGMY_STRING

c, Zápis do externího souboru s explicitně vkládaným znakem "LF"

Aby bylo dosaženo stejného výsledku jako u případu a, musí být naprogramováno následující:

Programový kód

```
...
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
```

Výsledný výstup:

MY_STRING

MY_STRING

4.5.2 Vymazání souboru (DELETE)

Pomocí příkazu `DELETE` je možné vymazat všechny soubory bez ohledu na to, zda tyto soubory vznikly příkazem `WRITE` nebo ne. Pomocí příkazu `DELEE` mohou být mazány také soubory, které byly vytvořeny s vyšší úrovní přístupových práv.

Syntaxe

```
DEF INT <chyba>
DELETE(<chyba>,"<název souboru>")
```

Význam

DELETE:	Příkaz pro vymazání zadaného souboru.		
<chyba>:	Proměnná pro výslednou chybovou hodnotu		
	Typ:	INT	
	Hodnota:	0	žádná chyba
		1	Cesta není povolena
		2	Cesta není nalezena
		3	Soubor není nalezen
		4	nesprávný datový typ
		11	Soubor je využíván
		12	žádné volné kapacity
		20	jiná chyba
<název souboru>:	Název souboru, který má být vymazán.		
	Typ:	STRING	
	Před vlastním názvem souboru může být uvedena absolutní cesta. Pokud cesta není udána, bude soubor vyhledán v aktuálním adresáři (= adresář zvoleného programu). Pokud budete potřebovat informace o pravidlech pro zadávání cesty, viz "Adresování souborů v paměti pro programy (Strana 559)".		

Příklad:

Programový kód	**Komentář**
N10 DEF INT ERROR	; definice chybové proměnné
N15 STOPRE	; zastavení předběžného zpracování
N20 DELETE(ERROR,"/_N_SPF_DIR/_N_TEST1_SPF")	; vymazání souboru TEST1 v adresáři podprogramu
N30 IF ERROR	; vyhodnocování chyby
N40 MSG("Fehler bei DELETE-Befehl:" <<ERROR)	
N50 M0	
N60 ENDIF	

4.5.3 Čtení řádků v souboru (READ)

Příkaz READ načte v zadaném souboru jeden nebo více řádků a načtené informace uloží do pole typu STRING. Každý načtený řádek obsadí v tomto poli jeden jeho prvek.

Předpoklady

Momentálně nastavená úroveň ochrany musí být stejná nebo vyšší, než je úroveň oprávnění pro čtení souboru pomocí příkazu READ. Pokud tomu tak není, přístup je odmítnut a vypíše se chybové hlášení (výsledná hodnota v chybové proměnné = 13).

Syntaxe

```
DEF INT <chyba>
DEF STRING[<délka řetězce>] <výsledek>[<n>,<m>]
READ(<chyba>,"<název souboru>",<počáteční řádek>,<počet
řádků>,<výsledek>)
```

Význam

READ:	Příkaz pro čtení řádků ze zadaného souboru a pro uložení těchto řádků do pole proměnných.		
<chyba>:	Proměnná pro výslednou chybovou hodnotu (parametr Call-By-Reference)		
	Typ:	INT	
	Hodnota:	0	žádná chyba
		1	Cesta není povolena
		2	Cesta není nalezena
		3	Soubor není nalezen
		4	nesprávný datový typ
		11	Soubor je využíván
		13	Nedostatečná přístupová oprávnění
		21	Řádek není k dispozici (parametr <počáteční řádek> nebo parametr <počet řádků> je větší než počet řádků v zadaném souboru).
		22	Délka pole v proměnné pro výsledky (<výsledek>) je příliš malá.
		23	Oblast řádků je zvolena příliš velká (parametr <počet řádků>, takže čtení probíhá až za konec souboru.
<název souboru>:	Název souboru, z něhož se má číst (parametr Call-By-Value)		
	Typ:	STRING	
	Před vlastním názvem souboru může být uvedena absolutní cesta. Pokud cesta není udána, bude soubor vyhledán v aktuálním adresáři (= adresář zvoleného programu). Pokud budete potřebovat informace o pravidlech pro zadávání cesty, viz "Adresování souborů v paměti pro programy (Strana 559)".		
<počáteční řádek>:	Počáteční řádek oblasti v souboru, z něhož se má číst (parametr Call-By-Value)		
	Typ:	INT	
	Hodnota:	0	Načte se určitý počet řádků před koncem souboru, přičemž jejich počet je zadán parametrem <počet řádků>
		1 ... n	Číslo prvního řádku, která se má načíst.
<počet řádků>:	Počet řádků, které se mají načíst (parametr Call-By-Value)		
	Typ:	INT	

<výsledek>:	Proměnná pro výsledek (parametr "Call-By-Reference") Pole proměnných, do kterého se ukládá načtený text.	
	Typ:	STRING (max. délka: 255)
	Jestliže parametr <počet řádků> specifikuje méně řádků, než kolik odpovídá velikosti pole výsledků [<n>,<m>], potom zůstanou zbývající prvky pole nezměněny. Konec řádku reprezentovaný řídícími znaky "LF" (Line Feed) nebo "CR LF" (Carriage Return Line Feed) se do pole výsledků **neukládá**. Jestliže je řádek delší, než je definovaná délka řetězce, jsou načtené řádky oříznuty. Nevypisuje se žádné chybové hlášení.	

Poznámka

Binární soubory není možné načítat. V takovém případě je nahlášena chyba "Nesprávný datový typ" (výsledná hodnota v chybové proměnné = 4). Následující datové typy není možné číst: _BIN, _EXE, _OBJ, _LIB, _BOT, _TRC, _ACC, _CYC, _NCK.

Příklad:

Programový kód	**Komentář**
N10 DEF INT ERROR	; definice chybové proměnné
N20 DEF STRING[255] RESULT[5]	; definice proměnné pro výsledky
N30 READ(ERROR,"/_N_CST_DIR/_N_TESTFILE_MPF",1,5,RESULT)	; název souboru s doménou, příponou souboru a zadání cesty
N40 IF ERROR <>0	; vyhodnocování chyby
N50 MSG("FEHLER"<<ERROR<<"BEI READ-BEFEHL")	
N60 M0	
N70 ENDIF	
...	

4.5.4 Kontrola existence souboru (ISFILE)

Pomocí příkazu ISFILE je možné zkontrolovat, zda určitý soubor existuje v paměti programů.

Syntaxe

```
<výsledek>=ISFILE("<název souboru>")
```

Význam

ISFILE:	Příkaz pro kontrolu, zda určitý soubor existuje		
<název souboru>:	Název souboru, jehož existence má být zkontrolována.		
	Typ:	STRING	
	Před vlastním názvem souboru může být uvedena absolutní cesta. Pokud cesta není udána, bude soubor vyhledán v aktuálním adresáři (= adresář zvoleného programu). Pokud budete potřebovat informace o pravidlech pro zadávání cesty, viz "Adresování souborů v paměti pro programy (Strana 559)".		
<výsledek>:	Proměnná, do které se uloží výsledek kontroly		
	Typ:	BOOL	
	Hodnota:	TRUE	Soubor existuje
		FALSE	Soubor neexistuje

Příklady

Příklad 1

Programový kód	Komentář
N10 DEF BOOL RESULT	; definice proměnné pro výsledky
N20 RESULT=ISFILE("TESTFILE")	
N30 IF(RESULT==FALSE)	
N40 MSG("SOUBOR NEEXISTUJE")	
N50 M0	
N60 ENDIF	
...	

Příklad 2

Programový kód	Komentář
N10 DEF BOOL RESULT	; definice proměnné pro výsledky
N20 RESULT=ISFILE("TESTFILE")	
N30 IF(NOT ISFILE("TESTFILE"))	
N40 MSG("SOUBOR NEEXISTUJE")	
N50 M0	
N60 ENDIF	
...	

4.5.5 Zjišťování informací o souboru (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO)

Prostřednictvím příkazů FILEDATE, FILETIME, FILESIZE, FILESTAT a FILEINFO mohou být zjišťovány určité informace o daném souboru, jako jsou datum/přesný čas, kdy došlo k poslednímu přístupu za účelem zápisu, momentální velikost souboru, stav souboru nebo shrnutí všech těchto informací.

Předpoklady

Momentálně nastavená úroveň ochrany musí být stejná nebo vyšší, než je úroveň oprávnění pro vyvolání výpisu nadřazeného adresáře. Pokud tomu tak není, přístup je odmítnut a vypíše se chybové hlášení (výsledná hodnota v chybové proměnné = 13).

Syntaxe

```
FILE....(<chyba>,"<název souboru>",<výsledek>)
```

Význam

FILEDATE:	Poskytuje **datum** posledního přístupu do souboru za účelem zápisu		
FILETIME:	Poskytuje **přesný čas** posledního přístupu do souboru za účelem zápisu		
FILESIZE:	Poskytuje údaj o **momentální velikosti** souboru		
FILESTAT:	Poskytuje pro **status** následující **oprávnění**, která se týkají určitého souboru. • Čtení (r: read) • Zápis (w: write) • Spustit (x: execute) • Zobrazit (s: show) • Vymazat (d: delete) **Poznámka:** Tyto úrovně ochrany jsou speciálními vlastnostmi pasivního systému souborů. Při přístupu do externí paměti pro programy proto příkaz FILESTAT vrací pouze předdefinovaná přístupová oprávnění (77777).		
FILEINFO:	Pro daný soubor poskytuje **součet všech informací**, které je možné načíst pomocí funkcí FILEDATE, FILETIME, FILESIZE a FILESTAT.		
<chyba>:	Proměnná pro výslednou chybovou hodnotu (parametr Call-By-Reference)		
	Typ:	VAR INT	
	Hodnota:	0	žádná chyba
		1	Cesta není povolena
		2	Cesta není nalezena
		3	Soubor není nalezen
		4	nesprávný datový typ
		13	Nedostatečná přístupová oprávnění
		22	Délka řetězce v proměnné pro výsledky (<výsledek>) je příliš malá.
<název souboru>:	Název souboru, o němž mají být zjištěny požadované informace		
	Typ:	CHAR[160]	
	Před vlastním názvem souboru může být uvedena absolutní cesta. Pokud cesta není udána, bude soubor vyhledán v aktuálním adresáři (= adresář zvoleného programu). Pokud budete potřebovat informace o pravidlech pro zadávání cesty, viz "Adresování souborů v paměti pro programy (Strana 559)".		

<výsledek>:	Proměnná pro výsledek (Call-By-Reference-parametr) Proměnná, do které se ukládají požadované informace o souboru.			
	Typ:	VAR CHAR[8]	kde:	FILEDATE Formát: „dd.mm.rr"
		VAR CHAR[8]	kde:	FILETIME Formát: "hh:mm.ss"
		VAR INT	kde:	FILESIZE Výpis velikosti souboru v bytech.
		VAR CHAR[5]	kde:	FILESTAT Formát: "rwxsd" (r: **r**ead, w: **w**rite, x: e**x**ecute, s: **s**how, d: **d**elete)
		VAR CHAR[32]	kde:	FILEINFO Formát: "rwxsd nnnnnnnn dd.mm.yy hh:mm:ss"

Příklad:

Programový kód	**Komentář**
N10 DEF INT ERROR	; definice chybové proměnné
N20 STRING[32] RESULT	; definice proměnné pro výsledky
N30 FILEINFO(ERROR,"/_N_MPF_DIR/_N_TESTFILE_MPF",RESULT)	; název souboru s doménou, příponou souboru a udáním cesty
N40 IF ERROR <> 0	; vyhodnocování chyby
N50 MSG("FEHLER"<<ERROR<<"u příkazu FILEINFO")	
N60 M0	
N70 ENDIF	
...	

Program v příkladu by mohl do proměnné pro výsledek RESULT uložit například následující řetězec:

"77777 12345678 26.05.00 13:51:30"

4.6 Chráněné oblasti

4.6.1 Definice chráněných oblastí (CPROTDEF, NPROTDEF)

Chráněné oblasti, které mají chránit součásti stroje proti kolizím, jsou ve výrobním programu definovány v určitých blocích. Ty se skládají z následujících prvků:

1. Definice pracovní roviny
   Před vlastní definicí chráněné oblasti musí být zvolena pracovní rovina, na kterou se má popis kontury chráněné oblasti vztahovat.
2. Začátek definice
   V závislosti na příkazu NC jazyka bude založena chráněná oblast vztahující se ke kanálu nebo ke specifickému stroji.
3. Popis kontury chráněné oblasti
   Kontura chráněné oblasti je popisována prostřednictvím pohybů pracovním posuvem. Tyto pohyby se neuskutečňují a nemají žádnou souvislost s předcházejícími nebo následujícími popisy geometrie. Definují pouze chráněnou oblast.
4. Konec definice

Syntaxe

```
DEF INT <Var>
G17/G18/G19
CPROTDEF/NPROTDEF(<n>,<t>,<AppLim>,<AppPlus>,<AppMinus>)
G0/G1/... X/Y/Z...
...
EXECUTE(<Var>)
```

Význam

`DEF INT <Var>`:	Definice lokálních pomocných proměnných datového typu INTEGER	
`<Var>`:	Název pomocných proměnných	
`G17/G18/G19`:	Pracovní rovina **Upozornění:** Pracovní rovina nesmí být před koncem definice změněna. Programování aplikáty mezi začátkem a koncem definice je nepřípustné.	
`CPROTDEF()`:	Předem definovaná procedura za účelem definice **kanálové** chráněné oblasti	
`NPROTDEF()`:	Předem definovaná procedura za účelem definice **strojní** chráněné oblasti	
`<n>`:	Číslo definované chráněné oblasti	
	Datový typ:	INT

<t>:	Typ chráněné oblasti		
	Datový typ:	BOOL	
	Hodnota:	TRUE	Chráněná oblast vztahující se k **nástroji**:
		FALSE	Chráněná oblast vztahující se k **obrobku**:
<AppLim>:	Druh ohraničení ve 3. rozměru		
	Datový typ:	INT	
	Hodnota:	0	žádné ohraničení
		1	Ohraničení v kladném směru
		2	Ohraničení v záporném směru
		3	Ohraničení v kladném a záporném směru
<AppPlus>:	Hodnota ohraničení v kladném směru ve 3. rozměru		
	Datový typ:	REAL	
<AppMinus>:	Hodnota ohraničení v záporném směru ve 3. rozměru		
	Datový typ:	REAL	
G0/G1/... X/Y/Z... ...:	Kontura chráněné oblasti je popsána maximálně 11 posuvovými pohyby ve zvolené pracovní rovině. První posuvový pohyb je přitom pohyb na konturu. Poslední bod popisu kontury se vždy musí shodovat s pevním bodem popisu kontury. Jako chráněná platí oblast vlevo od kontury: • Vnitřní chráněná oblast Konturu pro vnitřní chráněnou oblast je potřeba popisovat proti směru hodinových ručiček. • Vnější chráněná oblast (přípustná pouze pro chráněné oblasti vztahující se k obrobku!) Konturu pro vnější chráněnou oblast je potřeba popisovat ve směru hodinových ručiček. Přípustné jsou následující konturové prvky: • příkazy G0, G1 pro přímé konturové prvky • G2 pro kruhové oblouky ve směru hodinových ručiček Přípustné pouze pro chráněné oblasti vztahující se k obrobku! Nepřípustné u chráněných oblastí vztahujících se k nástroji, protože ty smí být jen konvexní. • G3 pro kruhové oblouky proti směru hodinových ručiček **Upozornění:** Chráněná oblast nemůže být popsána prostřednictvím celé kružnice. Celá kružnice musí být rozdělena na dva půlkruhy. **Upozornění:** Posloupnost G2 → G3 příp. G3 → G2 je nepřípustná. Mezi oba kruhové bloky musí být vložen krátký blok s příkazem G1.		
EXECUTE(<Var>):	Předem definovaná procedura, která označuje konec definice Pomocí příkazu EXECUTE se vrátíte zpět k normálnímu zpracovávání programu.		

Příklad

Viz příklad „Aktivování/deaktivování chráněné oblasti (CPROT, NPROT) (Strana 583)“.

Další informace

Chráněné oblasti vztahující se ke specifickému stroji

Chráněná oblast vztahující se ke konkrétnímu stroji, příp. její kontura, je definována prostřednictvím geometrických os, tzn. je vztažena k základnímu souřadnému systému (BCS) kanálu. Aby se mohlo uskutečňovat správné monitorování chráněných oblastí ve všech kanálech, ve kterých je příslušná strojní chráněná oblast aktivní, musí být základní souřadný systém (NCS) ve všech kanálech, jichž se to týká, identický:

- Poloha počátku souřadného systému vztažená na počátek souřadného systému stroje
- Orientace souřadných os

Vztažný bod popisu kontury

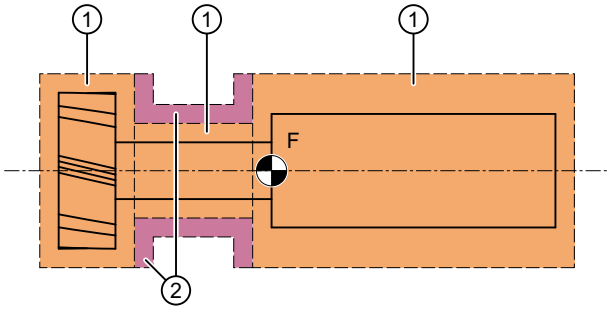
- Chráněné oblasti vztahující se k nástroji
  Souřadnice pro chráněné oblasti vztahující se k **nástroji** je potřeba zadávat ve vztahu ke **vztažnému bodu držáku nástroje F**.
- Chráněné oblasti vztahující se k obrobku
  Souřadnice pro chráněné oblasti vztahující se k **obrobku** je potřeba zadávat ve vztahu k počátku **základního souřadného systému (BCS)**.

Rotačně symetrické chráněné oblasti

U rotačně symetrických chráněných oblastí (např. vřetenové sklíčidlo) musí být popsána celá kontura, nejen kontura po osu otáčení.

Chráněné oblasti vztahující se k nástroji

Chráněné oblasti vztahující se k nástroji musí být vždy konvexní. Jestliže je požadována konkávní chráněná oblast, je potřeba ji rozložit na několik konvexních chráněných oblastí.



① Konvexní chráněné oblasti
② Konkávní chráněné oblasti (**nepřípustné!**)
F Vztažný bod držáku nástroje

Okrajové podmínky

V průběhu definice chráněné oblasti nesmí být aktivní, příp. nesmí být použity, následující funkce:

- Korekce rádiusu nástroje (korekce rádiusu frézy, korekce rádiusu břitu)
- Transformace
- Najíždění na referenční bod (`G74`)

- Najíždění na pevný bod (G75)
- Doba prodlevy (G4)
- Zastavení předběžného zpracování (STOPRE)
- Konec programu (M17, M30)
- M-funkce: M0, M1, M2

4.6.2 Aktivování/deaktivování chráněné oblasti (CPROT, NPROT)

Chráněné oblasti, které byly předtím definovány ve výrobním programu, mohou být kdykoli aktivovány, příp. mohou být předběžně aktivovány pro pozdější aktivování prostřednictvím uživatelského programu PLC. Aktivní chráněné oblasti mohou být kdykoli opět deaktivovány.

Při aktivování, příp. při předběžném aktivování, existuje kromě toho možnost relativního posunutí vztažného bodu chráněné oblasti.

Poznámka

Chráněná oblast je zohledňována teprve po najetí na referenční bod všemi geometrickými osami kanálu, ve kterém byla aktivována.

Poznámka

Monitorování chráněných oblastí

Jestliže není aktivní žádná chráněná oblast vztahující se k nástroji, kontroluje se dráha nástroje vzhledem ke chráněné oblasti vztahující se k obrobku.

Jestliže není aktivní žádná chráněná oblast vztahující se k obrobku, žádné monitorování chráněných oblastí se neuskutečňuje.

Syntaxe

```
CPROT(<n>,<Status>,<XMov>,<YMov>,<ZMov>)
NPROT(<n>,<Status>,<XMov>,<YMov>,<ZMov>)
```

Význam

CPROT:	Předem definovaná procedura za účelem aktivování **kanálové** chráněné oblasti	
NPROT:	Předem definovaná procedura za účelem aktivování **strojní** chráněné oblasti	
<n>:	Číslo chráněné oblasti	
	Datový typ:	INT

<Status>:	Pomocí tohoto parametru se definuje stav aktivování ve specifickém kanálu		
	Datový typ:	INT	
	Hodnota:	0	Deaktivovat chráněnou oblast
		1	Chráněnou oblast předběžně aktivovat
		2	Aktivovat chráněnou oblast
		3	Předběžné aktivování chráněné oblasti s podmíněným zastavením
<XMov>,<YMov>,<ZMov>:	Aditivní hodnoty posunutí ve směru X/Y/Z Posunutí může být provedeno v 1, 2 nebo ve 3 směrech. Hodnoty posunutí jsou vztaženy na: • počátek souřadného systému stroje v případě chráněné oblasti vztahující se k **obrobku** • vztažný bod držáku nástroje F v případě chráněné oblasti vztažené k **nástroji**		
	Datový typ:	REAL	

Příklad:

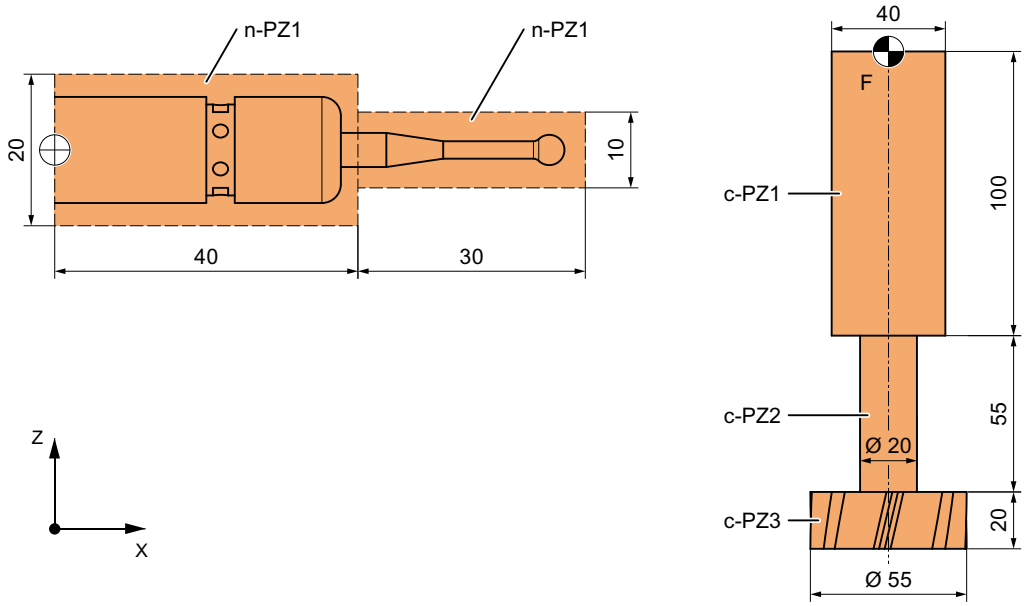
V případě frézky mají být monitorovány možné kolize frézy s měřicí sondou. Délka měřicí sondy má být při svém aktivování zadána prostřednictvím posunutí.

Pro tento účel jsou definovány následující chráněné oblasti:

- Specifická chráněná oblast daného stroje vztažená na obrobek pro držák měřicí sondy (n-PZ1) a pro měřicí sondu samotnou (n-PZ2).
- Chráněná oblast daného kanálu vztažená k nástroji pro držák frézy (c-PZ1), stopku frézy (c-PZ2) a frézu samotnou (c-PZ3).

Potvrzení všech chráněných oblastí je ve směru osy Z.

Souřadnice polohy vztažného bodu měřicí sondy při aktivování mají být X = -120, Y = 60 a Z = 80.



① Vztažný bod pro chráněnou oblast měřicí sondy

F Vztažný bod držáku nástroje

Programový kód	**Komentář**
`DEF INT PROTZONE`	`; definice pomocných proměnných`
`G17`	`; pracovní rovina XY`
`; definice chráněné oblasti:`	
`NPROTDEF(1,FALSE,3,10,-10)`	`; chráněná oblast n-PZ1`
`G01 X0 Y-10`	
`X40`	
`Y10`	
`X0`	
`Y-10`	
`EXECUTE(PROTZONE)`	
`NPROTDEF(2,FALSE,3,5,-5)`	`; chráněná oblast n-PZ2`
`G01 X40 Y-5`	
`X70`	
`Y5`	
`X40`	
`Y-5`	
`EXECUTE(PROTZONE)`	
`CPROTDEF(1,TRUE,3,0,-100)`	`; chráněná oblast c-PZ1`
`G01 X-20 Y-20`	
`X20`	
`Y20`	
`X-20`	
`Y-20`	
`EXECUTE(PROTZONE)`	

Programový kód	**Komentář**
CPROTDEF(2,TRUE,3,-100,-150)	; chráněná oblast c-PZ2
G01 X0 Y-10	
G03 X0 Y10 J10	
X0 Y-10 J-10	
EXECUTE(PROTZONE)	
CPROTDEF(3,TRUE,3,-150,-170)	; chráněná oblast c-PZ3
G01 X0 Y-27.5	
G03 X0 Y27.5 J27.5	
X0 Y27.5 J-27.5	
EXECUTE(PROTZONE)	
; aktivování chráněných oblastí:	
NPROT(1,2,-120,60,80)	; aktivování chráněné oblasti n-PZ1 s posunutím
NPROT(2,2,-120,60,80)	; aktivování chráněné oblasti n-PZ2 s posunutím
CPROT(1,2,0,0,0)	; aktivování chráněné oblasti c-PZ1
CPROT(2,2,0,0,0)	; aktivování chráněné oblasti c-PZ2
CPROT(3,2,0,0,0)	; aktivování chráněné oblasti c-PZ3

Další informace

Stav aktivování po rozběhu řídícího systému

Chráněná oblast může být aktivní už po náběhu řídícího systému a najetí osami na referenční bod. Jestliže se jedná o tento případ, jsou pro chráněno oblast nastaveny následující systémové proměnné na hodnotu TRUE:

- $SN_PA_ACTIV_IMMED[<n>] (pro chráněnou oblast vztahující se ke stroji), příp.
- $SC_PA_ACTIV_IMMED[<n>] (pro chráněnou oblast vztahující se ke kanálu)
  Index „<n>" odpovídá číslu chráněné oblastí: 0 = 1. chráněná oblast

Chráněná oblast je aktivována se stavem = 2 a bez posunutí.

Vícenásobné aktivování jedné chráněné oblasti

Chráněná oblast vztažená ke stroji může být v platnosti i ve více kanálech současně (např. chráněná oblast pinoly v případě dvou proti sobě umístěných saní). Monitorování chráněných oblasti se uskutečňuje jen tehdy, pokud bylo všemi geometrickými osami najeto na referenční bod.

Chráněná oblast nemůže být v jednom kanálu aktivována současně s různými hodnotami posunutí.

Monitorování chráněné oblasti v případě aktivní korekce rádiusu nástroje

Pokud je korekce rádiusu nástroje aktivní, je funkční monitorování chráněných oblastí možné jen tehdy, pokud je rovina korekce rádiusu nástroje identická s rovinou, ve které je definována chráněná oblast.

4.6.3 Kontrola narušení chráněné oblasti, ohraničení pracovního pole a softwarových koncových spínačů (CALCPOSI)

Funkce

Funkce CALCPOSI kontroluje v souřadném systému obrobku (WCS), zda když se vychází z počáteční polohy, mohou **geometrické osy** urazit předem zadanou dráhu, aniž by došlo k narušení aktivních ohraničení. V případě, že dráhu uvažovaného posuvu není možné z důvodu ohraničení realizovat úplně celou, poskytne funkce na svém výstupu kladné decimálně kódované stavové slovo a maximální možnou dráhu posuvu.

Definice

```
INT CALCPOSI(VAR REAL[3] <Start>, VAR REAL[3] <Dist>, VAR REAL[5]
<Limit>, VAR REAL[3] <MaxDist>, BOOL <MeasSys>, INT <TestLim>)
```

Syntaxe

```
<Status> = CALCPOSI(VAR <Start>, VAR <Dist>, VAR <Limit>, VAR
<MaxDist>, <MeasSys>, <TestLim>)
```

Význam

`CALCPOSI(...)`:	Předem definovaná funkce pro zkoušku narušení ohraničení se zřetelem na geometrické osy	
	Zastavení předběžného zpracování:	ne
	Samostatný blok:	ano

<Status>: (část 1)	Výsledná hodnota funkce. Záporná hodnota signalizuje chybový stav.		
	Datový typ:		INT
	Rozsah hodnot:		-8 ≤ x ≤ 100000
	Hodnota:	0	Dráhu posuvu je možné realizovat úplně celou
		-1	V proměnné <Limit> je nejméně jedna složka záporná.
		-2	Chyba při výpočtu transformace. Příklad: Dráha posuvu prochází singularitou, takže polohy os nejsou definovány.
		-3	Uváděná dráha posuvu <Dist> a maximální možná dráha posuvu <MaxDist> jsou lineárně závislé. **Upozornění** Může se vyskytnout pouze ve spojení s nastavením parametru <TestLim>, bit 4 == 1.
		-4	Projekce směru pohybu obsažená v parametru <Dist> na ohraničující plochu představuje nulový vektor, příp. směr pohybu je kolmý na narušovanou ohraničující plochu. **Upozornění** Může se vyskytnout pouze ve spojení s nastavením parametru <TestLim>, bit 5 == 1.
		-5	V parametru <TestLim> jsou bit 4 == 1 A bit 5 == 1
		-6	S minimálně jednou strojní osou, která musí být pro přezkoumání ohraničení pohybů uvažována, nebylo najeto na referenční bod.
		-7	Funkce prevence kolize: Neplatná definice kinematického řetězce nebo chráněné oblasti.
		-8	Funkce prevence kolize: Funkce nemohla být zpracována kvůli chybě paměti.
<Status>: (část 2)	Místo jednotek **Upozornění** Pokud je narušeno více ohraničení současně, je uvedeno to, které způsobuje nejsilnější omezení předem zadané dráhy posuvu.		
	Hodnota:	1	Dráha je omezena softwarovým koncovým spínačem
		2	Dráha je omezena ohraničením pracovního pole
		3	Dráha je omezena chráněnými oblastmi
		4	Funkce prevence kolize: Dráha je omezena chráněnými oblastmi
	Místo desítek		
	Hodnota:	1x	Počáteční hodnota narušuje hranici
		2x	Předem zadaná přímka narušuje hranici. Tato hodnota je výsledkem také tehdy, pokud samotný koncový bod žádnou hranici nenarušuje, ale na dráze z počátečního do koncového bodu k narušení některé z mezních hodnot došlo (např. průjezd chráněnou oblastí, zakřivené polohy softwarových koncových spínačů ve WCS kvůli nelineárním transformacím, např. Transmit).

<Status>: (část 3)	Místo stovek		
	Hodnota:	1xx	A místo jednotek == 1 nebo 2: Je narušena kladná mezní hodnota.
			A místo jednotek == 3 [1]: Je narušena chráněná oblast specifikovaná v NC systému.
		2xx	A místo jednotek == 1 nebo 2: Je narušena záporná mezní hodnota.
			A místo jednotek == 3 [1]: Je narušena chráněná oblast specifického kanálu.
<Status>: (část 4)	Místo tisíců		
	Hodnota:	1xxx	A místo jednotek == 1 nebo 2: Faktor, kterým se násobí číslo osy, která narušuje ohraničení. Počítání os začíná 1. Přiřazení: • Softwarový koncový spínač: Strojní osy • Ohraničení pracovního pole: Geometrické osy A místo jednotek == 3 [1]: Faktor, se kterým se násobí číslo narušované chráněné oblasti.
<Status>: (část 5)	Místo stovek tisíců		
	Hodnota:	0xxxxx	Místo stovek tisíců == 0: Parametr <Dist> zůstává nezměněn.
		1xxxxx	V parametru <Dist> funkce ukládá směrový vektor, který definuje další směr pohybu na ohraničující ploše. Mohou se vyskytnout následující okrajové podmínky: • Je narušen softwarový koncový spínač nebo ohraničení pracovního pole (nikoli v počátečním bodě) • Transformace **není** aktivní • <TestID>, bit 4 nebo bit 5 == 1
<Start>:	Referenční vztah na vektor s počátečními pozicemi: • <Start> [0]: 1. geometrická osa • <Start> [1]: 2. geometrická osa • <Start> [2]: 3. geometrická osa		
	Typ parametru:		Vstup
	Datový typ:		VAR REAL[3]
	Rozsah hodnot:		-max. hodnota REAL ≤ x[n] ≤ +max. hodnota REAL

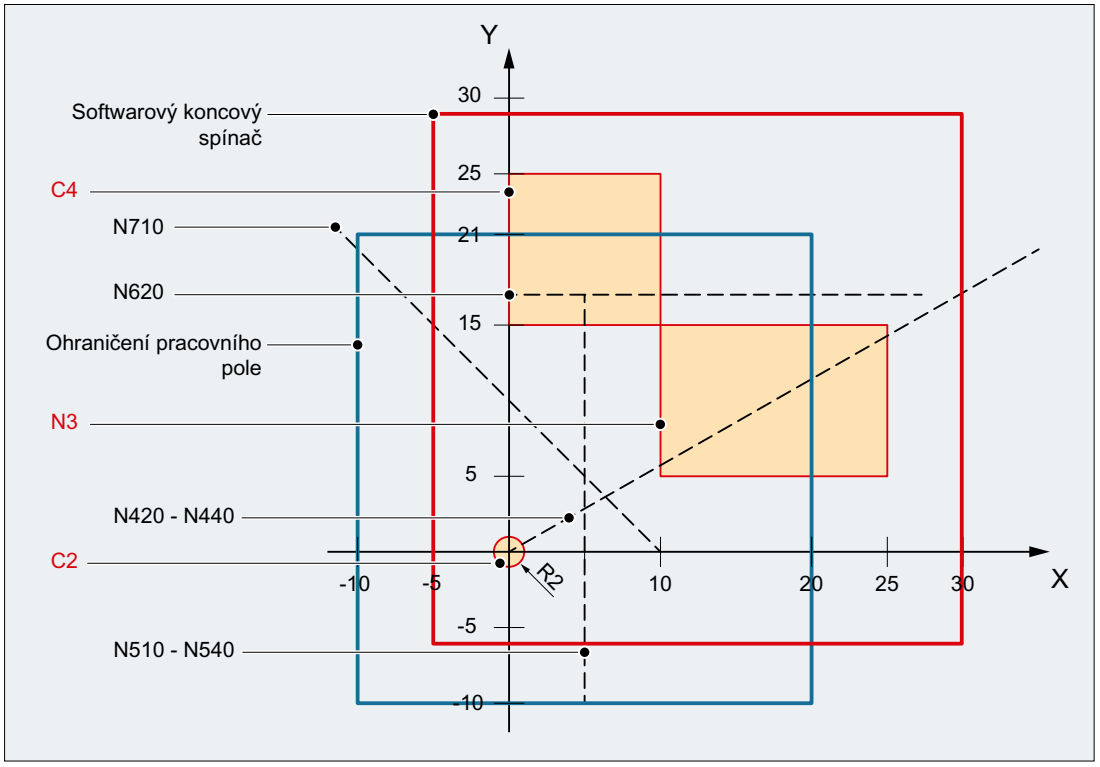
<Dist>:	Referenční vztah na vektor.	
	Vstup: inkrementální dráha posuvu • <Dist> [0]: 1. geometrická osa • <Dist> [1]: 2. geometrická osa • <Dist> [2]: 3. geometrická osa	
	Výstup (jen pokud je aktivováno místo stovek tisíců v proměnné <Status>):	
	Parametr <Dist> obsahuje jako výstupní hodnotu jednotkový vektor **v**, který definuje další směr posuvu ve WCS.	
	Případ 1: Stanovení vektoru **v** v případě, že <TestID>, bit 4 == 1 Vstupní vektory <Dist> a <MaxDist> definují rovinu pohybu. Tato rovina protíná narušovanou ohraničující plochu. Průsečnice obou rovin definuje směr vektoru **v**. Orientace (znaménko) je přitom zvolena tak, aby úhel mezi vstupním vektorem <MaxDist> a vektorem **v** nebyl větší než 90 stupňů.	
	Případ 2: Stanovení vektoru **v** v případě, že <TestID>, bit 5 == 1 Vektor **v** je jednotkový vektor ve směru projekce vektoru posuvu obsaženého v parametru <Dist> do ohraničující plochy. Jestliže je projekce vektoru posuvu do ohraničující plochy nulový vektor, výpočet skončí s chybou.	
	Typ parametru:	Vstup/výstup
	Datový typ:	VAR REAL[3]
	Rozsah hodnot:	-max. hodnota REAL ≤ x[n] ≤ +max. hodnota REAL
<Limit>:	Odkaz na pole délky 5. • <Limit> [0 - 2]: Minimální vzdálenosti geometrických os od ohraničení: – <Limit> [0]: 1. geometrická osa – <Limit> [1]: 2. geometrická osa – <Limit> [2]: 3. geometrická osa Minimální vzdálenosti jsou dodrženy v případech: – Ohraničení pracovního pole: Žádná omezení – Softwarový koncový spínač: Pokud buď není aktivní žádná transformace nebo je aktivní jedna transformace, u které je ale možné jednoznačné přiřazení geometrických os lineárním strojním osám, např. u 5-osé transformace. • <Limit> [3]: Obsahuje minimální vzdálenost pro lineární strojní osy, které nemohou být např. na základě nelineární transformace přiřazeny žádné geometrické ose. Tato hodnota se kromě toho používá jako mezní hodnota při kontrole tradičních chráněných oblastí a oblastí chráněných kvůli prevenci kolize. • <Limit> [4]: Obsahuje minimální vzdálenost pro kruhové osy stroje, které např. na základě nelineární transformace nemohou být přiřazeny žádné geometrické ose. **Upozornění** Tato hodnota se uplatňuje pouze při monitorování softwarových koncových spínačů při speciálních transformacích.	
	Typ parametru:	Vstup
	Datový typ:	VAR REAL[5]
	Rozsah hodnot:	-max. hodnota REAL ≤ x[n] ≤ +max. hodnota REAL

<MaxDist>:	Odkaz na vektor s inkrementální dráhou posuvu, u kterého je překročena předem zadaná minimální vzdálenost od ohraničení osy pro všechny strojní osy podílející se na pohybu: • <Dist> [0]: 1. geometrická osa • <Dist> [1]: 2. geometrická osa • <Dist> [2]: 3. geometrická osa Pokud dráha není nijak omezena, je obsah tohoto výsledného parametru roven obsahu parametru <Dist>.		
	V případě, že <TestID>, bit 4 == 1: <Dist> a <MaxDist> <MaxDist> a <Dist> musí jako vstupní hodnoty obsahovat vektory, které popisují rovinu, ve které se uskutečňuje pohyb. Oba vektory musí být vůči sobě lineárně nezávislé. Absolutní hodnota <MaxDist> je libovolná. Pokud budete potřebovat informace o výpočtu směru pohybu, viz popis parametru <Dist>.		
	Typ parametru:		Výstup
	Datový typ:		VAR REAL[3]
	Rozsah hodnot:		-max. hodnota REAL ≤ x[n] ≤ +max. hodnota REAL
<MeasSys>:	Systém měřicích jednotek (palce/metrický systém) pro údaje poloh a délek (**nepovinný**)		
	Datový typ:		BOOL
	Hodnota:	FALSE (předdefinované nastavení)	Systém měřicích jednotek v souladu s momentálně aktivním příkazem G-funkce ze skupiny 13 (G70, G71, G700, G710). **Upozornění** Když je aktivní příkaz G70 a základní systém je metrický nebo když je aktivní příkaz G71 a základní systém je imperiální (palce), jsou systémové proměnné $AA_IW a $AA_MW udávány v základním systému a v případě nutnosti musí být při použití funkce CALCPOSI přepočítány.
		TRUE	Systém měřicích jednotek v souladu s nastaveným základním systémem: MD52806 $MN_ISO_SCALING_SYSTEM

<TestLim>:	Bitový kód pro volbu ohraničení, které má být kontrolováno (**nepovinný**)		
	Datový typ:		INT
	Předdefinovaná hodnota:		Bit 0, 1, 2, 3, 6, 7 == 1 (207)
	Bit	**decimální**	**Význam**
	0	1	Softwarový koncový spínač
	1	2	Ohraničení pracovního pole
	2	4	Aktivované tradiční chráněné oblasti
	3	8	Předběžně aktivované tradiční chráněné oblasti
	4	16	V případě narušení softwarových koncových spínačů, příp. ohraničení pracovního pole, je na výstupu v parametru <Dist> směr pohybu v souladu s **případem 1** (viz výše).
	5	32	V případě narušení softwarových koncových spínačů, příp. ohraničení pracovního pole, je na výstupu v parametru <Dist> směr pohybu v souladu s **případem 2** (viz výše).
	6	64	Aktivované oblasti chráněné kvůli prevenci kolize
	7	128	Předběžně aktivované oblasti chráněné kvůli prevenci kolize
	8	256	Dvojice aktivovaných a předběžně aktivovaných oblastí chráněných kvůli prevenci kolize
[1)] Jestliže je narušeno několik chráněných oblastí, je na výstupu chráněná oblast, která způsobuje nejsilnější omezení předem zadané dráhy.			

Příklad:

Ohraničení



Příklad ukazuje platné softwarové koncové spínače a ohraničení pracovního pole v rovině X-Y a následující tři chráněné oblasti:

- C2: Kanálová chráněná oblast vztahující se k nástroji, aktivní, kruhová, rádius = 2 mm
- C4: Kanálová chráněná oblast vztahující se k obrobku, předběžně aktivovaná, čtvercová, délka strany = 10 mm
- N3: Chráněná oblast specifického stroje, aktivní, pravoúhlá, délky stran = 10 mm x 15 mm

NC program

V NC programu jsou napřed definovány chráněné oblasti a ohraničení pracovního pole. Potom je vyvolána funkce `CALCPOSI()` s různými nastaveními parametrů.

Programový kód

```
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
```

Programový kód

```
; chráněná oblast C2 vztahující se k nástroji
N70 CPROTDEF(2, TRUE, 0)
N80 G17 G1 X-2 Y0
N90 G3 I2 X2
N100 I-2 X-2
N110 EXECUTE(_PA)
; chráněná oblast C4 vztahující se k obrobku
N120 CPROTDEF(4, FALSE, 0)
N130 G17 G1 X0 Y15
N140 X10
N150 Y25
N160 X0
N170 Y15
N180 EXECUTE(_PA)
; chráněná oblast N3 vztahující se ke specifickému stroji
N190 NPROTDEF(3, FALSE, 0)
N200 G17 G1 X10 Y5
N210 X25
N220 Y15
N230 X10
N240 Y5
N250 EXECUTE(_PA)
; aktivování, příp. předběžné aktivování chráněných oblastí
N260 CPROT(2, 2, 0, 0, 0)
N270 CPROT(4, 1, 0, 0, 0)
N280 NPROT(3, 2, 0, 0, 0)
; definice ohraničení pracovního pole
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.
```

Programový kód

```
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)
; framy z řádků N690 a N700 opět vymazat
N720 TRANS
N730 _START[0] = 0.
N740 _START[1] = 10.
N750 _START[2] = 0.
; vektory _DIST a _MAXDIST definují rovinu pohybu
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
```

Výsledky funkce CALCPOSI()

N...	<Status>	<MaxDist>[0] ≙ X	<MaxDist>[1] ≙ Y	Poznámky
420	3123	8.040	4.594	N3 je narušena.
430	1122	20.000	11.429	Žádné monitorování chráněných oblastí, došlo k narušení ohraničení pracovního pole.
440	1121	30.000	17.143	Je aktivní ještě monitorování softwarových koncových spínačů.
510	4213	0.000	0.000	Počáteční bod narušuje C4
520	0000	0.000	-27.000	Předběžně aktivovaná C4 není monitorována. Zadanou dráhu je možné objet úplně celou.
540	2222	0.000	-25.000	Kvůli nastavení _LIMIT[1] = 2 je dráha omezena kvůli ohraničení pracovního pole.

N...	<Status>	<MaxDist>[0] ≙ X	<MaxDist>[1] ≙ Y	Poznámky
620	4223	-13.000	0.000	Vzdálenost k C4 je kvůli C2 a parametru _LIMIT[3] celkem 4 mm. Vzdálenost C2 → N3 o hodnotě 0,1 mm nezpůsobuje omezení dráhy pohybu.
710	1221	0.000	21.213	Je aktivní frame se složkou posunu a otočení. Přípustná dráha pohybu v parametru _DIST platí v posunutém a otočeném systému WCS.
820	102121	18.000	18.000	Dochází k narušení softwarového koncového spínače osy Y. Prostřednictvím nastavení <_TESTLIM> = 17 si vyžádáte výpočet směru dalšího pohybu. Tento směr je v proměnné <_DIST> (0.707, 0.0, 0.707). Hodnota je platná, protože v proměnné <_STATUS> je aktivováno místo stovek tisíců.

Další informace

Stav osy „najeto na referenční bod"

Všemi strojními osami, které mají být příkazem `CALCPOSI()` vyhodnocovány, musí být najeto na referenční bod.

Údaje dráhy vztažené na kruh

Všechny údaje dráhy zahrnující kruhové úseky jsou **vždy** interpretovány jako hodnoty rádiusu. Na to je potřeba myslet zejména u příčných os, pro něž je aktivováno programování průměrů (`DIAMON` / `DIAM90`).

Zkrácení dráhy posuvu

Jestliže je dráha posuvu uváděné osy omezena, jsou ve výstupní hodnotě `<MaxDist>` funkce odpovídajícím způsobem zkráceny dráhy posuvu také ostatních os. Výsledný koncový bod proto i nadále leží na předem zadané dráze.

Kruhové osy

Kruhové osy jsou monitorovány jen tehdy, pokud se nejedná o osy typu modulo.

Je přípustné, aby pro jednu nebo více od podílejících se na pohybu nebyly definovány žádné softwarové koncové spínače, příp. ohraničení pracovního pole nebo chráněné oblasti.

Stav softwarových koncových spínačů a ohraničení pracovního pole

Softwarové koncové spínače a ohraničení pracovního pole jsou zohledňovány jen tehdy, pokud jsou při zpracovávání funkce `CALCPOSI()` aktivní. Jejich stav může být ovlivňován např. prostřednictvím následujícího:

- Strojní parametry: MD21020 $MC_WORKAREA_WITH_TOOL_RADIUS
- Nastavované parametry: $AC_WORKAREA_CS_...
- Signály rozhraní NC/PLC: DB31, ... DBX12.2 / 3
- Příkazy: `WALIMON` / `WALIMOF`

Softwarové koncové spínače a transformace

U funkce CALCPOSI() se může stát, že u různých kinematických transformací (např. TRANSMIT) nebude vždy možné kvůli víceznačnosti na určitých místech v oblasti pohybu jednoznačně stanovit polohy strojních os (MCS) z poloh geometrických os (WCS). Při normálních posuvech vyplývá jednoznačnost zpravidla z předcházejícího dění a z podmínky, že spojitý pohyb ve WCS musí odpovídat spojitému pohybu os stroje v MCS. Při monitorování softwarových koncových spínačů se proto v případech tohoto typu připojuje poloha stroje v okamžiku zpracování funkce CALCPOSI(), aby se víceznačnost odstranila.

Poznámka

Zastavení předběžného zpracování

Při použití funkce CALCPOSI() ve spojení s transformacemi je ale samotný uživatel odpovědný za to, že kvůli synchronizaci poloh os stroje s předběžným zpracováním bude před příkazem CALCPOSI() naprogramováno zastavení předběžného zpracování (STOPRE).

Vzdálenost chráněné oblasti a tradiční chráněné oblasti

U tradičních chráněných oblastí **není** zaručeno, že při posuvech po předem zadané dráze bude dodržena bezpečnostní vzdálenost vůči všem chráněným oblastem, která je zadána v parametru <Limit>[3]. Je zaručeno pouze to, že při prodloužení koncového bodu, který je funkcí poskytován v parametru <Dist>, o bezpečnostní vzdálenost ve směru posuvu nebude narušena žádná chráněná oblast. Přímka se ale může na své dráze dostat libovolně blízko k chráněné oblasti.

Vzdálenost od chráněné oblasti a oblasti chráněné kvůli prevenci kolize

U oblastí chráněných kvůli prevenci kolize je zaručeno, že při posuvech po předem zadané dráze bude dodržena bezpečnostní vzdálenost vůči všem chráněným oblastem, která je zadána v parametru <Limit>[3].

Bezpečnostní vzdálenost zadaná v parametru <Limit>[3] je v platnosti jen tehdy, pokud platí:

<Limit>[3] > (MD10619 $MN_COLLISION_TOLERANCE)

Pokud je nastaven bit 4 v parametru <TestLim> (výpočet dalšího směru pohybu), je směrový vektor obsažený v parametru <DIST> platný jen tehdy, pokud je nastaveno místo stovek tisíců ve výstupní hodnotě funkce (<status>). Jestliže takový směr nemůže být stanoven, buď proto, že by došlo k narušení chráněné oblasti, nebo proto, že je aktivní transformace, zůstává vstupní hodnota v parametru <DIST> nezměněna. Žádné další chybové hlášení se neaktivuje.

4.7 Speciální příkazy dráhy

4.7.1 Najíždění na kódované pozice (CAC, CIC, CDC, CACP, CACN)

Pomocí příkazů pohybů typu „najíždění na kódované pozice" je možné prostřednictvím zadání čísla pozice najíždět lineárními a kruhovými osami na pevně dané pozice os, které jsou uloženy v tabulkách daných strojními parametry.

Syntaxe

```
CAC(<n>)
CIC(<n>)
CACP(<n>)
CACN(<n>)
```

Význam

`CAC(<n>)`:	Najíždění na kódovanou pozici s číslem polohy n.
`CIC(<n>)`:	Najíždění na kódovanou pozici, která je uvedena o n míst pozic před (+n) nebo za (-n) od aktuálního čísla pozice.
`CDC(<n>)`:	Najíždění na kódovanou pozici s číslem polohy n po nejkratší dráze (jen pro kruhové osy)
`CACP(<n>)`:	Najíždění na kódovanou pozici s číslem polohy n v kladném směru dráze (jen pro kruhové osy)
`CACN(<n>)`:	Najíždění na kódovanou pozici s číslem polohy n v záporném směru dráze (jen pro kruhové osy)
`<n>`:	Číslo pozice v tabulce strojních parametrů Rozsah hodnot: 0, 1, ... (max. počet míst v tabulce - 1)

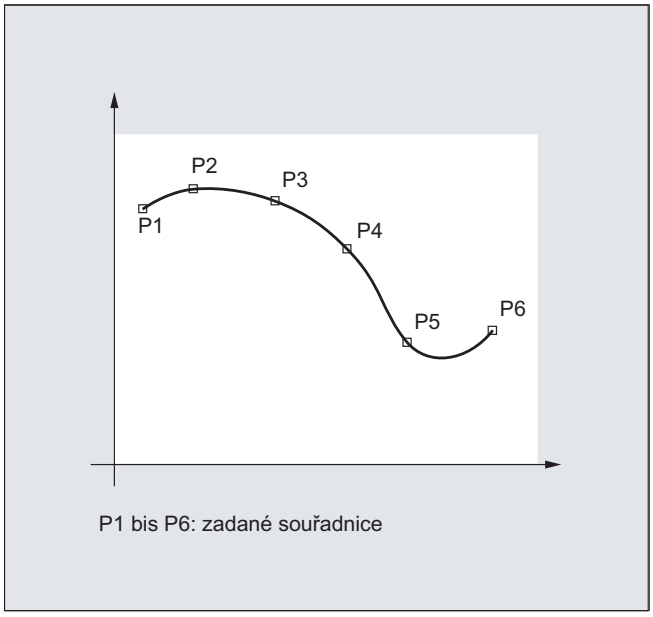
Příklad: Najíždění polohovací osou na kódované pozice

Programový kód	**Komentář**
`N10 FA[B]=300`	`; posuv pro polohovací osu B`
`N20 POS[B]=CAC(10)`	`; najíždění na kódovanou pozici s číslem polohy 10`
`N30 POS[B]=CIC(-4)`	`; najíždění na kódovanou pozici s číslem "aktuální číslo polohy" - 4`

4.7.2 Splinová interpolace (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

Libovolně zakřivené kontury na obrobku nemusí být analyticky popsány úplně přesně. Kontury tohoto druhu jsou proto aproximovány pomocí omezeného počtu uzlových bodů, např. při digitalizaci povrchů. Za účelem vytvoření digitalizovaného povrchu na obrobku musí být uzlové body spojeny, takže vznikne popsaná kontura. To umožňuje splinová interpolace.

Spline definuje křivku, která se skládá z polynomů 2. nebo 3. stupně. Vlastnosti na uzlových bodech splinu mohou být definovány **v závislosti na použitém typu splinu**.



U systému SINUMERIK solution line jsou k dispozici následující typy splinů:

- A-Spline
- B-Spline
- C-Spline

Syntaxe

Všeobecně:

```
ASPLINE X... Y... Z... A... B... C...
BSPLINE X... Y... Z... A... B... C...
CSPLINE X... Y... Z... A... B... C...
```

U B-splinů může být ještě navíc naprogramováno:

```
PW=<n>
SD=2
PL=<hodnota>
```

U A- a C-splinů může být ještě navíc naprogramováno:

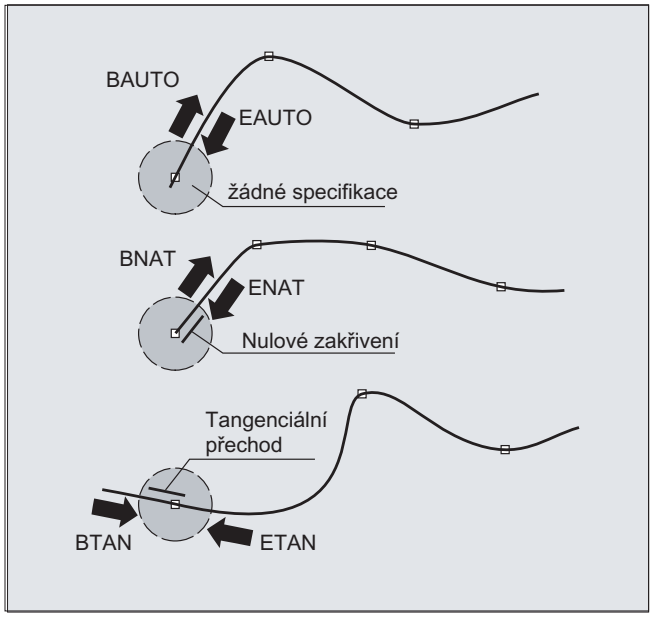
`BAUTO` / `BNAT` / `BTAN`

`EAUTO` / `ENAT` / `ETAN`

Význam

Typ splinové interpolace:	
`ASPLINE`:	Příkaz pro aktivování interpolace pomocí A-splinů
`BSPLINE`:	Příkaz pro aktivování interpolace pomocí B-splinů
`CSPLINE`:	Příkaz pro aktivování interpolace pomocí C-splinů
	Příkazy `ASPLINE`, `BSPLINE` a `CSPLINE` mají modální platnost a patří do skupiny příkazů dráhy.

Uzlové, příp. řídící body:			
`X... Y... Z...` `A... B... C...`	Polohy v kartézských souřadnicích.		
Váha bodu (jen B-spliny):			
`PW`:	Pomocí příkazu `PW` je možno pro každý uzlový bod naprogramovat tak zvanou "váhu bodu".		
`<n>`:	"Váha bodu"		
	Rozsah hodnot:	0 ≤ n ≤ 3	
	Velikost kroku:	0.0001	
	Působnost:	n > 1	Křivka je silněji přitahována k uzlovému bodu.
		n < 1	Křivka je přitahována k uzlovému bodu méně silně.
Stupeň splinu (jen B-spliny):			
`SD`:	Standardně se používají se polynomy 3. stupně. Naprogramováním příkazu `SD=2` se mohou ale používat také polynomy 2. stupně.		
vzdálenost mezi uzly (jen B-spliny):			
`PL`:	Vzdálenosti mezi uzly jsou interně vypočítávány tak, aby byly vhodné. Řídící systém ale může pracovat i se zadanými vzdálenostmi uzlů, které se udávají pomocí příkazu `PL` jako tzv. parametr interval-délka.		
`<hodnota>`:	Parametr interval-délka		
	Rozsah hodnot:	stejně jako rozměrový údaj dráhy	
Chování na přechodu na začátku splinové křivky (jen A- a C-spliny):			
`BAUTO`:	Žádný údaj pro chování na přechodu. Začátek vyplývá z polohy prvního bodu.		
`BNAT`:	Nulové zakřivení		
`BTAN`:	Tangenciální přechod k předcházejícímu bloku.		
Chování na přechodu na konci splinové křivky (jen A- a C-spliny):			
`EAUTO`:	Žádný údaj pro chování na přechodu. Konec vyplývá z polohy posledního bodu.		
`ENAT`:	Nulové zakřivení		

ETAN:	Tangenciální přechod k předcházejícímu bloku.
	

Poznámka

Naprogramované chování na přechodu nemá na B-spliny žádný vliv. B-spline je v počátečním a koncovém bodě vždy napojen tangenciálně k řídícímu mnohoúhelníku.

Okrajové podmínky

- Korekce rádiusu nástroje může být uplatněna.
- Monitorování kolize se uskutečňuje v projekci v rovině.

Příklady

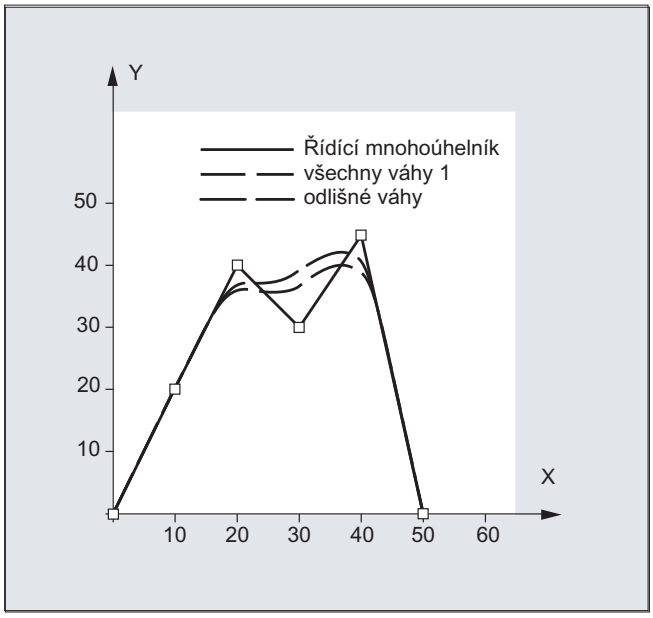
Příklad 1: B-Spline

```
Programový kód 1 (všechny váhy 1)
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
N60 X40 Y45
N70 X50 Y0
```

Programový kód 2 (různé váhy)

```
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
```

Programová kód 3 (řídící mnohoúhelník)	**Komentář**
N10 G1 X0 Y0 F300 G64	
N20	; odpadá
N30 X10 Y20	
N40 X20 Y40	
N50 X30 Y30	
N60 X40 Y45	
N70 X50 Y0	



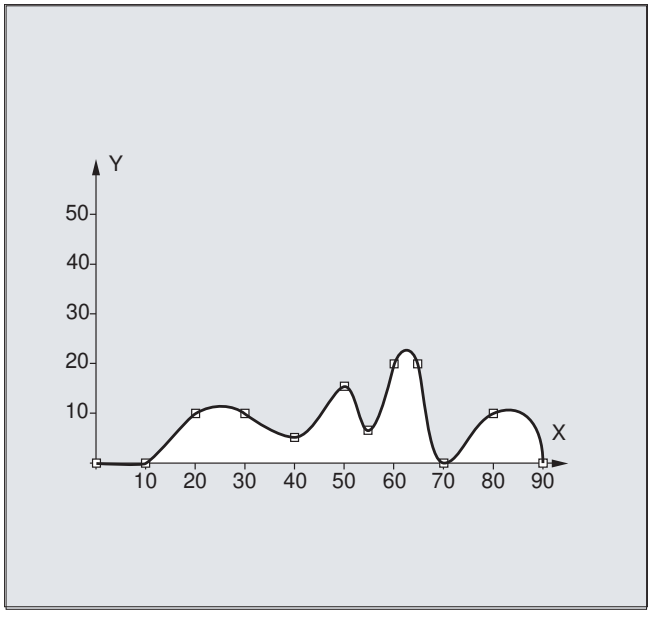
Příklad 2: C-spline, na začátku a na konci nulové zakřivení

Programový kód

```
N10 G1 X0 Y0 F300
N15 X10
N20 BNAT ENAT
N30 CSPLINE X20 Y10
```

Programový kód

```
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
```



Příklad 3: Splinová interpolace (A-spline) a transformace souřadného systému (ROT)

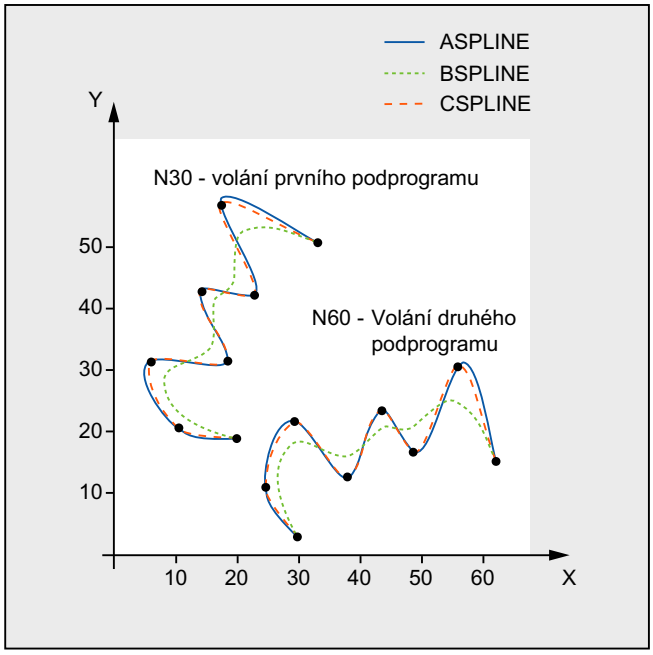
Hlavní program:

Programový kód	Komentář
N10 G00 X20 Y18 F300 G64	; najetí na počáteční bod
N20 ASPLINE	; aktivování typu interpolace A-Spline
N30 KONTUR	; první volání podprogramu
N40 ROT Z-45	; transformace souřadného systému: Otočení WCS o -45° okolo osy Z.
N50 G00 X20 Y18	; najíždění na počáteční bod kontury
N60 KONTUR	; druhé volání podprogramu
N70 M30	; konec programu

Podprogram "Kontura" (obsahuje souřadnice uzlového bodu):

Programový kód
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

V následujícím obrázku jsou vedle splinové křivky, která vyplynula z příkladu programování (ASPLINE), obsaženy ještě i splinové křivky, které by vznikly při aktivování interpolace pomocí B- nebo C-splinů (BSPLINE, CSPLINE):



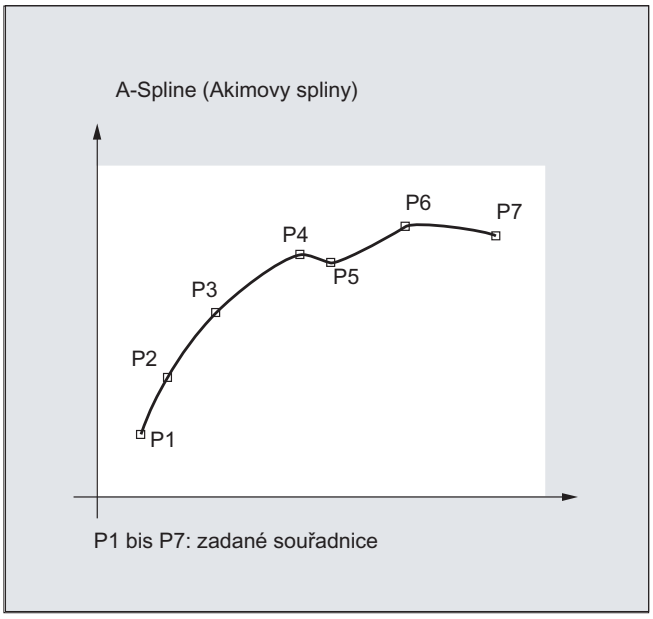
Další informace

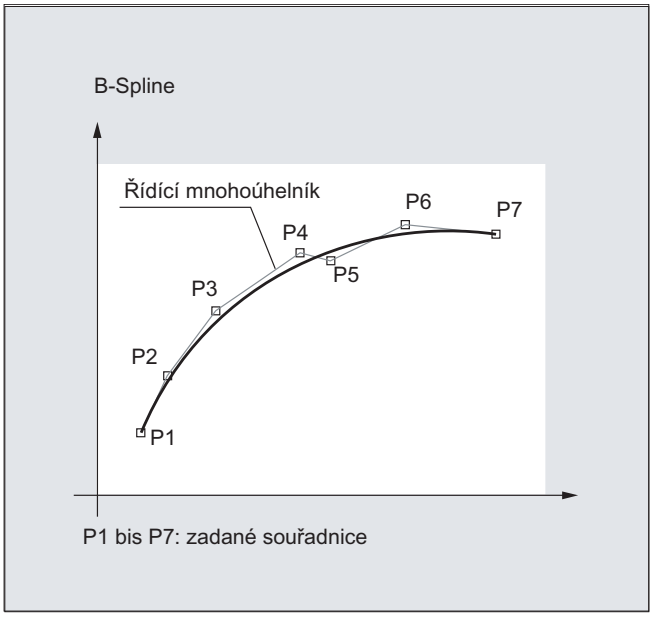
Výhody splinové interpolace

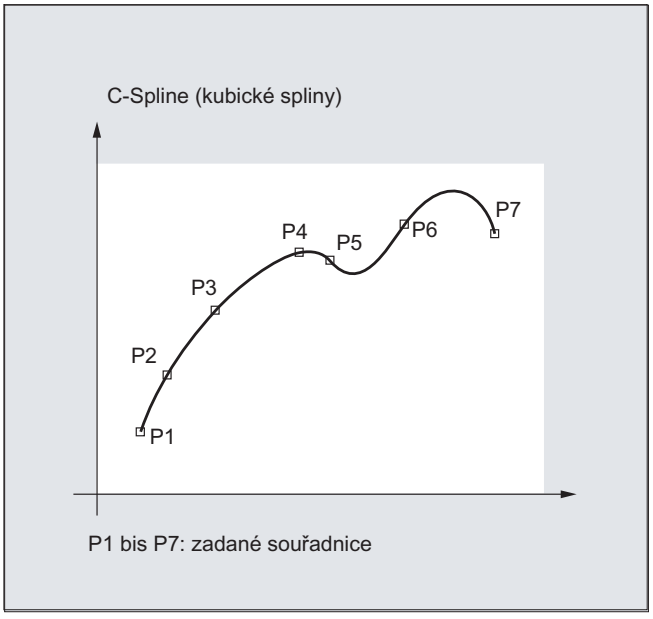
Použitím splinové interpolace se dá oproti použití přímkové interpolace G01 dosáhnout následujících výhod:

- Snížení počtu potřebných bloků výrobního programu pro popis kontury
- Hladší průběh kontury na přechodech mezi programovými bloky, čímž se šetří mechanika stroje

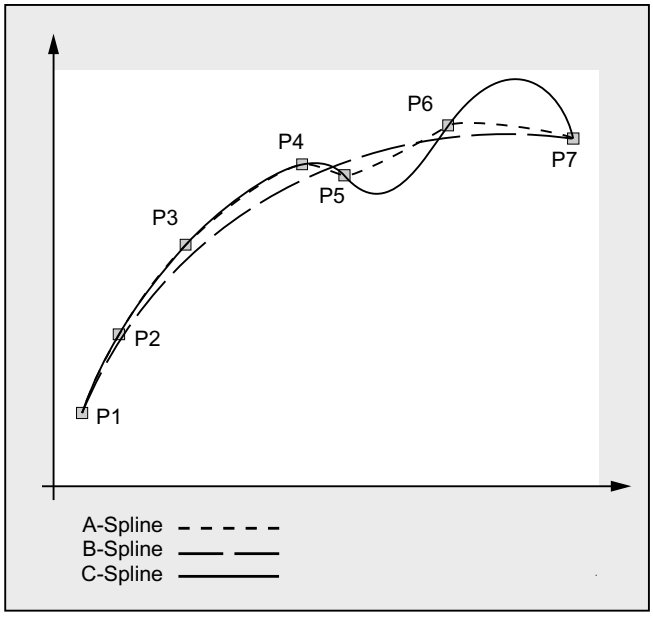
Vlastnosti a použití různých typů splinů

Typ splinu	Vlastnosti a použití
A-Spline	 **Vlastnosti:** • Probíhá přesně zadanými body. • Průběh křivky má tangenciální přechody, nemá však spojité zakřivení • Zřídka způsobuje nežádoucí vibrace. • Oblast vlivu změny uzlového bodu je lokální, tzn. změna jednoho uzlového bodu bude mít vliv pouze na maximálně 6 sousedících uzlových bodů. **Použití:** A-spliny se hodí především pro interpolaci křivkových průběhů s velkými změnami stoupání (např. pro schodovité průběhy křivek).

Typ splinu	Vlastnosti a použití
B-Spline	 **Vlastnosti:** • Křivka neprochází zadanými uzlovými body, nýbrž v jejich blízkosti. Křivka je přitahována k uzlovým bodům. Přiřazováním váhy k uzlovým bodům pomocí faktoru je možné průběh křivky ještě dodatečně ovlivňovat. • Průběh křivky má tangenciální přechody a spojité zakřivení • Nezpůsobuje žádné nežádoucí vibrace. • Oblast vlivu změny uzlového bodu je lokální, tzn. změna jednoho uzlového bodu bude mít vliv pouze na maximálně 6 sousedících uzlových bodů. **Použití:** B-spliny jsou zamýšleny především jako rozhraní pro systémy CAD.

Typ splinu	Vlastnosti a použití
C-Spline	 **Vlastnosti:** • Probíhá přesně zadanými body. • Průběh křivky má tangenciální přechody a spojité zakřivení • Často způsobuje nežádoucí vibrace, a to zejména na místech s velkými změnami stoupání. • Oblast vlivu změny uzlového bodu je globální, tzn. změna jednoho uzlového bodu bude mít vliv celý průběh křivky. **Použití:** C-spliny mohou být výborně uplatněny tehdy, pokud se uzlové body nacházejí na analyticky známé křivce (kruh, parabola, hyperbola).

Porovnání třech typů splinů pomocí stejných uzlových bodů



Minimální počet bloků splinu

Příkazy G-funkcí ASPLINE, BSPLINE a CSPLINE spojují koncové body bloků pomocí splinů. Za tím účelem musí být při předběžném zpracování současně vypočítána řada bloků (koncových bodů). Velikost vyrovnávací paměti pro tento výpočet standardně činí 10 bloků. Ne každá informace v bloku je koncovým bodem splinu. Řídící systém však potřebuje z 10 bloků určitý počet bloků s koncovými body splinů:

Typ splinu	Minimální počet bloků splinu
A-Spline:	Z každých 10 bloků musí být minimálně **4** bloky splinové. Bloky komentářů a výpočty parametrů se nepočítají.
B-Spline:	Z každých 10 bloků musí být minimálně **6** bloků splinových. Bloky komentářů a výpočty parametrů se nepočítají.
C-Spline:	Potřebný minimální počet splinových bloků vyplývá z následujícího součtu: Hodnota z parametru MD20160 $MC_CUBIC_SPLINE_BLOCKS + 1 Do parametru MD20160 se zadává počet bodů, přes které se splinový úsek vypočítává. Standardní nastavení je 8. Z každých 10 bloků musí proto být ve standardním případě minimálně **9** bloků splinových.

Poznámka

V případě překročení tolerovatelných hodnot se aktivuje alarm, stejně tak, jako když je osa podílející se na splinu naprogramována jako polohovací osa.

Shrnutí krátkých splinových bloků

Při splinové interpolaci mohou vzniknout krátké splinové bloky, což má za následek zbytečné snížení rychlosti pohybu po dráze. Pomocí funkce "Shrnutí krátkých splinových bloků" mohou být tyto bloky soustředěny takový způsobem, že výsledná délka bloků je dostatečně velká a nemá za následek snížení rychlosti pohybu po dráze.

Funkce se aktivuje pomocí kanálového strojního parametru:

MD20488 $MC_SPLINE_MODE (Nastavení pro splinovou interpolaci)

Další informace: Příručka k funkcím, Základní funkce

4.7.3 Splinový svazek (SPLINEPATH)

Pomocí příkazu `SPLINEPATH` jsou vybrány osy, které mají být interpolovány ve splinovém svazku. Při splinové interpolaci je možné pracovat s až osmi dráhovými osami.

Poznámka

Jestliže příkaz `SPLINEPATH` není explicitně naprogramován, zachází se s prvními třemi osami kanálu jako se splinovým svazkem.

Syntaxe

Definice splinového svazku se uskutečňuje v samostatném bloku:

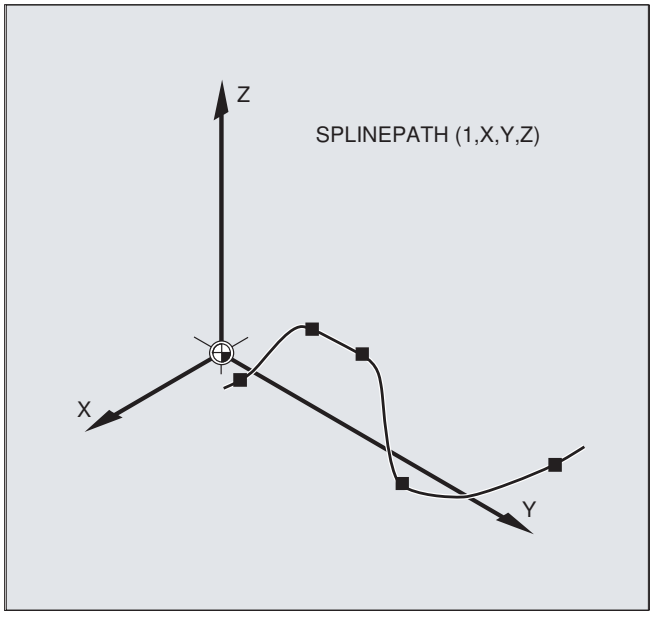
```
SPLINEPATH(n,X,Y,Z,…)
```

Význam

`SPLINEPATH`:	Příkaz pro vytvoření splinového svazku.
`n`:	=1 (pevná hodnota)
`X,Y,Z,…` :	Identifikátory dráhových os, které mají být interpolovány ve splinovém svazku.

Příklad: Splinový svazek se třemi dráhovými osami

Programový kód	**Komentář**
`N10 G1 X10 Y20 Z30 A40 B50 F350`	
`N11 SPLINEPATH(1,X,Y,Z)`	`; splinový svazek`
`N13 CSPLINE BAUTO EAUTO X20 Y30 Z40 A50 B60`	`; C-Spline`
`N14 X30 Y40 Z50 A60 B70`	`; uzlové body`
…	
`N100 G1 X… Y…`	`; deaktivování splinové interpolace`



4.7.4 Aktivování/deaktivování komprese NC bloků (COMPON, COMPCURV, COMPCAD, COMPSURF, COMPOF)

Funkce pro kompresi lineárních bloků (a v závislosti na nastavení parametrů také kruhových bloků a bloků rychlým posuvem) se aktivuje/deaktivuje pomocí příkazů G-funkcí ze skupiny 30. Příkazy mají modální platnost.

Syntaxe

```
COMPON / COMPCURV / COMPCAD / COMPSURF
...
COMPOF
```

Význam

COMPON:	Aktivování funkce kompresoru COMPON
COMPCURV:	Aktivování funkce kompresoru COMPCURV
COMPCAD:	Aktivování funkce kompresoru COMPCAD
COMPSURF:	Aktivování funkce kompresoru COMPSURF
COMPOF:	Deaktivování momentálně aktivní funkce kompresoru

Poznámka

K dalšímu zlepšení jakosti povrchu se mohou použít ještě i funkce pro přechodová zaoblení G642 a funkce pro omezení ryvu SOFT. Tyto příkazy je zapotřebí zapsat na začátku programu.

Příklad: COMPCAD

Programový kód	Komentář
N10 G00 X30 Y6 Z40	
N20 G1 F10000 G642	; aktivování: funkce pro přechodová zaoblení G642
N30 SOFT	; aktivování: Omezení ryvu SOFT
N40 **COMPCAD**	; aktivování: Funkce kompresoru COMPCAD
N50 FIFOCTRL	
N24050 Z32.499	; 1. blok posuvu
N24051 X41.365 Z32.500	; 2. blok posuvu
...	
N99999 X... Z...	; poslední blok posuvu
COMPOF	; vypnutí funkce kompresoru
...	

4.7.5 Polynomická interpolace (POLY, POLYPATH, PO, PL)

V zásadě se u polynomické interpolace (POLY) nejedná o druh splinové interpolace. V první řadě je míněna jako rozhraní pro programování externě vytvořených splinových křivek. Tímto způsobem mohou být splinové úseky naprogramovány přímo.

Tento druh interpolace zbavuje NC systém nutnosti vypočítávat koeficienty polynomu. Může být optimálně využit tehdy, pokud jsou koeficienty dodávány přímo ze systému CAD nebo ze systému pro dodatečné zpracování.

Syntaxe

Polynom 3. stupně:
```
POLY PO[X]=(xe,a2,a3) PO[Y]=(ye,b2,b3) PO[Z]=(ze,c2,c3) PL=n
```

Polynomy 5. stupně a nová syntaxe polynomu:
```
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")
```

Poznámka

Součet koeficientů polynomu naprogramovaných v NC bloku a os nesmí překročit maximální přípustný počet os najeden blok.

Význam

POLY :	Polynomická interpolace se aktivuje blokem s příkazem POLY.
POLYPATH :	Polynomická interpolace může být vybrána pro obě skupiny os AXIS nebo VECT
PO[identifikátor osy/proměnná] :	Koncové body a koeficienty polynomu

`X, Y, Z:`	Identifikátor osy
`xe, ye, ze :`	Údaj koncové pozice pro příslušné osy, rozsah hodnot a jednotka
`a2, a3, a4, a5:`	Koeficienty a_2, a_3, a_4 a a_5 jsou popsány svou hodnotou, rozsahem hodnot a jednotkou. Poslední koeficient může odpadnout, jestliže má nulovou hodnotu.
`PL:`	Délka intervalu parametru, na kterém je polynom definován (definiční oblast funkce f(p)). Interval má svůj počátek vždy v 0, p může nabývat hodnot od 0 do PL. Teoretický rozsah hodnot pro PL: 0,0001 ... 99 999,9999 **Upozornění** Hodnota PL platí pro blok, ve kterém se nachází. Pokud hodnota PL není naprogramována, platí PL =1.

Aktivování/deaktivování polynomické interpolace.

Polynomická interpolace se ve výrobním programu zapíná příkazem G-funkce `POLY`.

Příkaz G-funkce `POLY` patří spolu s příkazy `G0`, `G1`, `G2`, `G3`, `ASPLINE`, `BSPLINE` a `CSPLINE` do 1. G-skupiny.

Osy, které jsou naprogramovány pouze svým názvem a koncovým bodem (např. `X10`) se pohybují lineárně. Pokud jsou tímto způsobem naprogramovány všechny osy v NC bloku, chová se řídící systém jako při funkci G1.

Naprogramováním jiného příkazu z 1. skupiny (např. `G0`, `G1`) se polynomická interpolace implicitně deaktivuje.

Koeficienty polynomu

Hodnota `PO (PO[]=)`, příp. `...=PO(...)` udává všechny koeficienty polynomu pro danou osu. V souladu se stupněm polynomu se větší počet hodnot zadává oddělených čárkou. V rámci jednoho bloku mohou být pro jednotlivé osy zadány polynomy různých stupňů.

Podprogram POLYPATH

Pomocí příkazu `POLYPATH(...)` může být polynomická interpolace selektivně odblokována pro určité skupiny os.

Jen dráhové osy a doplňkové osy:	`POLYPATH("AXES")`
Jen orientační osy: (při pohybech s transformací orientace)	`POLYPATH("VECT")`

Osy, které nebyly tímto způsobem odblokovány, se pohybují jako osy lineární.

Standardně je polynomická interpolace uvolněna pro obě skupiny os.

Naprogramováním příkazu `POLYPATH( )` bez parametrů se polynomická interpolace pro všechny osy deaktivuje.

Příklad:

Programový kód	Komentář
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	; aktivování polynomické interpolace
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	; smíšené údaje pro osy
N27 PO[X]=(10,2.5) PO[Y]=(2.3)	; hodnota PL není naprogramována, platí PL =1
N30 G1 X… Y… Z.	; deaktivování polynomické interpolace
…	

Příklad: Nová syntaxe polynomu

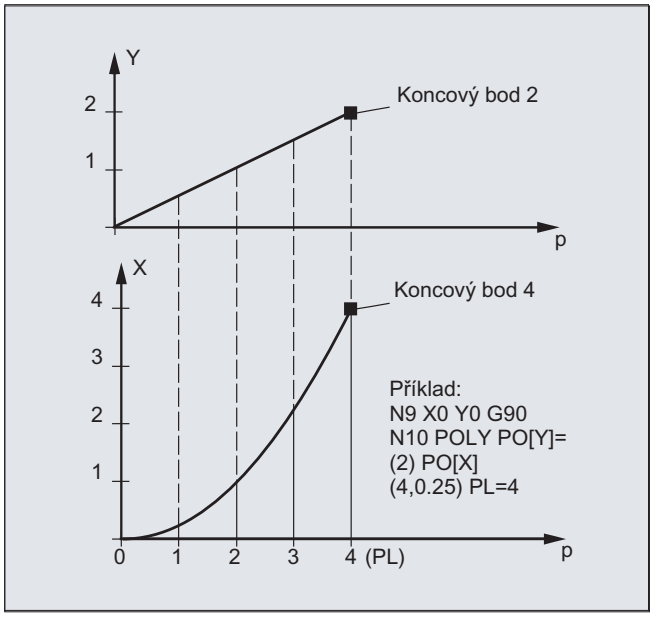
Dále platná syntaxe polynomu	Nová syntaxe polynomu
PO[identifikátor osy]=(.. , ..)	Identifikátor osy=PO(.. , ..)
PO[PHI]=(.. , ..)	PHI=PO(.. , ..)
PO[PSI]=(.. , ..)	PSI=PO(.. , ..)
PO[THT]=(.. , ..)	THT=PO(.. , ..)
PO[]=(.. , ..)	PO(.. , ..)
PO[proměnná]=IC(.. , ..)	proměnná=PO IC(.. , ..)

Příklad: Křivka v rovině X/Y

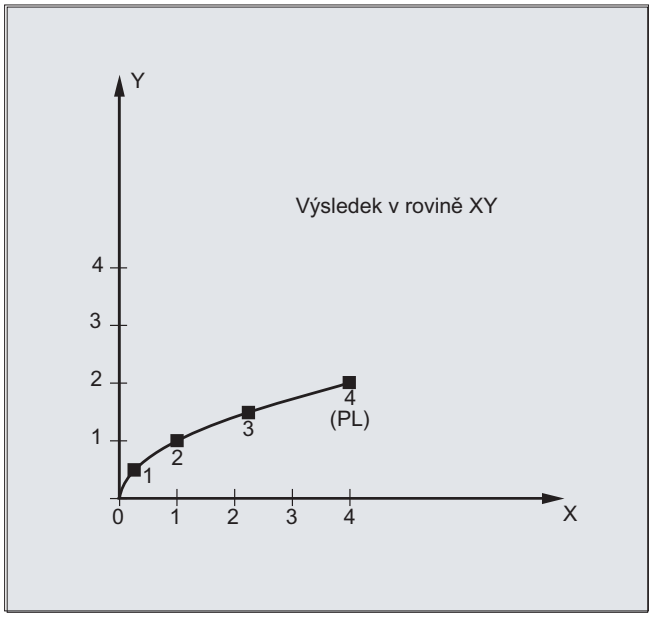
Programování

Programový kód
N9 X0 Y0 G90 F100
N10 POLY PO[Y]=(2) PO[X]=(4,0.25) PL=4

Průběh křivky X(p) a Y(p)



Průběh křivky v rovině XY



Popis

V obecném tvaru má polynomická funkce následující podobu:

$f(p)= a_0 + a_1p + a_2p^2 + . . . + a_np^n$

kde: a_i: konstantní koeficienty (i = 0, 1, ..., n)
p: Parametr

V řídícím systému je možné programovat maximálně polynomy 5. stupně:

$$f(p)= a_0 + a_1p + a_2p^2 + a_3p^3 + a_4p^4 + a_5p^5$$

Dosazením konkrétních hodnot koeficientům je možné vytvářet různé průběhy křivek, jako jsou přímky, paraboly a mocninné funkce.

Přímka vzniká dosazením $a_2 = a_3 = a_4 = a_5 = 0$:

$$f(p)= a_0 + a_1p$$

Kromě toho platí:

$a_{0:}$ Poloha osy na konci předcházejícího bloku
$p = PL$
$a_1 = (x_E - a_0 - a_2*p^2 - a_3*p^3) / p$

Je možné naprogramovat polynomy, **aniž** by byla polynomická interpolace příkazem G-funkce `POLY` aktivována. V tomto případě nejsou interpolovány naprogramované polynomy, ale na naprogramované koncové body os se najíždí po lineárních drahách (`G1`). Teprve po explicitním aktivování polynomické interpolace ve výrobním programu (`POLY`) jsou pohyby uskutečňovány po křivkových drahách odpovídajících naprogramovaným polynomům.

Zvláštnost: Polynom jmenovatele

Pro geometrické osy je možno pomocí příkazu `PO[]=(...)` bez udání názvu osy naprogramovat také společný polynom jmenovatele, což znamená, že pohyb geometrické osy bude potom interpolován jako podíl dvou polynomů.

Tímto způsobem se dají přesně zobrazit např. kuželosečky (kruh, elipsa, parabola, hyperbola).

Příklad:

Programový kód	**Komentář**
`POLY G90 X10 Y0 F100`	`; geometrické osy lineárně najíždějí na pozici X10 Y0`
`PO[X]=(0,-10) PO[Y]=(10) PO[]=(2,1)`	`; geometrické osy najíždějí po čtvrtkruhu na bod X0 Y10`

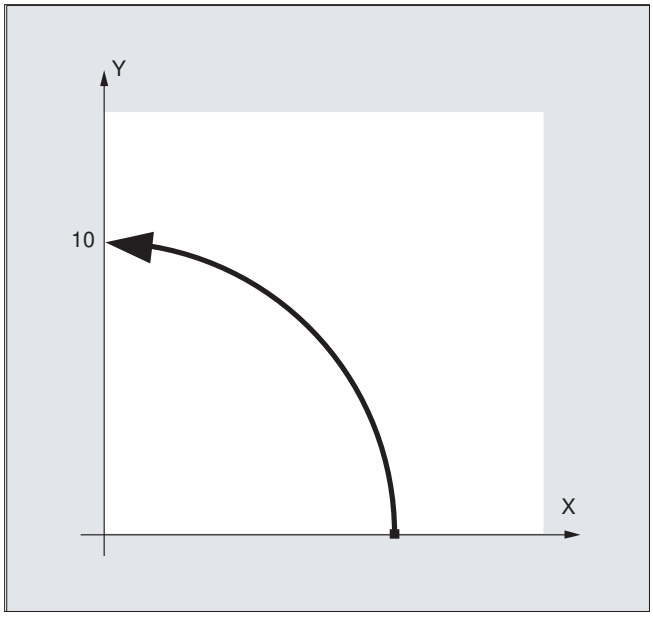
Koeficient konstanty (a_0) polynomu jmenovatele je vždy nastaven na 1. Naprogramovaný koncový bod je nezávislý na příkazech `G90` / `G91`.

Z naprogramovaných hodnot se vypočítají hodnoty X(p) a Y(p):

$X(p) = (10 - 10 * p^2) / (1 + p^2)$
$Y(p) = 20 * p / (1 + p^2)$
kde $0 \le p \le 1$

Na základě naprogramovaných počátečních bodů, koncových bodů, koeficientu a_2 a PL=1 vyplývají následující mezivýsledky:

Čitatel (X) = $10 + 0 * p - 10 * p^2$
Čitatel (Y) = $0 + 20 * p + 0 * p^2$
Jmenovatel = $1 + p^2$



Když je aktivována polynomická interpolace, je naprogramování polynomu jmenovatele s nulovými koeficienty v rámci intervalu `[0,PL]` odmítnuto a aktivuje se alarm. Pohyb doplňkových os nemá na polynom jmenovatele žádný vliv.

Poznámka

Korekce rádiusu nástroje může být v případě polynomické interpolace pomocí příkazů `G41`, `G42` aktivována/deaktivována a může se používat stejně jako při přímkové nebo kruhové interpolaci.

4.7.6 Nastavitelný vztah k dráze (SPATH, UPATH)

V případě polynomické interpolace (`POLY`, `ASPLINE`, `BSPLINE`, `CSPLINE`, `COMP...`) jsou polohy dráhové osy i definovány prostřednictvím polynomu $p_i(U)$. Křivkový parametr U se přitom v rámci NC bloku pohybuje v hodnotách od 0 do 1.

Pomocí příkazu `FGROUP` jsou vybrány osy (osy FGROUP), na které se vztahuje posuv po dráze `F`. Interpolace s konstantní rychlostí na úseku dráhy S u těchto os FGROUP však v průběhu polynomické interpolace zpravidla znamená nekonstantní změnu křivkového parametru U. Pro osy, které nejsou obsaženy v příkazu `FGROUP`, si proto můžete vybrat ze dvou možností, jak tyto osy mají sledovat osy FGROUP:

- Synchronizovaně úsekem dráhy S (`SPATH`)
- Synchronizovaně vůči křivkovému parametru U (`UPATH`)

Syntaxe

```
SPATH
UPATH
```

Význam

SPATH:	Osy, které nejsou obsaženy v příkazu FGROUP, se budou pohybovat ve vztahu k úseku dráhy S
UPATH:	Osy, které nejsou obsaženy v příkazu FGROUP, se budou pohybovat ve vztahu ke křivkovému parametru U

Poznámka

Funkce UPATH a SPATH určují také souvislost mezi polynomem F-slova (FPOLY, FCUB, FLIN) a pohybem po dráze.

Okrajové podmínky

Příkazy SPATH, příp. UPATH, nemají v následujících případech žádný význam:

- Lineární interpolace (G1)
- Kruhová interpolace (G2, G3)
- Bloky pro výrobu závitu (G33, G34, G35, G33x, G63)
- Všechny dráhové osy jsou obsaženy ve skupině FGROUP

Příklad

Následující příklad ukazuje rozdíl mezi oběma druhy řízení pohybu.

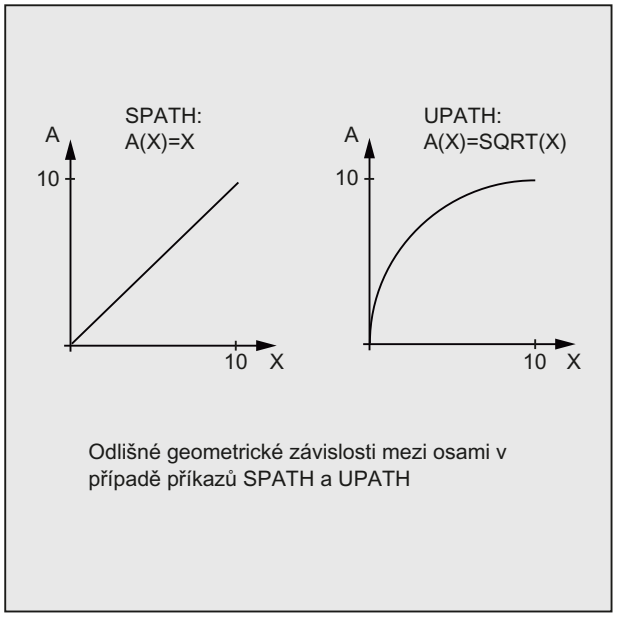
Programový kód

```
N10 FGROUP(X,Y,Z)
N15 G1 X0 A0 F1000 SPATH                ; SPATH
N20 POLY PO[X]=(10,10) A10
```

Programový kód

```
N10 FGROUP(X,Y,Z)
N15 G1 X0 A0 F1000 UPATH                ; UPATH
N20 POLY PO[X]=(10,10) A10
```

V obou úsecích programu závisí v bloku N20 dráha S os ve skupině FGROUP na druhé mocnině křivkového parametru U. Z toho vyplývají různé polohy synchronní osy A podél dráhy osy X podle toho, jestli je aktivován příkaz SPATH nebo UPATH.



Odlišné geometrické závislosti mezi osami v případě příkazů SPATH a UPATH

Další informace

Chování řídícího systému při resetu a strojní parametry/parametry doplňků

Po resetu je prostřednictvím parametru MD20150 $MC_GCODE_RESET_VALUES[44] aktivován příkaz určité G-funkce (45. G-skupina).

Hodnota základního nastavení pro druh přechodových zaoblení je definována parametrem MD20150 $MC_GCODE_RESET_VALUES[9] (10. G-skupina).

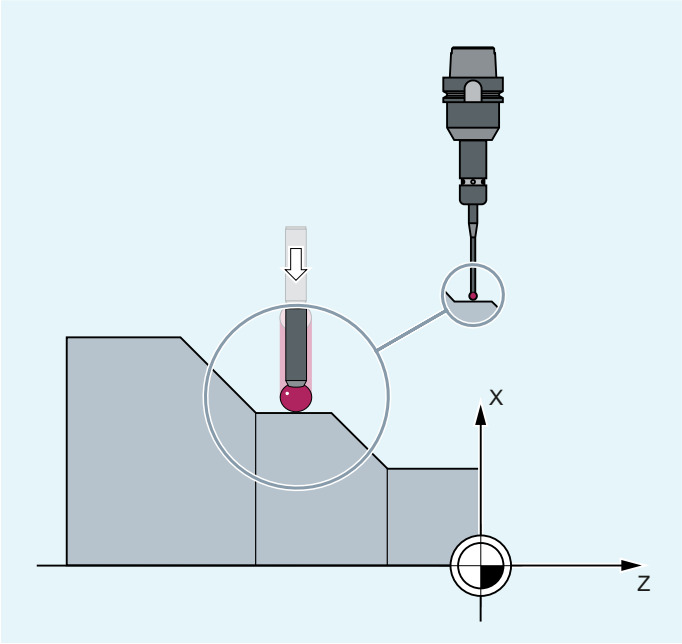
Osový strojní parametr MD33100 $MA_COMPRESS_POS_TOL[<n>] má rozšířený význam: Obsahuje tolerance pro funkci kompresoru a pro přechodová zaoblení pomocí příkazu G642.

4.7.7 Měření ve specifickém kanálu (MEAS, MEAW)

V případě měření ve specifickém kanálu probíhá aktivování měřicí operace pro daný kanál v NC systému vždy z výrobního programu, který je v příslušném kanálu zpracováván. V měřicím bloku je vždy naprogramována **jedna** spouštěcí událost (náběžná nebo klesající hrana signálu měřicí sondy) a **jeden** režim měření s vymazáním zbytkové dráhy (MEAS) nebo bez vymazání zbytkové dráhy (MEAW). Na měřicí operaci se potom podílejí **všechny** osy naprogramované v bloku měření.

Jakmile je blok s příkazem měření aktivován, měřicí sonda najíždí na obrobek. Když je zaznamenána spínací hrana signálu měřicí sondy, pro všechny osy naprogramované v měřicím bloku jsou změřeny jejich pozice a pro všechny osy jsou tyto údaje zapsány do příslušných paměťových buněk.

Výsledky měření pak mohou být načteny ve výrobním programu nebo pomocí synchronních akcí, a to jak v souřadném systému stroje, tak také v souřadném systému obrobku.



Syntaxe

```
MEAS=<TE> G... X... Y... Z...
MEAW=<TE> G... X... Y... Z...
```

Význam

MEAS:	Měření **s** vymazáním zbytkové dráhy		
	Platnost:		bloková
MEAW:	Měření **bez** vymazání zbytkové dráhy		
	Platnost:		bloková
	Použití:		Pro měřicí úlohy, u kterých se má v každém případě najet na naprogramovanou pozici.
<TE>:	Událost sepnutí za účelem spuštění měření		
	Typ:	INT	
	Rozsah hodnot:		-2, -1, 1, 2
	Hodnota:	(+)1	náběžná hrana signálu měřicí sondy 1 (na měřicím vstupu 1)
		-1	sestupná hrana signálu měřicí sondy 1 (na měřicím vstupu 1)
		(+)2	náběžná hrana signálu měřicí sondy 2 (na měřicím vstupu 2)
		-2	sestupná hrana signálu měřicí sondy 2 (na měřicím vstupu 2)
	Upozornění: Existují maximálně 2 měřicí sondy (v závislosti na úrovni vybavení).		
G...:	Druh interpolace, např. G0, G1, G2 nebo G3		
X... Y... Z...:	Koncové body v kartézských souřadnicích		

Poznámka

Příkazy MEAS a MEAW mají blokovou platnost a jsou naprogramovány spolu s příkazy pro pohyb. Posuv a druh interpolace (G0, G1, ...), stejně jako počet os, přitom musí být přizpůsobeny konkrétnímu měřicímu problému.

Příklad:

Programový kód	Komentář
...	
N10 MEAS=1 G1 F1000 X100 Y730 Z40	; blok měření s měřicí sondou na prvním měřicím vstupu a přímková interpolace Je automaticky generováno zastavení předběžného zpracování.
...	

Další informace

Zjištění stavu

Jestliže je v programu požadováno vyhodnocení, zda byla měřicí sonda vychýlena, resp. zda sepnula, pak je možné její stav zjistit prostřednictvím následujících systémových proměnných:

Systémová proměnná	Význam	Datový typ	Hodnota	
$A_PROBE[<n>]	Stav měřicí sondy	INT	0	Měřicí sonda není vychýlena.
			1	Měřicí sonda je vychýlena.
$AC_MEA[<n>]	Stav sepnutí měřicí sondy Na začátku měření se parametr $AC_MEA[<n>] automaticky vynuluje.	INT	0	Měřicí sonda nesepnula.
			1	Měřicí sonda sepnula.

<n> = číslo měřicí sondy

Odečítání změřených hodnot

Při měření ve specifickém kanálu jsou zaznamenávány pozice všech dráhových a polohovacích os v bloku, které se pohybovaly (maximální počet os závisí na konfiguraci řídícího systému). V případě příkazu MEAS se pohyb po sepnutí měřicí sondy definovaným způsobem zastaví.

Poznámka

Jestliže je v měřicím bloku naprogramována geometrická osa, jsou změřené hodnoty uloženy pro všechny momentálně definované geometrické osy.

Jestliže je v měřicím bloku naprogramována osa podílející se na nějaké transformaci, jsou změřené hodnoty uloženy i pro všechny na této transformaci se podílející osy.

Načtení výsledků měření

Údaje poloh os nasnímané měřicí sondou mohou být načítány ve výrobním programu a v synchronních akcích prostřednictvím následujících systémových proměnných:

Systémová proměnná	Význam
$AA_MM[<osa>]	Hodnoty změřené měřicí sondou v souřadném systému stroje.
$AA_MW[<osa>]	Hodnoty změřené měřicí sondou v souřadném systému obrobku.

4.7.8 Měření pomocí specifické osy (MEASA, MEAWA, MEAC) (volitelný doplněk)

V případě měření pomocí specifické osy může být aktivování realizováno ve výrobním programu **nebo** v synchronních akcích. Jestliže jsou pro danou osu k dispozici dva měřicí systémy, mohou být pro měření použity oba dva.

K dispozici jsou následující metody měření:

- Měření s vymazáním zbytkové dráhy (MEASA)
- Měření bez mazání zbytkové dráhy (MEAWA)
- Kontinuální měření bez mazání zbytkové dráhy (MEAC)

Pomocí příkazů MEASA, příp. MEAWA jsou pro příslušnou naprogramovanou osu zaznamenány v každém měření až čtyři změřené hodnoty, které se pak ukládají podle události spouštění do systémových proměnných.

V případě kontinuálního měření pomocí příkazu MEAC jsou výsledky měření ukládány do proměnných typu FIFO.

Syntaxe

```
MEASA[<osa>]=(<režim>,<TE1>,...,<TE4>)
MEAWA[<osa>]=(<režim>,<TE1>,...,<TE4>)
MEAC[<osa>]=(<režim>,<MeasMem>,<TE1>,...,<TE4>)
```

Poznámka

Příkazy MEASA a MEAWA mají blokovou platnost a mohou být naprogramovány společně v jednom bloku. Pokud je oproti tomu příkaz MEASA/MEAWA naprogramován v jednom bloku spolu s příkazem MEAS/MEAW, je generováno chybové hlášení.

Význam

MEASA:	Měření se specifickou osou a s vymazáním zbytkové dráhy	
	Platnost:	bloková
MEAWA:	Měření se specifickou osou a bez vymazání zbytkové dráhy	
	Platnost:	bloková
MEAC:	Kontinuální měření se specifickou osou bez vymazání zbytkové dráhy	
	Platnost:	bloková

<osa>:	Název kanálové osy použité pro měření	
<režim>:	Dvoumístné číslo (xx) pro zadávání provozního režimu (režim měření a měřicí systém)	
	Místo jednotek: Režim měření Tento parametr definuje, zda mají být spouštěcí události aktivovány v časové nebo naprogramované posloupnosti.	
	x**0**	Přerušení měřicí úlohy.
	x**1**	Až 4 **různé** spouštěcí události, které mohou být aktivovány současně.
	x**2**	Až 4 spouštěcí události, které mohou být aktivovány jedna po druhé.
	x**3**	Až 4 spouštěcí události, které mohou být aktivovány jedna po druhé, avšak žádné monitorování spouštěcí události 1 při spuštění (alarmy 21700/21703 jsou potlačeny). **Upozornění** V případě příkazu MEAC není tento režim možný.
	Místo desítek: Měřicí systém Tento parametr definuje, pomocí kterého měřicího systému má být měření provedeno.	
	0x (nebo žádný údaj)	aktivní měřicí systém
	1x	Měřicí systém 1
	2x	Měřicí systém 2
	3x	Oba měřicí systémy
<TE>:	Událost sepnutí za účelem spuštění měření	
	Typ:	INT
	Rozsah hodnot:	-2, -1, 1, 2
	(+)1	náběžná hrana měřicí sondy 1
	-1	sestupná hrana měřicí sondy 1
	(+)2	náběžná hrana měřicí sondy 2
	-2	sestupná hrana měřicí sondy 2
	Poznámka: Pokud má být měřicí operace provedena se dvěma měřicími systémy, mohou být naprogramovány maximálně dvě spouštěcí události (náběžná nebo sestupná hrana signálu). Při každé z těchto spouštěcích událostí se zaznamenají změřené hodnoty z obou dvou měřicích systémů.	
<MeasMem>:	Číslo FIFO (zásobníková paměť)	

Příklady

Příklad 1: Měření se specifickou osou a s vymazáním zbytkové dráhy v režimu 1 (vyhodnocování v časové posloupnosti)

a) Měření s jedním měřicím systémem

Programový kód	**Komentář**
...	
N100 **MEASA[X]=(1,1,-1)** G01 X100 F100	; měření v režimu 1 s aktivním měřicím systémem Čekání na měřicí signál s náběžnou/sestupnou hranou signálu měřicí sondy 1 na dráze po X=100.

Programový kód	**Komentář**
N110 IF $AC_MEA[1]==FALSE GOTOF ENDE	; kontrola úspěšného měření
N120 R10=$AA_MM1[X]	; uložit změřenou hodnotu patřící k první naprogramované spouštěcí události (náběžná hrana)
N130 R11=$AA_MM2[X]	; uložit změřenou hodnotu patřící ke druhé naprogramované spouštěcí události (sestupná hrana)
N140 ENDE:	

b) Měření se dvěma měřicími systémy

Programový kód	**Komentář**
...	
N200 **MEASA[X]=(31,1,-1)** G01 X100 F100	; měření v režimu 1 s oběma měřicími systémy Čekání na měřicí signál s náběžnou/sestupnou hranou signálu měřicí sondy 1 na dráze po X=100.
N210 IF $AC_MEA[1]==FALSE GOTOF ENDE	; kontrola úspěšného měření
N220 R10=$AA_MM1[X]	; uložení změřené hodnoty z měřicího systému 1 při náběžné hraně signálu
N230 R11=$AA_MM2[X]	; uložení změřené hodnoty z měřicího systému 2 při náběžné hraně signálu
N240 R12=$AA_MM3[X]	; uložení změřené hodnoty z měřicího systému 1 při sestupné hraně signálu
N250 R13=$AA_MM4[X]	; uložení změřené hodnoty z měřicího systému 2 při sestupné hraně signálu
N260 ENDE:	

Příklad 2: Měření se specifickou osou a s vymazáním zbytkové dráhy v režimu 2 (vyhodnocování v naprogramované posloupnosti)

Programový kód	**Komentář**
...	
N100 **MEASA[X]=(2,1,-1,2,-2)** G01 X100 F100	; měření v režimu 2 s aktivním měřicím systémem Čekání na měřicí signály v posloupnosti náběžná hrana měřicí sondy 1, sestupná hrana měřicí sondy 1, náběžná hrana měřicí sondy 2, sestupná hrana měřicí sondy 2 na dráze po X=100.
N110 IF $AC_MEA[1]==FALSE GOTOF MESSTASTER2	; kontrola úspěšného měření měřicí sondou 1
N120 R10=$AA_MM1[X]	; uložit změřenou hodnotu patřící k první naprogramované spouštěcí události (náběžná hrana měřicí osy 1)
N130 R11=$AA_MM2[X]	; uložit změřenou hodnotu patřící ke druhé naprogramované spouštěcí události (náběžná hrana měřicí osy 1)
N140 MESSTASTER2:	

Programový kód	**Komentář**
N150 IF $AC_MEA[2]==FALSE GOTOF ENDE	; kontrola úspěšného měření měřicí sondou 2
N160 R12=$AA_MM3[X]	; uložit změřenou hodnotu patřící ke třetí naprogramované spouštěcí události (náběžná hrana měřicí osy 2)
N170 R13=$AA_MM4[X]	; uložit změřenou hodnotu patřící ke čtvrté naprogramované spouštěcí události (náběžná hrana měřicí osy 2)
N180 ENDE:	

Příklad 3: Kontinuální měření se specifickou osou v režimu 1 (vyhodnocování v časové posloupnosti)

a) Měření až 100 změřených hodnot

Programový kód	**Komentář**
...	
N110 DEF REAL MESSWERT[100]	
N120 DEF INT Schleife=0	
N130 **MEAC[X]=(1,1,-1)** G01 X1000 F100	; měření v režimu 1 s aktivním měřicím systémem, s ukládáním změřených hodnot do proměnné $AC_FIFO1, čekání na měřicí signál se sestupnou hranou signálu měřicí sondy 1 na dráze po X=1000
N135 STOPRE	
N140 MEAC[X]=(0)	; ukončení měření po dosažení polohy osy
N150 R1=$AC_FIFO1[4]	; ukládání počtu zaznamenaných změřených hodnot do parametru R1
N160 FOR Schleife=0 TO R1-1	
N170 MESSWERT[Schleife]=$AC_FIFO1[0]	; načítání a ukládání změřených hodnot z proměnné $AC_FIFO1
N180 ENDFOR	

b) Měření s vymazáním zbytkové dráhy po 10 změřených hodnotách

Programový kód	**Komentář**
...	
N10 WHEN $AC_FIFO1[4]>=10 DO MEAC[x]=(0) DELDTG(x)	; vymazání zbytkové dráhy
N20 **MEAC[x]=(1,1,1,-1)** G01 X100 F500	
N30 MEAC[X]=(0)	
N40 R1=$AC_FIFO1[4]	; počet změřených hodnot
...	

c) Měření sestupného/náběžného boku zubu pomocí 2 měřicích sond

Programový kód	**Komentář**
...	
N110 DEF REAL MESSWERT[16]	
N120 DEF INT Schleife=0	
N130 **MEAC[X]=(1,1,-1,2)** G01 X100 F100	; měření v režimu 1 s aktivním měřicím systémem, s ukládáním změřených hodnot do proměnné $AC_FIFO1, čekání na měřicí signál v posloupnosti sestupná hrana signálu měřicí sondy 1, vzestupná hrana měřicí sondy 2 na dráze po X=100
N140 STOPRE	; Zastavení předběžného zpracování.
N150 MEAC[X]=(0)	; ukončení měření po dosažení polohy osy
N160 R1=$AC_FIFO1[4]	; ukládání počtu zaznamenaných změřených hodnot do parametru R1
N170 FOR Schleife=0 TO R1-1	
N180 MESSWERT[Schleife]=$AC_FIFO1[0]	; načítání a ukládání změřených hodnot z proměnné $AC_FIFO1
N190 ENDFOR	

Další informace

Měřicí úloha

Programování měřicích úloh se může uskutečňovat ve výrobním programu nebo ze synchronní akce. Pro každou osu přitom může být v jednom a tomtéž okamžiku aktivní jen jedna měřicí úloha.

Poznámka

Posuv je potřeba přizpůsobit příslušnému měřicímu problému.

U příkazů MEASA a MEAWA mohou být správné výsledky zaručeny jedině při hodnotách posuvu, při kterých se nemůže vyskytnout více než jedna stejná a více než 4 různé spouštěcí události na jeden takt regulátoru polohy.

V případě kontinuálního měření s funkcí MEAC nesmí být poměr mezi taktem interpolátoru a taktem polohového regulátoru větší než 1:8.

Spouštěcí událost

Spouštěcí událost se skládá z čísla měřicí sondy a ze spouštěcího kritéria (náběžná nebo sestupná hrana) signálu měřicí sondy.

Pro každé měření mohou být zpracovány vždy až 4 spouštěcí události spínající měřicí sondy, tedy až dvě měřicí sondy, každá se dvěma hranami měřicího signálu. Posloupnost zpracování, jakož i také maximální počet spouštěcích událostí jsou přitom závislé na zvoleném režimu.

Pokud má být měřicí operace provedena se dvěma měřicími systémy, mohou být naprogramovány maximálně dvě spouštěcí události (náběžná nebo sestupná hrana signálu). Při každé z těchto spouštěcích událostí se zaznamenají změřené hodnoty z obou dvou měřicích sond.

Jestliže je používán telegram PROFIBUS 391 (standardní nastavení pro komunikaci po sběrnici PROFIBUS), je možná jen jedna změřená hodnota na jednu spouštěcí událost a takt regulátoru polohy.

Jestliže se používá telegram PROFIBUS 395, může být ve spojení s příkazem MEAC v provozním režimu 1 zvýšen počet změřených hodnot na celkem 8 změřených hodnot s náběžnou hranou a 8 hodnot s klesající hranou na jednu spouštěcí událost a takt regulátoru polohy:

- Jedna měřicí sonda: 8 změřených hodnot s náběžnou a 8 s klesající hranou signálu
- Dvě měřicí sondy: 4 změřené hodnoty s náběžnou a 4 s klesající hranou signálu na jednu měřicí sondu

Díky tomu je možno při použití telegramu PROFIBUS 395 realizovat vyšší rychlosti posuvu nebo otáčky.

Provozní režim

Pomocí první číslice (místo desítek) provozního režimu se vybírá požadovaný měřicí systém. Jestliže je k dispozici jen jeden měřicí systém, je však naprogramován ten druhý, automaticky se použije ten, který je k dispozici.

Pomocí druhé číslice (místo jednotek) se nastavuje požadovaný režim měření. Tímto způsobem může být měřicí operace možnostem příslušného řídícího systému.

- **Režim1**
  Vyhodnocování spouštěcích událostí se uskutečňuje v časové posloupnosti, v jaké se vyskytly.
- **Režim 2**
  Vyhodnocování spouštěcích událostí se uskutečňuje v naprogramované posloupnosti.
- **Režim 3**
  Vyhodnocování spouštěcí události se uskutečňuje v naprogramované posloupnosti, neprovádí se však žádné monitorování spouštěcí události 1 při spuštění.

Měření s a bez vymazání zbytkové dráhy

Jestliže je naprogramován příkaz MEASA, vymazání zbytkové dráhy se uskuteční až po zaznamenání všech požadovaných měřených hodnot.

Příkaz MEAWA se používá v případě speciálních měřicích úloh, u kterých se má v každém případě najet na naprogramovanou pozici.

Poznámka

Příkaz MEASA nemůže být naprogramován v synchronních akcích. Místo toho může být jako synchronní akce naprogramován příkaz MEAWA plus vymazání zbytkové dráhy.

Pokud je ze synchronní akce spuštěna měřicí úloha s příkazem MEAWA, jsou změřené hodnoty k dispozici pouze v souřadném systému stroje.

Geometrické osy / transformace

Jestliže má být měření pomocí specifické osy spuštěno pro geometrickou osu, musí být stejná měřicí úloha explicitně naprogramována také pro všechny zbývající geometrické osy. Totéž platí i pro osy, které se podílejí na transformaci.

Příklady:

```
N10 MEASA[Z]=(1,1) MEASA[Y]=(1,1) MEASA[X]=(1,1) G0 Z100
```

nebo
```
N10 MEASA[Z]=(1,1) POS[Z]=100
```

Zjištění stavu

Jestliže je v programu požadováno vyhodnocení, zda byla měřicí sonda vychýlena, resp. zda sepnula, pak je možné její stav zjistit prostřednictvím následujících systémových proměnných:

Systémová proměnná	Význam	Datový typ	Hodnota	
$A_PROBE[<n>]	Stav měřicí sondy	INT	0	Měřicí sonda není vychýlena.
			1	Měřicí sonda je vychýlena.
$AC_MEA[<n>]	Stav sepnutí měřicí sondy Na začátku měření se parametr $AC_MEA[<n>] automaticky vynuluje.	INT	0	Měřicí sonda nesepnula.
			1	Měřicí sonda sepnula (všechny spouštěcí události v naprogramovaném bloku měření se uskutečnily).

<n> = číslo měřicí sondy

Poznámka

Pokud je měření spuštěno ze synchronní akce, proměnná $AC_MEA se neaktualizuje. V tomto případě je potřeba zjistit stav signálů rozhraní NC/PLC DB31, ... DBX62.3, příp. proměnné $AA_MEAACT[<osa>], která má stejný význam.

$AA_MEAACT==1: Měření je aktivní

$AA_MEAACT==0: Měření není aktivní

Omezení měřicí sondy

Prostřednictvím systémové proměnné $A_PROBE_LIMITED je možno v NC programu nebo v synchronní akci zjistit stav omezení měřicí sondy, když se používá telegram PROFIBUS 395:

$A_PROBE_LIMITED[<n>] == 0: Omezení měřicí sondy není aktivní/je resetováno

$A_PROBE_LIMITED[<n>] == 1: Omezení měřicí sondy je aktivní

<n> = číslo měřicí sondy

Výsledky měření pro příkazy MEASA / MEAWA

Výsledky měření nasnímané měřicí sondou mohou být načítány ve výrobním programu a v synchronních akcích prostřednictvím následujících systémových proměnných:

Systémová proměnná	Význam
$AA_MM1[<osa>]	Změřená hodnota nasnímaná měřicí sondou při spouštěcí události 1 v MCS
...	...
$AA_MM4[<osa>]	Změřená hodnota nasnímaná měřicí sondou při spouštěcí události 4 v MCS
$AA_MW1[<osa>]	Změřená hodnota nasnímaná měřicí sondou při spouštěcí události 1 ve WCS
...	...
$AA_MW4[<osa>]	Změřená hodnota nasnímaná měřicí sondou při spouštěcí události 4 ve WCS

<osa> = měřicí osa

Jestliže se měřicí úloha uskutečňuje se dvěma měřicími systémy, zaznamenává se každá s obou možných spouštěcích událostí z obou měřicích systémů. Obsazení systémových proměnných je potom následující:

$AA_MM1[<osa>]	nebo	$AA_MW1[<osa>]	Změřená hodnota z měřicího systému 1 při spouštěcí události 1
$AA_MM2[<osa>]	nebo	$AA_MW2[<osa>]	změřená hodnota z měřicího systému 2 při spouštěcí události 1
$AA_MM3[<osa>]	nebo	$AA_MW3[<osa>]	Změřená hodnota z měřicího systému 1 při spouštěcí události 2
$AA_MM4[<osa>]	nebo	$AA_MW4[<osa>]	Změřená hodnota z měřicího systému 2 při spouštěcí události 2

Kontinuální měření (MEAC)

V případě funkce MEAC jsou změřené hodnoty v souřadném systému stroje a ukládají se do zadané paměti typu FIFO[<n>] (zásobníková paměť). Jestliže je nastaveno, že měření má probíhat se dvěma měřicími sondami, jsou hodnoty změřené druhou sondou ukládány odděleně do pro tento účel vytvořené paměti FIFO[<n>+1] (nastavitelné pomocí strojního parametru).

Paměť FIFO je zásobníkovou pamětí, do které jsou změřené hodnoty ukládány v proměnné $AC_FIFO na principu "první dovnitř - první ven".

Poznámka

Obsah paměti FIFO je možné ze zásobníkové paměti přečíst jen jednou. Pokud potřebujete změřené údaje používat vícekrát, musíte je přechodně uložit do uživatelské paměti.

Pokud počet změřených hodnot pro paměť FIFO překročí maximální možný počet definovaný ve strojním parametru, měření se automaticky ukončí.

Nekonečné měření je možné realizovat pomocí cyklického načítání změřených hodnot. Načítání se přitom musí uskutečňovat s minimálně stejnou četností, s jakou probíhá přísun nových změřených hodnot.

Další informace: Příručka k funkcím, Synchronní akce

Ochrana před chybami v programu

Jsou rozpoznávány následující chyby v programování a při jejich výskytu se vypíše chyba:

- Příkaz MEASA/MEAWA je naprogramován v jednom bloku spolu s příkazem MEAS/MEAW
  Příklad:
  ```
  N01 MEAS=1 MEASA[X]=(1,1) G01 F100 POS[X]=100
  ```
- MEASA/MEAWA s počtem parametrů < 2 nebo > 5
  Příklad:
  ```
  N01 MEAWA[X]=(1) G01 F100 POS[X]=100
  ```
- MEASA/MEAWA se spouštěcí událostí nerovnající se 1/ -1/ 2/ -2
  Příklad:
  ```
  N01 MEASA[B]=(1,1,3) B100
  ```
- MEASA/MEAWA s nesprávným režimem
  Příklad:
  ```
  N01 MEAWA[B]=(4,1) B100
  ```
- MEASA/MEAWA s dvakrát naprogramovanou spouštěcí událostí
  Příklad:
  ```
  N01 MEASA[B]=(1,1,-1,2,-1) B100
  ```
- MEASA/MEAWA a chybějící geometrická osa
  Příklad:
  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) G01 X50 Y50 Z50 F100 ;GEO-Achse X/Y/Z
  ```
- Měřicí úloha u geometrických os bez jednotek
  Příklad:
  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) MEASA[Z]=(1,1,2) G01 X50 Y50 Z50 F100
  ```

Viz také

Synchronní akce (Strana 964)

4.7.9 Speciální funkce pro uživatele OEM (OMA1 ... OMA5, OEMIPO1, OEMIPO2, G810 ... G829)

Adresy OEM

Význam adres OEM určuje uživatel OEM. Funkce se instalují prostřednictvím cyklů překladače. Je rezervováno 5 adres OEM (`OMA1` ... `OMA5`). Identifikátory adres jsou nastavitelné. Adresy OEM jsou přípustné v každém bloku.

Rezervovaná volání příkazů G-funkcí

Pro uživatele OEM jsou rezervována následující volání příkazů G-funkcí:

- `OEMIPO1`, `OEMIPO2` (ze skupiny G-funkcí č. 1)
- `G810` ... `G819` (skupina G-funkcí č. 31)
- `G820` ... `G829` (skupina G-funkcí č. 32)

Funkce se instalují prostřednictvím cyklů překladače.

Funkce a podprogramy

Kromě toho mohou uživatelé OEM zakládat také předdefinované funkce a podprogramy s předáváním parametrů.

Poznámka

Simulace obrobku

Až do verze SW 4.4 nebyly při simulaci obrobku podporovány žádné cykly překladače (CC), od verze SW 4.4 jsou podporovány jen vybrané.

Příkazy jazyka ve výrobním programu, které spadají do nepodporovaných cyklů překladače (`OMA1` ... `OMA5`, `OEMIPO1/2`, `G810` ... `G829`, některé procedury a funkce), pokud nejsou individuálně ošetřeny, mají proto za následek alarmové hlášení a přerušení simulace.

Řešení: Pro chybějící specifické prvky jazyka CC ve výrobním programu zajistěte individuální zpracování (zjišťování stavu $P_SIM).
Příklad:

```
N1 G01 X200 F500
IF (1==$P_SIM)
N5 X300 ;při simulaci CC nejsou aktivní
ELSE
N5 X300 OMA1=10
ENDIF
```

4.7.10 Snížení posuvu se zpožděním v rozích (FENDNORM, G62, G621)

V případě automatického zpožďování v rozích se hodnota posuvu krátce před dosažením příslušného rohu snižuje podle křivky zvonovitého tvaru. Kromě toho může být pomocí nastavovaných parametrů definován přídavek rozměru pro odpovídající chování nástroje v rohu:

- Začátek a konec snížené hodnoty posuvu
- Korekce, o kterou je posuv snížen
- Rozpoznávání odpovídajícího rohu

Rohy, kterých se toto snížení posuvu týká, budou ty, jejichž vnitřní úhel je menší než úhel stanovený příslušným nastavovaným parametrem.

Prostřednictvím předdefinované hodnoty FENDNORM se funkce automatické korekce v rozích vypíná.

Syntaxe

```
FENDNORM
G62 G41
G621
```

Význam

FENDNORM:	Deaktivování automatického zpoždění v rozích
G62:	Zpoždění na **vnitřních rozích** při aktivní korekci rádiusu nástroje
G621:	Zpoždění na **všech rozích** při aktivní korekci rádiusu nástroje

Příkaz G62 je v platnosti jen na vnitřních rozích, když je aktivní korekce rádiusu nástroje G41/G42 a aktivní režim řízení pohybu po dráze G64/G641.

Na odpovídající roh se najíždí se sníženou hodnotou posuvu, která se vypočítává následujícím způsobem:

F * (korekce pro snížení hodnoty posuvu) * korekce snížení hodnoty posuvu

Maximálního možného snížení rychlosti posuvu je dosaženo přesně tehdy, když nástroj, vztaženo na dráhu jeho středu, dosáhne bodu, v němž se má uskutečnit změna směru na příslušném rohu.

G621 funguje analogicky k příkazu G62 na každém rohu, na který osy stanovené příkazem FGROUP narazí.

4.7.11 Programovatelné kritérium konce pohybu (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)

Podobně jako kritérium přechodu na další blok v případě dráhové interpolace (G601, G602 a G603) může být ve výrobním programu, příp. v synchronních akcích pro příkazové/PLC osy naprogramováno u interpolace jednotlivých os kritérium konce pohybu

V závislosti na tom, které kritérium konce pohybu je nastaveno, jsou bloky výrobního programu, příp. bloky technologických cyklů s pohyby jednotlivých os ukončeny různě rychle. Totéž platí pro PLC prostřednictvím FC15/16/18.

Syntaxe

```
FINEA[<osa>]
COARSEA[<osa>]
IPOENDA[<osa>]
IPOBRKA(<osa>[,<časový okamžik>])
ADISPOSA(<osa>[,<režim>,<velikost okna>])
```

Význam

FINEA:	Kritérium konce pohybu: "Jemné přesné najetí"		
	Platnost:	modální	
COARSEA:	Kritérium konce pohybu: "Hrubé přesné najetí"		
	Platnost:	modální	
IPOENDA:	Pohybové kritérium: "zastavení interpolátoru"		
	Platnost:	modální	
IPOBRKA:	Kritérium přechodu na další blok: Brzdná charakteristika		
	Platnost:	modální	
ADISPOSA:	Toleranční okno pro kritérium konce pohybu		
	Platnost:	modální	
<osa>:	Název kanálové osy (X, Y, ...)		
<časový okamžik>:	Časový okamžik přechodu na další blok, vztaženo ke hraně brzdné charakteristiky v %: • 100% = začátek brzdné charakteristiky • 0% = Konec brzdné charakteristiky, to má stejný význam jako příkaz IPOENDA		
	Typ:	REAL	
<režim>:	Vztah tolerančního okna		
	Rozsah hodnot:	0	Toleranční okno není aktivní
		1	Toleranční okno vztaženo na požadovanou polohu
		2	Toleranční okno vztaženo na skutečnou polohu
	Typ:	INT	
<velikost okna>:	Velikost tolerančního okna		
	Typ:	REAL	

Příklady

Příklad 1: Kritérium konce pohybu: "Zastavení interpolátoru"

Programový kód

```
; Najíždění polohovací osou X na pozici 100, rychlost
200 m/min, zrychlení 90%,
; Kritérium konce pohybu: Zastavení interpolátoru
N110 G01 POS[X]=100 FA[X]=200 ACC[X]=90 IPOENDA[X]


; synchronní akce:
; VŽDY KDYŽ: vstup 1 je nastaven
;POTOM najíždění polohovací osou X na pozici 50, ry-
chlost 200 m/min, zrychlení 140%,
;      Kritérium konce pohybu: Zastavení interpolátoru
N120 EVERY $A_IN[1] DO POS[X]=50 FA[X]=200 ACC[X]=140
IPOENDA[X]
```

Příklad 2: Kritérium přechodu na další blok: "Brzdná charakteristika"

Programový kód	Komentář
	; předdefinovaná nastavení jsou platná
N40 POS[X]=100	; polohovací pohyb osou X na pozici 100
	Kritérium přechodu na další blok: Jemné přesné najetí
N20 IPOBRKA(X,100)	; kritérium přechodu na další blok: "Brzdná charakteristika",
	100% = začátek brzdné charakteristiky
N30 POS[X]=200	; přechod na další blok se uskuteční, jakmile osa X začne brzdit
N40 POS[X]=250	; Osa X už nebrzdí na pozici 200, nýbrž jede dál na pozici 250.
	Jakmile osa začne brzdit, následuje přechod na další blok
N50 POS[X]=0	; Osa X brzdí a pak jede zpět na pozici 0.
	Přechod na další blok se uskutečňuje na pozici 0 a při "jemném přesném najetí".
N60 X10 F100	; Osa X najíždí jako dráhová osa na pozici 10.

Další informace

Systémové proměnné pro kritérium konce pohybu

Kritérium konce pohybu, které je momentálně v platnosti, je možné načíst pomocí systémové proměnné $AA_MOTEND.

Kritérium přechodu na další blok: "Brzdná charakteristika" (IPOBRKA)

Jestliže je při aktivování kritéria přechodu na další blok "Brzdná charakteristika" pro volitelný bod pro přechod na další blok naprogramována hodnota, bude tato hodnota v platnosti pro následující polohovací pohyb a synchronně se zpracováním hlavní větve se bude zapisovat do strojního parametru. Pokud pro okamžik přechodu na další blok není udána žádná hodnota, použije se aktuální hodnota z nastavovaného parametru.

SD43600 $SA_IPOBRAKE_BLOCK_EXCHANGE

Při následujícím naprogramování kritéria konce pohybu osy (FINEA, COARSEA , IPOENDA) se funkce IPOBRKA pro odpovídající osu deaktivuje.

Doplňkové kritérium přechodu na další blok: "Toleranční okno" (ADISPOSA)

Pomocí příkazu ADISPOSA může být jako doplňkové kritérium pro přechod na další blok definováno toleranční okno okolo koncového bodu bloku (lze zvolit, zda je to skutečná poloha nebo požadovaná poloha). Pro přechod na další blok musí být splněny obě podmínky:

- Kritérium přechodu na další blok: "Brzdná charakteristika"
- Kritérium přechodu na další blok: "Toleranční okno"

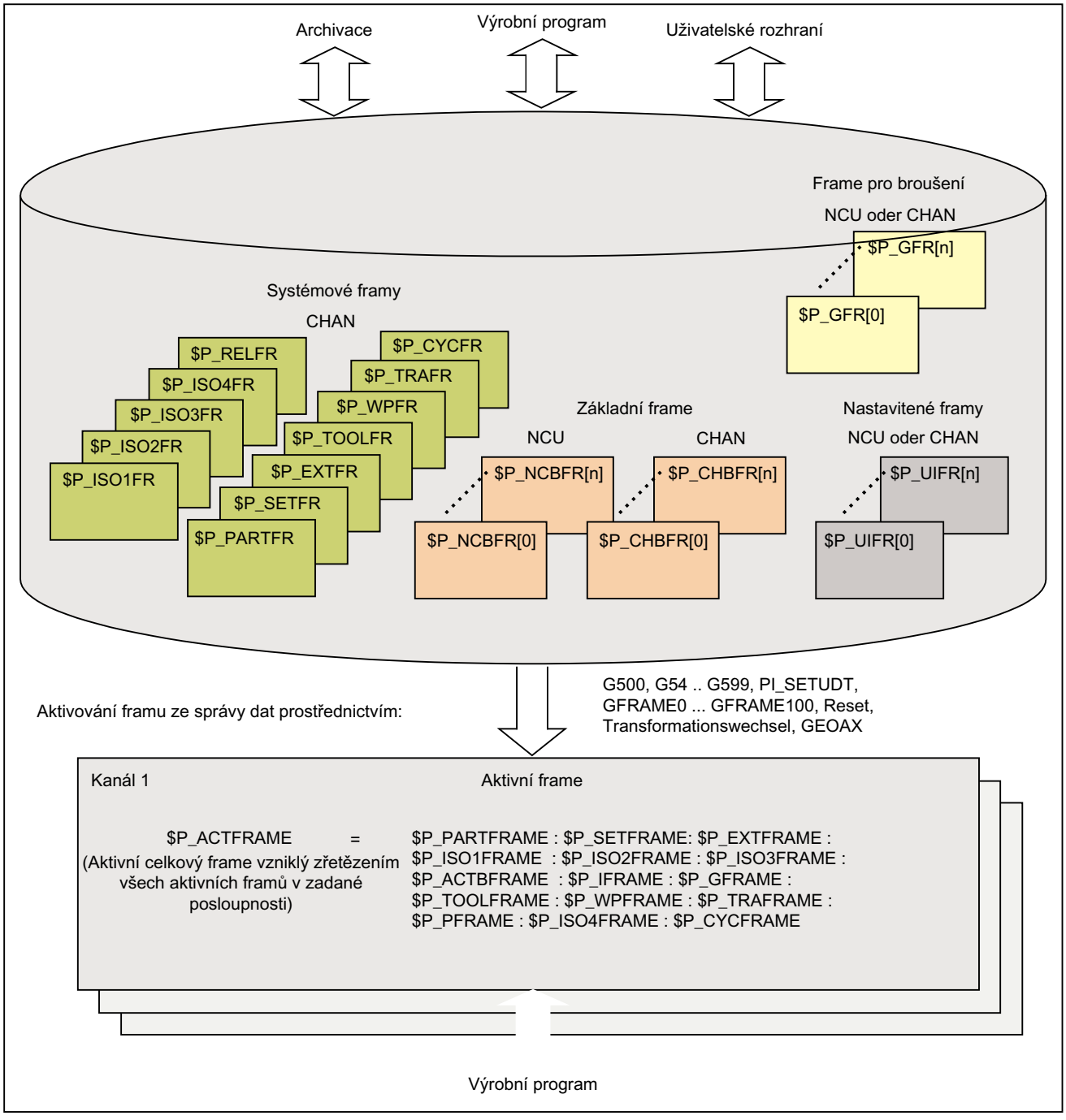
4.8 Transformace souřadného systému (Frame)

4.8.1 Transformace souřadnic pomocí proměnné typu frame

Kromě příkazů framů (Strana 309), jako jsou např. příkazy ROT, AROT, SCALE atd., může být souřadný systém obrobku (WCS) transformován také prostřednictvím proměnných typu frame $P_...FR (framy správy dat) a $P_...FRAME (aktivní framy).

Následující obrázek poskytuje přehled o strukturování proměnných typu frame:

- Frame správy dat
- Aktivní frame
- Aktivní celkový frame: Zřetězení Všech aktivních framů
- Globální framy v NCU
- Kanálové framy



Obrázek 4-1 Přehled proměnných typu frame

4.8.1.1 Předdefinované proměnné typu frame ($P_CHBFRAME, $P_IFRAME, $P_PFRAME, $P_ACTFRAME)

Aktivní: kanálové základní framy $P_CHBFRAME[<n>] ($P_BFRAME)

Poznámka

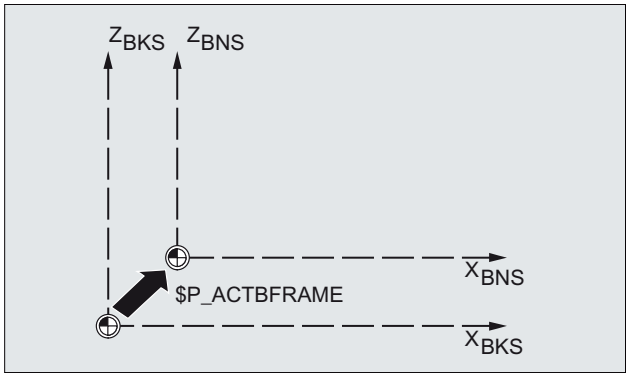
Aktuální základní frame `$P_BFRAME` a základní frame správy dat `$P_UBFR` jsou z důvodů kompatibility zachovány.

- `$P_BFRAME` ≙ `$P_CHBFRAME[0]`
- `$P_UBFR` ≙ `$P_CHBFR[0]`.

Proměnné framu $P_CHBFRAME[<n>] definují vztah mezi základním souřadným systémem (BCS) a základním souřadným systémem počátku (nuly) (BNS).

Pokud má být aktuální kanálový základní frame $P_CHBFRAME[`<n>`] v NC programu okamžitě v platnosti, máte k dispozici následující možnosti.

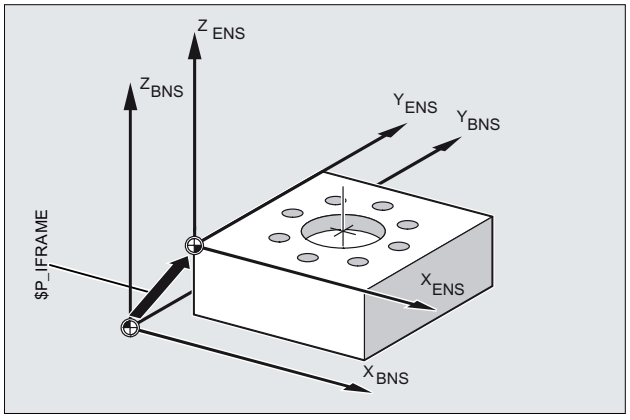
- Příkazy:
  - `G500` (deaktivování všech nastavitelných framů, základní frame zůstává aktivní)
  - `G54` ... `G599` (nastavitelná posunutí počátku)
- Přiřazení základního framu konkrétního kanálu ve správě dat určitému aktuálnímu kanálovému základnímu framu:
  `$P_CHBFRAME[<n>] = $P_CHBFR[<m>]`



Aktivní: kanálový nastavitelný frame $P_IFRAME

Framová proměnná $P_IFRAME, která definuje vztah mezi základním souřadným systémem počátku (nuly) (BNS) a nastavitelným souřadným systémem (ENS).

- `$P_IFRAME` odpovídá `$P_UIFR[$P_IFRNUM]`
- Proměnná `$P_IFRAME` obsahuje po naprogramování např. `G54` posunutí, otočení, změny měřítka a zrcadlová převrácení definovaná příkazem `G54`.

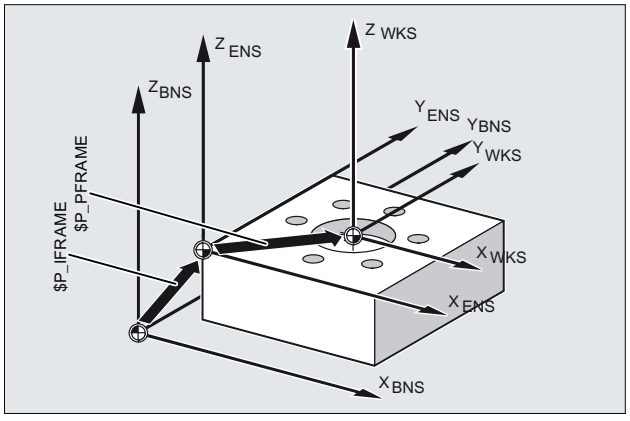


Aktivní: kanálový programovatelný frame $P_PFRAME

Framová proměnná $P_PFRAME, která definuje vztah mezi nastavitelným souřadným systémem (ENS) a souřadným systémem obrobku (WCS).

Proměnná `$P_PFRAME` obsahuje výsledný frame, který vyplývá

- **z naprogramovaných příkazů** `TRANS/ATRANS`, `ROT/AROT`, `SCALE/ASCALE`, `MIRROR/AMIRROR`, příp.
- **z přiřazení** hodnot pomocí příkazů `CTRANS`, `CROT`, `CMIRROR`, `CSCALE` do programovatelného framu

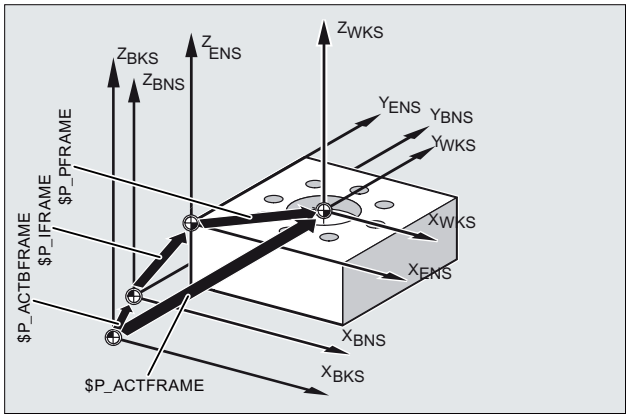


Aktivní: Celkový frame $P_ACTFRAME

Celkový frame, který je v platnosti v daném kanálu, vyplývá ze zřetězení všech framů, které jsou v daném kanálu v platnosti.

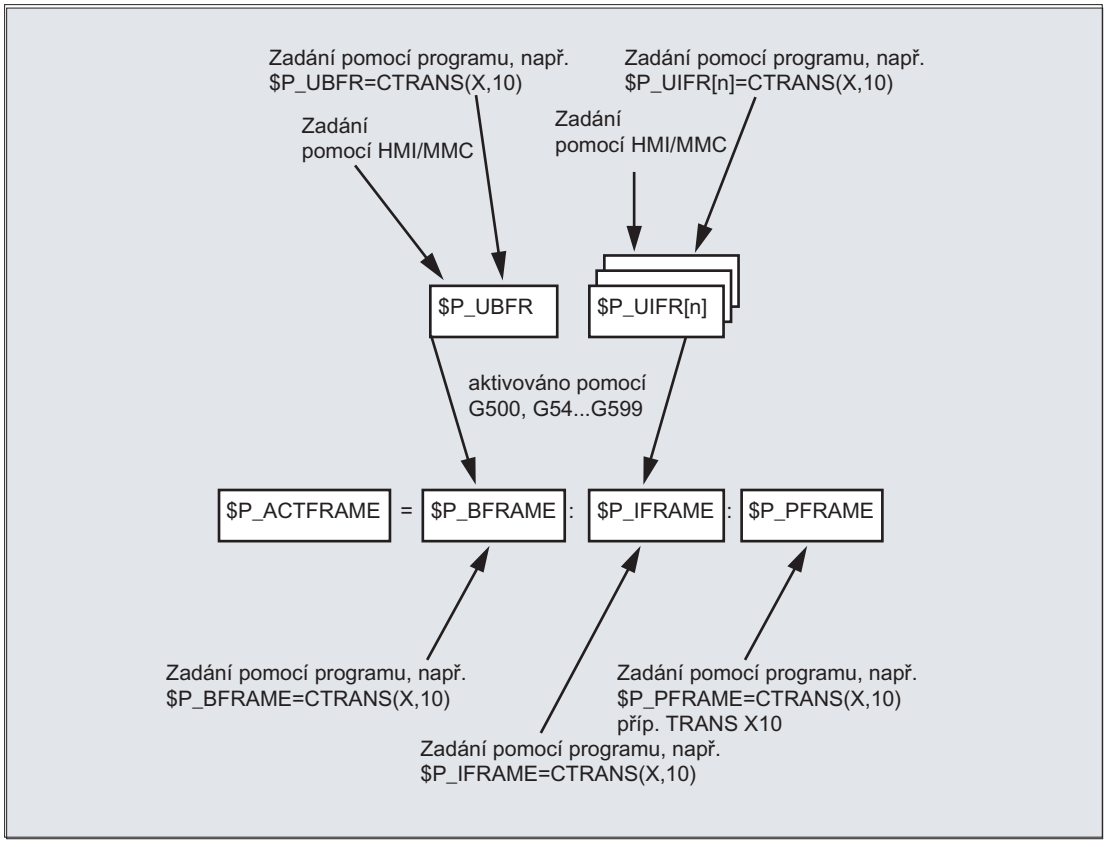
```
$P_ACTFRAME =    $P_PARTFRAME : $P_SETFRAME  : $P_EXTFRAME :
                 $P_ISO1FRAME : $P_ISO2FRAME : $P_ISO3FRAME :
                 $P_ACTBFRAME : $P_IFRAME    : $P_GFRAME :
                 $P_TOOLFRAME : $P_WPFRAME   : $P_TRAFRAME :
                 $P_PFRAME    : $P_ISO4FRAME : $P_CYCFRAME
```

$P_ACTFRAME popisuje počátek (nulu) aktuálního souřadného systému obrobku.



Obrázek 4-2 Proměnná typu FRAME - $P_ACTFRAME

Jestliže se změní jeden z následujících framů $P_BFRAME / $P_CHBFRAME[<n>], $P_IFRAME oder $P_PFRAME, aktuální celkový frame $P_ACTFRAME se nově vypočítá.



Základní frame a nastavitelný frame zůstávají v platnosti i po resetu, pokud je strojní parametr MD 20110 RESET_MODE_MASK nastaven následujícím způsobem:

Bit0=1, Bit14=1 --> $P_UBFR (základní frame) je v platnosti

Bit0=1, Bit5=1 --> $P_UIFR[$P_UIFRNUM] (nastavitelný frame) je v platnosti

Správa dat: Kanálový základní frame $P_CHBFR[<n>]

Prostřednictvím framových proměnných $P_CHBFR[<n>] je možno číst základní framy ve správě dat a zapisovat do nich. Když jsou do framu ve správě dat zapsána nějaká data, neaktivují se v kanálu okamžitě. Aktivování framu, do něhož byla zapsána data, se uskuteční, když:

- se provede reset kanálu a MD20110 $MC_RESET_MODE_MASK, bit0 == 1 a bit14 == 1
- Příkaz G500, G54 ... G57, G505 ... G599 (aktivování/deaktivování základního framu s následným novým výpočtem aktuálního celkového framu)

Správa dat: Kanálový nastavitelný frame $P_UIFR[<n>]

Prostřednictvím framových proměnných $P_UIFR[<n>] je možno číst nastavitelné framy ve správě dat a zapisovat do nich. Když jsou do framu ve správě dat zapsána nějaká data, neaktivují se v kanálu okamžitě. Započítání framu, do něhož byla v kanálu zapsána data, se uskuteční, když:

- Příkaz G500 (deaktivování všech nastavitelných framů, příp. posunutí počátku)
- Příkaz G54 ... G57, G505 ... G599 (aktivování nastavitelného framu, příp. posunutí počátku)

Aktivní nastavitelný frame	Frame ve správě dat	(odpovídá příkazu)
$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

4.8.2 Přiřazování hodnoty framu

4.8.2.1 Přímé přiřazení hodnoty (hodnota osy, úhlu, měřítko)

V NC programu můžete dosazovat hodnoty přímo do framů nebo do proměnných typu FRAME.

Syntaxe

Syntaxe

```
$P_PFRAME = CTRANS(X, <hodnota posunutí>, Y, <hodnota posunutí>, Z, <hodnota posunutí>, ...)
$P_PFRAME = ROT(X, <úhel>, Y, <úhel>, Z, <úhel>, ...)
$P_UIFR[..] = CROT(X, <úhel>, Y, <úhel>, Z, <úhel>, ...
```

```
$P_PFRAME = CSCALE(X, <Měřítko>, Y, <Měřítko>, Z, <Měřítko>, ...)
$P_PFRAME = CMIRROR(X, Y, Z)
```

Syntaxe pro příkaz `$P_CHBFRAME[<n>]` je identická jako u příkazu `$P_PFRAME`.

Význam

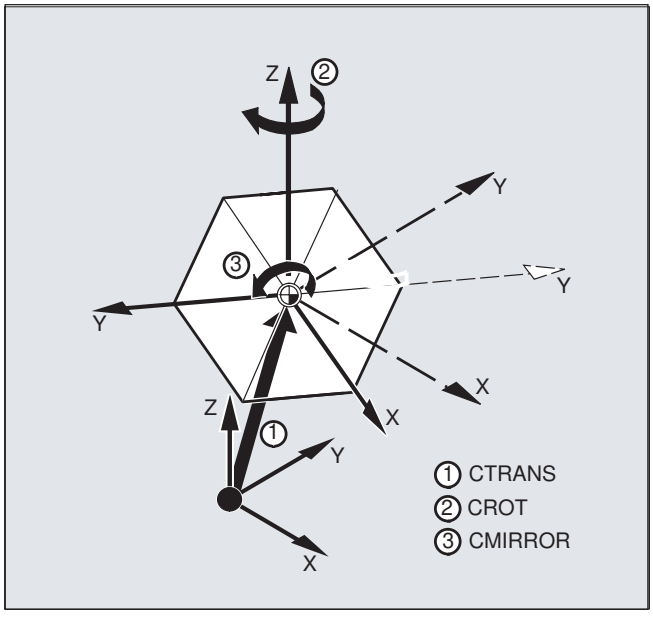
`CTRANS`:	Posunutí v uvedených osách
`CROT`:	Otočení okolo uvedených os
`CSCALE`:	Změna měřítka v uvedených osách
`CMIRROR`:	Zrcadlové převrácení pro uvedenou osu
`X, Y, Z`:	Hodnota posunutí ve směru uvedené geometrické osy
`<hodnota posunutí>`:	Hodnota posunutí
`<úhel>`:	Úhel, o který se provádí otočení
`<Měřítko>`:	Údaj měřítka

Příklady

Přiřazení hodnot složkám aktuálního programovatelného framu

Přiřazení hodnot složkám posunutí, otočení, změny měřítka a zrcadlového převrácení aktuálního programovatelného framu:

```
$P_PFRAME = CTRANS(X,10,Y,20,Z,5) : CROT(Z,45) : CMIRROR(Y)
```



Zápis do složek otočení framu

Přiřazení hodnot pro všechny tři osy složkám otočení nastavitelného framu ve správě dat `$P_UIFR` pomocí příkazu `CROT`:

```
$P_UIFR[5] = CROT(X, 0, Y, 0, Z, 0)
```

Další alternativou je přímé přiřazení jednotlivých hodnot pro příslušnou osu přímo složce otočení ve framu ve správě dat:

```
$P_UIFR[5, Y, RT]=0
$P_UIFR[5, X, RT]=0
$P_UIFR[5, Z, RT]=0
```

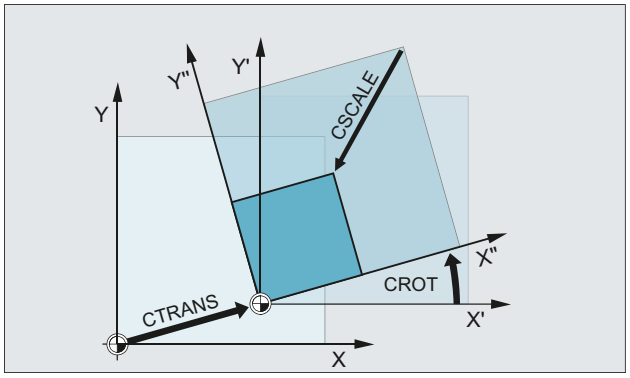
Popis

Větší počet operací na jednom framu je možno zapsat pomocí operátoru zřetězení : a tak je spojit. Operace se potom uskuteční postupně zleva doprava.

Příklad

Zřetězené operace ve framu $P_PFRAME s posunutím, otočením a změnou měřítka:

```
$P_PFRAME = CTRANS(...) : CROT(...) : CSCALE...
```



4.8.2.2 Načítání a editace složek framu (TR, FI, RT, SC, MI)

Pokud potřebujete, můžete mít přístup k **jednotlivým** datům framu, např. k hodnotám pro určité posunutí nebo úhel otočení. Tyto hodnoty můžete změnit nebo je můžete přiřadit jiné proměnné.

Syntaxe

`R10=$P_UIFR[$P_UIFNUM,X,RT]`	Úhel otočení RT okolo osy X z momentálně platného nastavitelného posunutí počátku $P_UIFRNUM má být přiřazen proměnné R10.
`R12=$P_UIFR[25,Z,TR]`	Hodnota posunutí TR ve směru osy Z z datového bloku nastavitelného framu č. 25 má být přiřazena proměnné R12.

`R15=$P_PFRAME[Y,TR]`	Hodnota posunutí TR ve směru osy Z z aktuálního programovatelného framu má být přiřazena proměnné R15.
`$P_PFRAME[X,TR] = 25`	Hodnota posunutí TR ve směru osy X z aktuálního programovatelného framu má být změněna. Od tohoto okamžiku platí X25.

Význam

`$P_UIFRNUM`:	`Pomocí této proměnné se automaticky vytváří vztah k momentálně platnému nastavitelnému posunutí počátku.`
`P_UIFR[n,…,…]`:	`Zadáním čísla framu n se otevírá přístup k nastavitelnému framu č. n.`
	`Zadání složky, která má být načtena nebo změněna.`
`TR`:	`TR - posunutí`
`FI`:	`FI - jemné posunutí`
`RT`:	`RT - otočení`
`SC`:	`SC - změna měřítka`
`MI`:	`MI - zrcadlové převrácení`
`X, Y, Z`:	`Kromě toho (viz příklady) je zapotřebí zadat odpovídající osy X, Y, Z.`

Rozsah hodnot pro otočení RT

Otočení okolo 1. geometrické osy: -180° až +180°
Otočení okolo 2. geometrické osy: -90° až +90°
Otočení okolo 3. geometrické osy: -180° až +180°

Popis

Vyvolání framu

Prostřednictvím systémové proměnné $P_UIFRNUM můžete mít přístup přímo k momentálně nastavenému posunutí počátku, které bylo zadáno pomocí příkazů $P_UIFR příp. G54, G55, ... ($P_UIFRNUM obsahuje číslo momentálně nastaveného framu).

Všechny ostatní uložené nastavitelné framy $P_UIFR vyvoláte prostřednictvím zadání odpovídajícího čísla $P_UIFR[n].

Pro předem definované framy a vlastní definované framy zadejte příslušné názvy, např. $P_IFRAME.

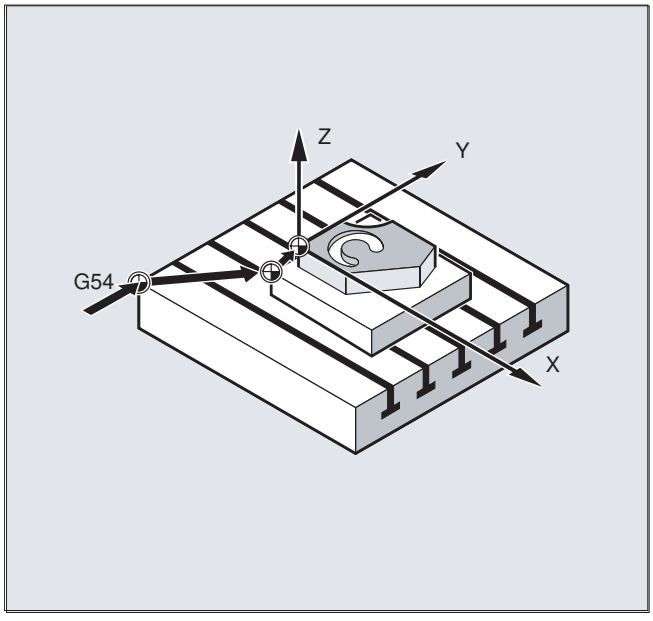
Vyvolávání dat

V hranatých závorkách jsou uvedeny hodnoty názvu osy a složky framu, kterým chcete mít přístup nebo které si přejete změnit, např. [X, RT] nebo [Z, MI].

4.8.2.3 Výpočety s framy

V NC programu může být frame přiřazen jinému framu nebo framy mohou být vzájemně zřetězovány.

Zřetězení framů je užitečné např. pro popis většího počtu obrobků, které jsou uspořádány na paletě a které mají být opracovány v rámci jedné výrobní operace.



Za účelem popisu obráběcích úloh na paletách by mohly složky framu obsahovat např. pouze určité dílčí hodnoty, jejichž zřetězením se vygenerují různé počátky souřadného systému obrobku.

Příklady

Přiřazení

Programový kód	Komentář
`DEF FRAME EINSTELLUNG_1`	`; definice lokální proměnné typu frame`
`EINSTELLUNG_1 = CTRANS(X,10)`	`; přiřazení výsledku funkce provedené s proměnnou typu frame`
`$P_PFRAME = EINSTELLUNG_1`	`; přiřazení framové proměnné aktuálnímu framu`
`DEF FRAME EINSTELLUNG_4`	`; definice lokální proměnné typu frame`
`EINSTELLUNG_4 = $P_PFRAME`	`; uložení aktuálního framu do proměnné typu frame`
`...`	
`$P_PFRAME = EINSTELLUNG_4`	`; zpětné načtení aktuálního framu z proměnné typu frame`

Zřetězení

Framy mohou být prostřednictvím operátoru : v naprogramované posloupnosti spolu zřetězeny. Složky framů, jako např. posunutí, otočení atd., jsou uskutečňovány aditivně jedna po druhé.

Programový kód	Komentář

`$P_IFRAME = $P_UIFR[15] : $P_UIFR[16]`	`; přiřazení výsledného framu ze zřetězení`
	`; obou nastavitelných framů ze správy dat`
	`; do nastavitelného celkového framu`
	`; příklad použití:`
	`; $P_UIFR[15]: Posunutí`
	`; $P_UIFR[16]: Otočení`

`$P_UIFR[3] = $P_UIFR[4] : $P_UIFR[5]`	`; přiřazení výsledného framu ze zřetězení`
	`; obou nastavitelných framů ze správy dat`
	`; do jiného nastavitelného framu ve správě dat`

4.8.2.4 Definice proměnných typu frame (DEF FRAME)

Kromě předem definovaných proměnných typu frame můžete definovat také své vlastní framové proměnné. Framové proměnné, které jste sami definovali, jsou uživatelskými proměnnými typu FRAME. Název framu může být libovolný, musí být jen dodržena pravidla pro uživatelské proměnné.

Framovým proměnným, které jste sami definovali, můžete dosazovat hodnoty pomocí funkcí CTRANS, CROT, CSCALE, CMIRROR.

Syntaxe

```
DEF FRAME <název>
```

Význam

`DEF FRAME`:	Definice uživatelské proměnné typu FRAME.
<název>:	Název framové proměnné

Příklad

Definice proměnné typu FRAME s názvem „PALETTE“ a přiřazení hodnot posunutí a otočení:

Programový kód	Komentář
`DEF FRAME PALETTE`	`; definice framové proměnné PALETTE`
`PALETTE = CTRANS(...) : CROT(...)`	`; přiřazení výsledného framu ze zřetězení` `; posunutí a otočení framové proměnné PALETTE`

4.8.3 Hrubé a jemné posunutí (CTRANS, CFINE)

Jemné posunutí

Jemné posunutí `CFINE(...)` se může použít v následujících framech:

- Nastavitelné framy: $P_UIFR, příp. $P_IFRAME
- Základní framy: $P_NCBFR[<n>], $P_CHBFR[<n>], příp. $P_CHBFRAMES[<n>] nebo $P_ACTBFRAME
- Programovatelný frame: $P_PFRAME

Pro programování jemného posunutí framu se používá příkaz `CFINE(...)`.

Hrubé posunutí

Hrubé posunutí `CTRANS(...)` se může používat ve všech framech.

Celkové posunutí

Celkové posunutí se skládá z hrubého a z jemného posunutí.

Strojní parametry

Odblokování jemného posunutí

Jemné posunutí je možno odblokovat pomocí strojního parametru:

MD18600 $MN_MM_FRAME_FINE_TRANS = 1

Syntaxe

Jemné posunutí

- Celkový frame
  - `<Frame> = CFINE(<K_1>,<hodnota>)`
  - `<Frame> = CFINE(<K_1>,<hodnota>, <K_2>, <hodnota>)`
  - `<Frame> = CFINE(<K_1>,<hodnota>, <K_2>, <hodnota>, <K_3>, <hodnota>)`
- Složky framu
  - `<Frame>[<n>, <K_1>, FI] = <hodnota>`

Hrubé posunutí

- Celkový frame
  - `<Frame> = CTRANS(<K_1>,<hodnota>)`
  - `<Frame> = CTRANS(<K_1>,<hodnota>, <K_2,<hodnota>)`
  - `<Frame> = CTRANS(<K_1>,<hodnota>, <K_2,<hodnota>, <K_3,<hodnota>)`
- Složky framu
  - `<Frame>[<n>,<K_1>,TR] = <hodnota>`

Speciálně pro programovatelný frame `$P_PFRAME`:

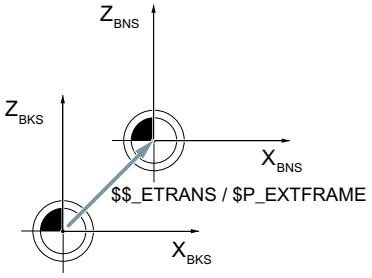
- `TRANS <K_1> <hodnota>`
- `TRANS <K_1> <hodnota><K_2> <hodnota>`
- `TRANS <K_1> <hodnota><K_2> <hodnota><K_3> <hodnota>`

Význam

`<Frame>`:	Frame, např. Nastavitelný frame ze správy dat `$P_UIFR[<n>]`
`CFINE`:	Jemné posunutí, aditivní posunutí.
`CTRANS`:	Hrubé posunutí, absolutní posunutí
`TRANS`:	Jen programovatelný frame: Hrubé posunutí, absolutní posunutí
`<K_n>`:	Osy souřadného systému X, Y, Z
`<hodnota>`:	Hodnota posunutí

4.8.4 Externí posunutí počátku ($AA_ETRANS)

Externí posunutí počátku je lineární posunutí mezi základním souřadným systémem (BCS) a základním systémem počátku (nuly) (BNS).



Externí posunutí počátku pomocí $AA_ETRANS funguje v závislosti na nastavení strojních parametrů dvěma způsoby:

1. Systémová proměnná $AA_ETRANS se uplatňuje po aktivování prostřednictvím signálu rozhraní NC/PLC přímo jako hodnota posunutí
2. Hodnota systémové proměnné $AA_ETRANS se uplatňuje po aktivování prostřednictvím signálu rozhraní NC/PLC v aktivním systémovém framu $P:EXTFRAME a přebírá se do framu $P_EXTFR ve správě dat. Potom se nově vypočítá aktivní celkový frame $P_ACTFRAME.

Strojní parametry

V souvislosti se systémovou proměnnou $AA_ETRANS je potřeba rozlišovat mezi dvěma postupy, které se vybírají prostřednictvím následujícího strojního parametru:

MD28082 $MC_MM_SYSTEM_FRAME_MASK,Bit1 = <hodnota>

<hodnota>	**Význam**
0	Funkce: Přímý zápis $AA_ETRANS[<osa>] prostřednictvím PLC, HMI nebo NC programu. Uvolnění za účelem posunu realizujícího posunutí počátku $AA_ETRANS[<osa>] v nejbližším možném bloku posuvu: DB31, ... DBX3.0
1	Funkce: Aktivování aktivního systémového framu $P:EXTFRAME a framu ve správě dat $P_EXTFR Uvolnění za účelem posunu realizujícího posunutí počátku $AA_ETRANS[<osa>] prostřednictvím: DB31, ... DBX3.0. V kanálu se potom stane následující: • Zastavení všech posuvů v kanálu (kromě příkazových a PLC-os) • Zastavení předběžného zpracování s následnou reorganizací (STOPRE) • Hrubé posunutí aktivního framu $P_EXTFRAME[<osa>] = $AA_ETRANS[<osa>] • Hrubé posunutí framu ve správě dat $P_EXTFR[<osa>] = $AA_ETRANS[<osa>] • Nový výpočet aktivního celkového framu $P_ACTFRAME • Uskutečnění posunutí v naprogramovaných osách. • Pokračování v přerušeném posunu, příp. ve zpracování NC programu

Programování

- Syntaxe
  `$AA_ETRANS[<osa>] = <hodnota>`
- Význam

`$AA_ETRANS`:	Systémová proměnná pro uložení externího posunutí počátku do vyrovnávací paměti
`<osa>`:	Kanálová osa
`<hodnota>`:	Hodnota posunutí

Signál rozhraní NC/PLC

DB31, ... DBX3.0 = 0 → 1 ⇒ $P_EXTFRAME[<osa>] = $P_EXTFR[<osa>] = $AA_ETRANS[<osa>]

4.8.5 Nastavení skutečné hodnoty se ztrátou stavu najetí na referenční bod (PRESETON)

Procedura PRESETON() nastavuje pro jednu nebo více os novou skutečnou hodnotu v souřadném systému stroje (MCS). To odpovídá posunutí počátku (nuly) MCS osy. Osa se v důsledku toho nepohybuje.

Příkazem PRESETON se spouští zastavení předběžného zpracování se synchronizací. Skutečná hodnota je ose přiřazena až v klidovém stavu.

Pokud osa při příkazu PRESETON není tomuto kanálu přiřazena, závisí další postup na specifické konfiguraci při výměně os:

MD30552 $MA_AUTO_GET_TYPE

Stav najíždění na referenční bod:

Prostřednictvím nastavení nové skutečné hodnoty v souřadném systému stroje se uskuteční reset stavu najetí strojní osou na referenční bod:

DB31, ... DBX60.4 / .5 = 0 (najeto na referenční bod / měřicí systém 1 / 2 synchronizován)

Proto se doporučuje používat příkaz PRESETON jen v případě os, u kterých není povinnost najíždět na referenční bod.

Jestliže si přejete opět obnovit původní souřadný systém stroje, je nutné z výrobního programu (G74) příslušnou strojní osou znovu najet na referenční bod, aby se synchronizoval měřicí systém.

⚠ **POZOR**
Ztráta stavu najetí na referenční bod
Prostřednictvím nastavení nové skutečné hodnoty v souřadném systému stroje pomocí příkazu PRESETON se uskuteční reset stavu najetí strojní osou na referenční bod, který pak je „není najeto na referenční bod/není synchronizováno“.

Programování

Syntaxe

```
PRESETON(<osa_1>, <hodnota_1> [, <osa_2>, <hodnota_2>, ... <osa_8>,
<hodnota_8>])
```

Význam

PRESETON:	Nastavení skutečné hodnoty se ztrátou stavu najetí na referenční bod	
	Zastavení předběžného zpracování:	ano
	Samostatný blok:	ano
<osa_x>:	Název osy stroje	
	Typ:	AXIS
	Rozsah hodnot:	identifikátory strojních os definovaných v kanálu

<hodnota_x>:	Nová skutečná hodnota strojní osy v souřadném systému stroje (MCS) Zadání se uskutečňuje v momentálně platném systému jednotek (palce / metrické jednotky). Aktivní programování průměrů (DIAMON) je zohledňováno.	
	Typ:	REAL

Další informace

Příkaz PRESETON v NC-programech

Podrobný popis věnovaný příkazu PRESETON v NC programech naleznete v příručce k funkcím, kapitola Základní funkce.

Příkaz PRESETON v synchronních akcích

Podrobný popis věnovaný příkazu PRESETON v synchronních akcích naleznete v příručce k funkcím, kapitola Synchronní akce.

4.8.6 Nastavení skutečné hodnoty bez ztráty stavu najetí na referenční bod (PRESETONS)

Procedura PRESETONS() nastavuje pro jednu nebo více os novou skutečnou hodnotu v souřadném systému stroje (MCS). To odpovídá posunutí počátku (nuly) MCS osy. Osa se v důsledku toho nepohybuje.

Příkazem PRESETONS se spouští zastavení předběžného zpracování se synchronizací. Skutečná hodnota je ose přiřazena až v klidovém stavu.

Pokud osa při příkazu PRESETONS není tomuto kanálu přiřazena, závisí další postup na specifické konfiguraci při výměně os:

MD30552 $MA_AUTO_GET_TYPE

Stav najíždění na referenční bod:

Prostřednictvím nastavení nové skutečné hodnoty v souřadném systému stroje (MCS) pomocí příkazu PRESETONS se stav najetí strojní osou na referenční bod **nezmění**.

Předpoklady

- **Typ snímače**
  Příkaz PRESETONS je možný pouze v případě následujících typů snímačů aktivního měřicího systému:
  - MD30240 $MA_ENC_TYPE[<Měřicí systém>] = 0 (simulovaný snímač)
  - MD30240 $MA_ENC_TYPE[<Měřicí systém>] = 1 (snímač vysílající surový signál)
- **Režim najíždění na referenční bod**
  Příkaz PRESETONS je možný pouze v případě následujících režimů najíždění na referenční bod aktivního měřicího systému:
  - MD34200 $MA_ENC_REFP_MODE[<měřicí systém>] = 0 (najíždění na referenční bod není možné)
  - MD34200 $MA_ENC_REFP_MODE[<měřicí systém>] = 1 (najíždění na referenční bod inkrementálních, otočných nebo lineárních měřicích systémů: Nulový impulz na výstupu snímače)

Programování

Syntaxe

```
PRESETONS(<osa_1>, <hodnota_1> [, <osa_2>, <hodnota_2>, ... <osa_8>,
<hodnota_8>])
```

Význam

PRESETONS:	Nastavení skutečné hodnoty bez ztráty stavu najetí na referenční bod	
	Zastavení předběžného zpracování:	ano
	Samostatný blok:	ano
<osa_x>:	Název osy stroje	
	Typ:	AXIS
	Rozsah hodnot:	identifikátory strojních os definovaných v kanálu
<hodnota_x>:	Nová aktuální skutečná hodnota strojní osy v souřadném systému stroje (MCS) Zadání se uskutečňuje v aktivním systému jednotek (palce / metrické jednotky). Aktivní programování průměrů (DIAMON) je zohledňováno.	
	Typ:	REAL

Další informace

Příkazy PRESETON v NC-programech

Podrobný popis věnovaný příkazu PRESETONS v NC programech naleznete v příručce k funkcím, kapitola Základní funkce.

Příkazy PRESETONS v synchronních akcích

Podrobný popis věnovaný příkazu PRESETONS v synchronních akcích naleznete v příručce k funkcím, kapitola Synchronní akce.

4.8.7 Výpočet framu na základě 3 změřených bodů v prostoru (MEAFRAME)

Funkce MEAFRAME se používá pro podporu v měřicích cyklech. Vypočítává frame na základě tří ideálních a vzájemně korespondujících změřených bodů.

Když je obrobek nastaven do polohy pro obrábění, je jeho pozice vzhledem ke kartézskému souřadnému systému stroje vůči jeho ideální poloze zpravidla jednak posunutá, ale i pootočená. Pro přesné opracování nebo měření je zapotřebí uskutečnit buď nákladné fyzické nastavování polohy nebo změny pohybů ve výrobním programu.

Frame může být definován nasnímáním trojice bodů v prostoru, jejichž ideální pozice je známa. Pro snímání polohy se používají buď dotykové nebo optické snímače, speciální na nosné destičce přesně fixované díry nebo měřicí kuličky.

Syntaxe

```
MEAFRAME(<ideální body>,<měřené body>,<kvalita>)
```

Význam

MEAFRAME:	Volání funkce		
<ideální body>:	2-rozměrné pole typu REAL, které obsahuje tři souřadnice ideálních bodů.		
<měřené body>:	2-rozměrné pole typu REAL, které obsahuje tři souřadnice měřených bodů.		
<kvalita>:	Proměnná, pomocí které systém vrací informace o kvalitě výpočtu proměnné typu FRAME.		
	Typ:	VAR REAL	
	Hodnota:	-1	Ideální body leží příliš blízko jedné přímky: Frame nemohl být vypočítán. Výsledná proměnná typu FRAME obsahuje neutrální frame.
		-2	Změřené body leží příliš blízko jedné přímky: Frame nemohl být vypočítán. Výsledná proměnná typu FRAME obsahuje neutrální frame.
		-4	Výpočet rotační matice skončil z nějakého jiného důvodu neúspěchem.
		≥ 0.0	Součet deformací (vzdálenosti mezi body), které jsou zapotřebí pro převedení změřeného trojúhelníku na trojúhelník, který se blíží ideálnímu.

Poznámka

Jakost měření

Aby změřené souřadnice bylo možné pomocí kombinace otočení/posunutí přiřadit ideálním souřadnicím, musí se trojúhelník tvořený změřenými body co možno nejvíce shodovat s ideálním trojúhelníkem. To bude zajištěno kompenzačním algoritmem, který se snaží minimalizovat součet druhých mocnin odchylek, kterými se změřený trojúhelník převádí na ideální trojúhelník.

Případně potřebná deformace poloh měřených bodů může posloužit jako ukazatel jakosti měření a z tohoto důvodu se ukládá jako doplňková proměnná funkce MEAFRAME.

Poznámka

Frame vytvořený pomocí funkce MEAFRAME může být pomocí funkce ADDFRAME transformován do jiného framu v řetězci framů (viz příklad "Zřetězení s příkazem ADDFRAME").

Příklady

Příklad 1:

Výrobní program 1:

Programový kód
...
DEF FRAME CORR_FRAME

Nastavení měřicích bodů:

Programový kód	Komentář
DEF REAL IDEAL_POINT[3,3]= 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]= SET(10.1,0.2,-0.2,-0.2,10.2,0.1,-0.2,0.2,9.8)	; na zkoušku
DEF REAL FIT_QUALITY=0	
DEF REAL ROT_FRAME_LIMIT=5	; povolené pootočení polohy součásti max. 5 stupňů
DEF REAL FIT_QUALITY_LIMIT=3	; povolené posunutí mezi ideálním a změřeným trojúhelníkem max. 3 mm
DEF REAL SHOW_MCS_POS1[3]	
DEF REAL SHOW_MCS_POS2[3]	
DEF REAL SHOW_MCS_POS3[3]	

Programový kód	Komentář
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	

Programový kód	**Komentář**
N250 IF CORR_FRAME[X,RT] > ROT_FRAME_LIMIT	; Omezení 1. úhlu RPY
SETAL(65020)	
GOTOF NO_FRAME	
ENDIF	
N260 IF CORR_FRAME[Y,RT] > ROT_FRAME_LIMIT	; Omezení 2. úhlu RPY.
SETAL(65021)	
GOTOF NO_FRAME	
ENDIF	
N270 IF CORR_FRAME[Z,RT] > ROT_FRAME_LIMIT	; Omezení 3. úhlu RPY
SETAL(65022)	
GOTOF NO_FRAME	
ENDIF	
N300 $P_IFRAME=CORR_FRAME	; aktivování nasnímaného framu s nastavitelným framem
	; kontrola framu najetím geometrickými osami na ideální body
N400 X=IDEAL_POINT[0,0] Y=IDEAL_POINT[0,1] Z=IDEAL_POINT[0,2]	
N410 SHOW_MCS_POS1[0]=$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	; deaktivování nastavitelného framu nulovým framem (předem není dosazena žádná hodnota)
No_FRAME	; deaktivování nastavitelného framu nulovým framem (předem není dosazena žádná hodnota)
M0	
M30	

Příklad 2: Zřetězení framů

Zřetězení framů MEAFRAME za účelem korekce

Výsledkem funkce MEAFRAME je korekční frame. Pokud je tento korekční grame zřetězen s nastavitelným framem $P_UIFR[1], který byl aktivní při vyvolání funkce (např. G54), je takto získán nastavitelný frame pro další přepočítávání pro potřeby posuvů nebo obrábění.

Zřetězení s příkazem ADDFRAME

Pokud se má tento korekční frame uplatnit v řetězci framů na nějakém jiném místě nebo pokud jsou před nastavitelným framem aktivní ještě nějaké jiné framy, potom je možné pro začlenění tohoto framu do řetězce v základním framu kanálu nebo v systémovém framu použít funkci ADDFRAME.

Ve framu přitom nesmí být aktivní:

- Zrcadlové převrácení s příkazem MIRROR
- Změna měřítka s příkazem SCALE

Vstupními parametry pro požadovanou a skutečnou hodnotu jsou souřadnice obrobku. V základním systému řídícího systému je potřeba tyto souřadnice vždy udávat v metrických jednotkách nebo v palcích (G71/G70) a jako rozměr vztažený k rádiusu (DIAMOF).

Pokud budete potřebovat další informace k příkazu ADDFRAME, viz Příručka k funkcím, Základní funkce.

4.8.8 Globální framy

V rámci jednoho řídícího systému existuje jen jeden globální frame platný pro všechny kanály. Globální framy mohou být ze všech kanálů čteny a všechny kanály do nich mohou také zapisovat. Aktivování globálního framu se uskutečňuje v příslušném kanálu.

Prostřednictvím globálního framu mohou být pro **kanálové osy a osy stroje** realizována posunutí, změny měřítka a zrcadlová převrácení.

Geometrické souvislosti a řetězce framů

V případě globálních framů neexistuje žádná geometrická souvislost mezi osami. Z tohoto důvodu není možné provádět otočení nebo používat programování identifikátorů geometrických os.

Na globální framy se nedají používat žádná otočení. Naprogramování otočení je odmítnuto a aktivuje se přitom alarm:"18310 Kanál %1 blok %2 Frame: Otočení nepřípustné".

Zřetězení globálních framů a kanálových framů je možné. Výsledný frame obsahuje všechny složky framu včetně otočení pro všechny osy. Přiřazení framu se složkami otočení globálnímu gramu je odmítnuto, přičemž se aktivuje alarm "Frame: Otočení nepřípustné".

Globální framy

Globální základní frame $P_NCBFR[n]

V konfiguraci může být nastaveno až 8 globálních základních framů:

Současně mohou být k dispozici kanálové základní framy.

Globální základní framy mohou být ze všech kanálů řídícího systému čteny a všechny tyto kanály do nich mohou také zapisovat. Při zápisu do globálního framu se musí uživatel postarat o koordinaci kanálů. To může být realizováno např. značkami pro čekání (WAITMC).

Poznámka

Výrobce stroje

Počet globálních základních framů se nastavuje v konfiguraci pomocí strojního parametru.

Další informace: Příručka k funkcím, Základní funkce

Nastavitelné framy $P_UIFR[n]

Všechny nastavitelné framy G500, G54...G599 mohou být konfigurovány buď jako globální nebo jako kanálové.

Poznámka

Výrobce stroje

Prostřednictvím strojního parametru MD18601 $MN_MM_NUM_GLOBAL_USER_FRAMES mohou být všechny nastavitelné framy překonfigurovány na globální framy.

Jako identifikátory os se u příkazů pro programování framů mohou používat identifikátory kanálových os a identifikátory os stroje. Programování geometrických os je odmítnuto, přičemž se aktivuje alarm.

4.8.8.1 Specifické kanálové framy ($P_CHBFR, $P_UBFR)

Do nastavitelných framů nebo základních framů je možné zapisovat nebo je číst pomocí výrobního programu nebo pomocí BTSS prostřednictvím ovládacího panelu a PLC.

Pro globální framy je možné také jemné posunutí. Potlačení globálních framů se uskutečňuje stejně jako u kanálových framů, tedy pomocí příkazů G53, G153, SUPA a G500.

Výrobce stroje

Pomocí strojního parametru MD28081 $MC_MM_NUM_BASE_FRAMES může být v konfiguraci nastaven počet základních framů v kanálu. Standardní konfigurace je nastavena tak, aby v každém kanálu existoval nejméně jeden základní frame. V každém kanálu je možných maximálně 8 základních framů. Kromě 8 základních framů může v kanálu existovat ještě dalších 8 globálních základních framů v NCU.

Kanálové framy

$P_CHBFR[n]

Pomocí systémové proměnné $P_CHBFR[n] je možné zkladní framy číst a zapisovat do nich. Při zápisu do základního framu není zřetězený celkový základní frame ještě aktivován; toto aktivování se uskuteční teprve se zpracováním některého z příkazů G500, G54 ... G599. Tato proměnná slouží především jako paměť pro operace zápisu do základního framu z HMI nebo z PLC. Tyto proměnné typu FRAME se ukládají prostřednictvím zálohování dat.

První základní frame v kanálu

Zápis do předem definované proměnné $P_UBFR neaktivuje základní frame s indexem pole 0 v daný okamžik; toto aktivování se uskuteční teprve se zpracováním některého z příkazů G500, G54 ... G599. Do této proměnné lze zapisovat také v programu a lze ji tam i číst.

$P_UBFR

Proměnná $P_UBFR je identická s proměnnou $P_CHBFR[0]. Standardně existuje v kanálu vždy jeden základní frame, takže systémová proměnná je kompatibilní i se staršími verzemi. Pokud žádný kanálový základní frame neexistuje, při zápisu nebo při čtení se aktivuje alarm: "Frame: Nepřípustný příkaz".

4.8.8.2 Framy aktivní v kanálu

V kanálu platné framy jsou ve výrobním programu zadávány prostřednictvím příslušných systémových proměnných těchto framů. K tomu patří i systémové framy. Pomocí těchto systémových proměnných je možné ve výrobním programu načíst aktuální systémový frame a lze do něj i zapisovat.

Aktuální framy aktivní v kanálu

Přehled

Aktuální systémové framy	pro:
$P_PARTFRAME	TCARR a PAROT
$P_SETFRAME	Dosazení skutečné hodnoty a škrábnutí
$P_EXTFRAME	Externí posunutí počátku
$P_NCBFRAME[n]	Aktuální globální základní framy
$P_CHBFRAME[n]	Aktuální kanálové základní framy
$P_BFRAME	Aktuální 1. základní frame v kanálu
$P_ACTBFRAME	Celkový základní frame
$P_CHBFRMASK a $P_NCBFRMASK	Celkový základní frame
$P_IFRAME	Aktuální nastavitelný frame
Aktuální systémové framy	pro:
$P_TOOLFRAME	TOROT a TOFRAME
$P_WPFRAME	Vztažné body obrobku
$P_TRAFRAME	Transformace
$P_PFRAME	Aktuální programovatelný frame
Aktuální systémový frame	pro:
$P_CYCFRAME	Cykly
P_ACTFRAME	Aktuální celkový frame
Zřetězení framů	Aktuální frame se skládá z celkového základního framu

$P_NCBFRAME[n] - Aktuální globální základní framy

Pomocí systémové proměnné $P_NCBFRAME[n] je možné prvky pole aktuálního globálního základního framu číst a zapisovat do nich. Výsledný celkový základní frame se prostřednictvím operace zápisu v kanálu započítá.

Změněný frame se aktivuje pouze v kanálu, v němž byl frame naprogramován. Jestliže má být frame změněn pro všechny kanály daného řídícího systému, musí být data zapsána současně do obou proměnných $P_NCBFR[n] a $P_NCBFRAME[n]. Ostatní kanály musí ještě frame aktivovat, např. příkazem G54. Při zápisu základního framu se celkový základní frame nově vypočítá.

$P_CHBFRAME[n] - Aktuální kanálové základní framy

Pomocí systémové proměnné $P_CHBFRAME[n] je možné prvky pole aktuálního kanálového základního framu číst a zapisovat do nich. Výsledný celkový základní frame se prostřednictvím operace zápisu v kanálu započítá. Při zápisu základního framu se celkový základní frame nově vypočítá.

$P_BFRAME - Aktuální 1. základní frame v kanálu

Prostřednictvím předem definované proměnné $P_BFRAME typu frame je možné ve výrobním programu číst aktuální základní frame s indexem 0, který je v kanálu platný, a lze do něj i zapisovat. Zapsaný základní frame se okamžitě započítá.

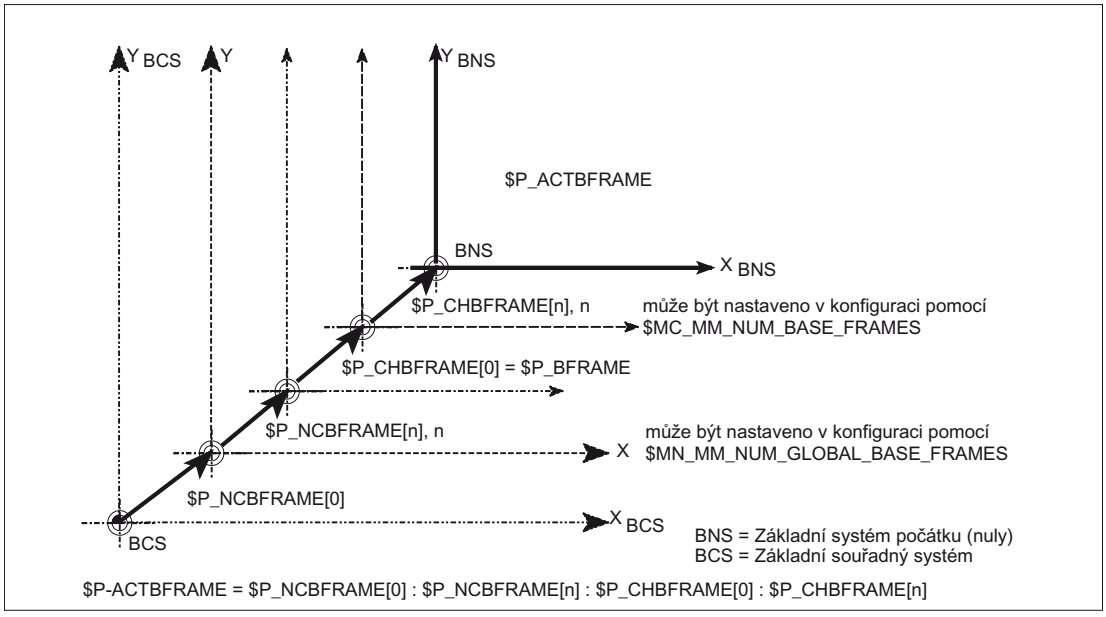
Proměnná $P_BFRAME je identická s proměnnou $P_BFRAME [0]. Tato systémová proměnná má standardně vždy platnou hodnotu. Pokud žádný kanálový základní frame neexistuje, při zápisu nebo při čtení se aktivuje alarm: "Frame: Nepřípustný příkaz".

$P_ACTBFRAME - celkový základní frame

Proměnná $P_ACTFRAME zjišťuje zřetězený celkový základní frame. Tato proměnná je jen pro čtení.

Proměnná $P_ACTFRAME odpovídá:

$P_NCBFRAME[0] : ... : $P_NCBFRAME[n] : $P_CHBFRAME[0] : ... : $P_CHBFRAME[n].



$P_CHBFRMASK a $P_NCBFRMASK - celkový základní frame

Pomocí systémových proměnných $P_CHBFRMASK a $P_NCBFRMASK může uživatel zvolit, které základní framy by chtěl zahrnout do výpočtu "celkového" základního framu. Proměnné mohou být naprogramovány jen v programu a mohou být čteny pomocí BTSS. Hodnota proměnné je interpretována jako bitová maska a udává, který prvek pole základního framu z proměnné $P_ACTFRAME je vtažen do výpočtu.

Pomocí parametrů $P_CHBFRMASK (kanálové základní framy) a $P_NCBFRMASK (základní globální framy) může být zadáno, které framy se započítávají.

Naprogramováním těchto proměnných se celkový základní frame a celkový frame znovu vypočítají. Po resetu a v základním nastavení jsou hodnoty $P_CHBFRMASK a $P_NCBFRMASK následující:

$P_CHBFRMASK = $MC_CHBFRAME_RESET_MASK

$P_NCBFRMASK = $MC_CHBFRAME_RESET_MASK

Příklad:

```
$P_NCBFRMASK = 'H81' ;$P_NCBFRAME[0] : $P_NCBFRAME[7]
$P_CHBFRMASK = 'H11' ;$P_CHBFRAME[0] : $P_CHBFRAME[4]
```

$P_IFRAME - Aktuální nastavitelný frame

Prostřednictvím předem definované proměnné $P_IFRAME typu frame je možné ve výrobním programu číst aktuální nastavitelný frame, který je v kanálu platný, a lze do něj i zapisovat. Zapsaný nastavitelný frame se okamžitě započítá.

V případě globálních nastavitelných framů se změněný frame se uplatňuje pouze v kanálu, v němž byl frame naprogramován. Jestliže má být frame změněn pro všechny kanály daného řídícího systému, musí být data zapsána současně do obou proměnných $P_UIFR[n] a $P_IFRAME. Ostatní kanály musí ještě odpovídající frame aktivovat, např. příkazem G54.

$P_PFRAME - Aktuální programovatelný frame

$P_PFRAME je programovatelný frame, který vyplývá z programových příkazů TRANS/ATRANS, G58/G59, ROT/AROT, SCALE/ASCALE, MIRROR/AMIRROR, příp. z přiřazení CTRANS, CROT, CMIRROR, CSCALE do programovatelné proměnné FRAME.

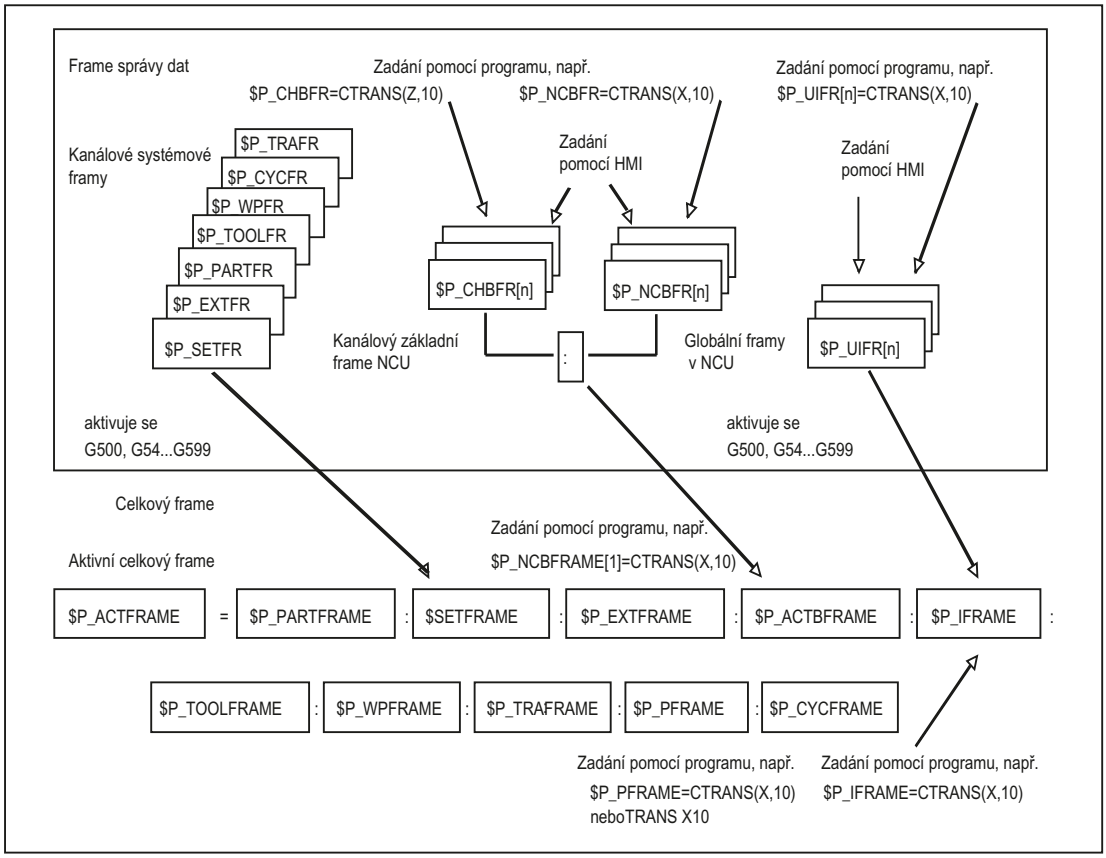
Proměnná aktuálního programovatelného framu, která definuje vztah mezi nastavitelným souřadným systémem (ENS) a souřadným systémem obrobku (WCS).

P_ACTFRAME - Aktuální celkový frame

Aktuální výsledný celkový frame $P_ACTFRAME nyní vzniká jako zřetězení všech základních framů. aktuálního nastavitelného framu a programovatelného framu. Aktuální frame se aktualizuje, kdykoli se změní některá ze složek tohoto framu.

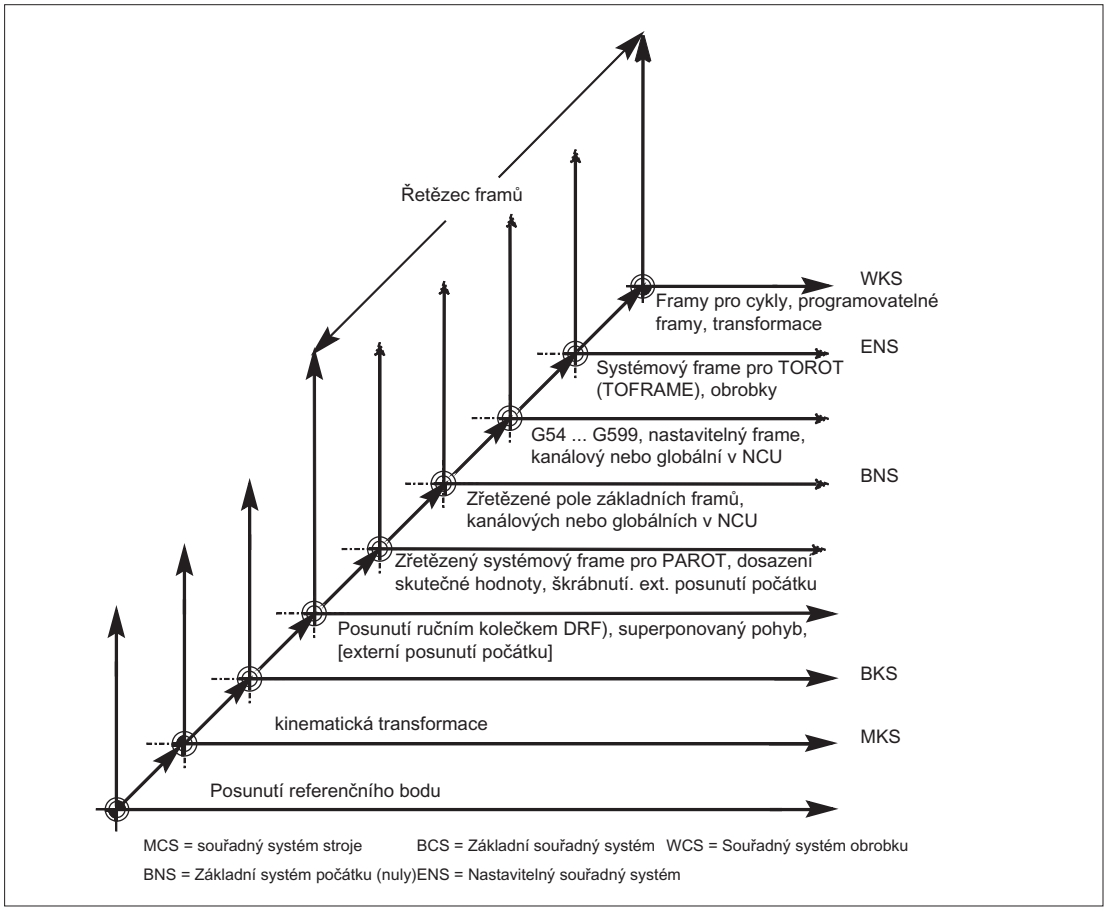
Proměnná `$P_ACTFRAME` odpovídá:

```
$P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME : $P_ACTBFRAME : $P_IFRAME :
$P_TOOLFRAME : $P_WPFRAME : $P_TRAFRAME : $P_PFRAME : $P_CYCFRAME
```


Frame správy dat
Zadání pomocí programu, např.
$P_CHBFR=CTRANS(Z,10)
$P_NCBFR=CTRANS(X,10)
Zadání pomocí programu, např.
$P_UIFR[n]=CTRANS(X,10)
Kanálové systémové framy
$P_TRAFR
$P_CYCFR
$P_WPFR
$P_TOOLFR
$P_PARTFR
$P_EXTFR
$P_SETFR
Zadání pomocí HMI
Zadání pomocí HMI
$P_CHBFR[n]
$P_NCBFR[n]
$P_UIFR[n]
Kanálový základní frame NCU
:
Globální framy v NCU
aktivuje se G500, G54...G599
aktivuje se G500, G54...G599
Celkový frame
Zadání pomocí programu, např.
$P_NCBFRAME[1]=CTRANS(X,10)
Aktivní celkový frame
$P_ACTFRAME = $P_PARTFRAME : $SETFRAME : $P_EXTFRAME : $P_ACTBFRAME : $P_IFRAME :
$P_TOOLFRAME : $P_WPFRAME : $P_TRAFRAME : $P_PFRAME : $P_CYCFRAME
Zadání pomocí programu, např.
$P_PFRAME=CTRANS(X,10)
neboTRANS X10
Zadání pomocí programu, např.
$P_IFRAME=CTRANS(X,10)

Zřetězení framů

Aktuální frame se skládá z celkového základního framu, nastavitelného framu, systémového framu a z programovatelného framu podle výše uvedeného aktuálního celkového framu.



4.9 Transformace

4.9.1 Všeobecné programování různých druhů transformace

Transformace přizpůsobují řídící systém různým kinematickým uspořádáním stroje. Aktivování transformace se uskutečňuje prostřednictvím volání funkce v NC programu. Pomocí těchto parametrů je možno odpovídajícím způsobem nastavit orientaci nástroje v prostoru, ale také provádět orientační pohyby kruhových os.

V případě transformací orientace (transformace probíhající ve třech, čtyřech a pěti osách) se naprogramované údaje polohy vždy vztahují na špičku nástroje, který je nastavován tak, aby byl vždy ortogonálně vůči obráběné ploše nacházející se v prostoru. Kartézské souřadnice jsou ze základního souřadného systému přepočítány do souřadného systému stroje a jsou vztaženy na

geometrické osy. Tyto souřadnice popisují pracovní bod. Virtuální kruhové osy popisují orientaci nástroje v prostoru.

V případě kinematických transformací (TRANSMIT, TRACYL, TRAANG) mohou být programovány polohy v kartézském souřadném systému. Řídící systém transformuje naprogramované pohyby posuvů v kartézském souřadném systému na pohyby zajišťující posuvy reálných os stroje.

Programování

Aktivování transformace orientace

```
TRAORI(<číslo transformace>,<vektor orientace>,<offsety kruhových
os>)
```

Aktivování kinematické transformace

```
TRANSMIT(<číslo transformace>)
```

```
TRACYL(<pracovní průměr>,<číslo transformace>)
```

```
TRAANG(<úhel šikmo postavené osy, číslo transformace>)
```

Deaktivování aktivní transformace

```
TRAFOOF
```

Transformace orientace

Transformace ve třech, čtyřech a pěti osách (TRAORI)

Za účelem optimálního opracování prostorově tvarovaných ploch v pracovním prostoru stroje potřebují obráběcí stroje kromě tří lineárních os X, Y a Z ještě další osy. Tyto doplňkové osy popisují orientaci v prostoru a jsou proto nazývány orientačními osami. U mnoha typů strojů s různou kinematikou jsou k dispozici jako kruhové osy:

1. Naklápěcí hlavička se dvěma osami
   Např. kardanová nástrojová hlavička s kruhovou osou rovnoběžně s lineární osou s pevným nástrojovým stolem.
2. Otočný stůl se dvěma osami
   Např . pevná naklápěcí hlavička s otočným nástrojovým stolem okolo dvou os.
3. Naklápěcí hlavička s jednou osou a otočný stůl s jednou osou
   Např . otočná naklápěcí hlavička s otočeným nástrojem a s otočným nástrojovým stolem okolo jedné osy.
4. Naklápěcí hlavička se dvěma osami a otočný stůl s jednou osou
   Např . s nástrojovým stolem otočným okolo jedné osy a s otočnou naklápěcí hlavičkou s nástrojem, který se může otáčet okolo sebe sama.

3- a 4-osá transformace jsou zvláštní případy 5-osé transformace a programují se analogicky k této 5-osé transformaci.

"**Generická 3-/4-/5-/6-osá transformace**" pokrývá svou funkcí transformace pravoúhle uspořádaných kruhových os, stejně jako transformace pro kardanovou frézovací hlavičku, a může být aktivována příkazem TRAORI stejně jako jakákoli jiná transformace orientace i pro tyto typy strojů se čtyřmi osami. V případě generické 5-/6-osé transformace má orientace nástroje

ještě další třetí stupeň volnosti, který umožňuje, že se nástroj nacházející se v libovolném směru v prostoru může ještě otáčet okolo své vlastní osy.

Další informace: Příručka k funkcím, Transformace

Základní nastavení orientace nástroje nezávislé na kinematice

ORIRESET

Pokud je aktivní transformace orientace TRAORI, potom je možné pomocí příkazu ORIRESET zadat základní polohy až 3 orientačních os s volitelnými parametry A, B, C. Přiřazení naprogramovaných parametrů kruhovým osám se uskutečňuje podle posloupnosti orientačních os definované prostřednictvím transformace. Naprogramování příkazu ORIRESET(A, B, C) způsobí, že orientační osy lineárně a synchronně najedou ze své momentální polohy do zadané základní polohy.

Kinematické transformace

TRANSMIT a TRACYL

V případě frézovacích prací na soustruzích je možno pro deklarované transformace aktivovat následující druhy obrábění:

- Opracování na čelní ploše při upnutí pro soustružení pomocí transformace TRANSMIT
- Obrábění drážek libovolného průběhu na válcových tělesech pomocí transformace TRACYL

TRAANG

Pokud má být přísuvná osa, např. pro technologii broušení, přisouvána také šikmo, může být pomocí transformace TRAANG pro deklarovanou transformaci naprogramován příslušný úhlový parametr.

Posuv PTP v kartézských souřadnicích

Ke kinematickým transformacím patří také "Posuv PTP" v kartézských souřadnicích", při kterém může být naprogramováno až 8 různých kloubových nastavení (STAT=...). Polohy jsou naprogramovány v kartézském souřadném systému, přičemž pohyb se uskutečňuje v souřadném systému stroje.

Další informace: Příručka k funkcím, Transformace

Zřetězené transformace

Mohou být za sebou poskládány vždy maximálně dvě transformace. U takto zřetězené druhé transformace se pohybové složky os přebírají z první transformace.

Jako první transformaci je možno zadat následující:

- Transformace orientace TRAORI
- Polární transformace TRANSMIT
- Transformace válcového pláště TRACYL
- Transformace Šikmá osa TRAANG

Druhou transformací musí být transformace „Šikmá osa TRAANG“.

4.9.1.1 Pohyby provádějící změnu orientace při transformacích

Posuvové pohyby a orientační pohyby

Posuvové pohyby programovatelných orientací závisí především na typu stroje. V případě tří-, čtyř- a pětiosé transformace s funkcí TRAORI popisují rotační osy nebo naklápěcí lineární osy orientační pohyby nástroje.

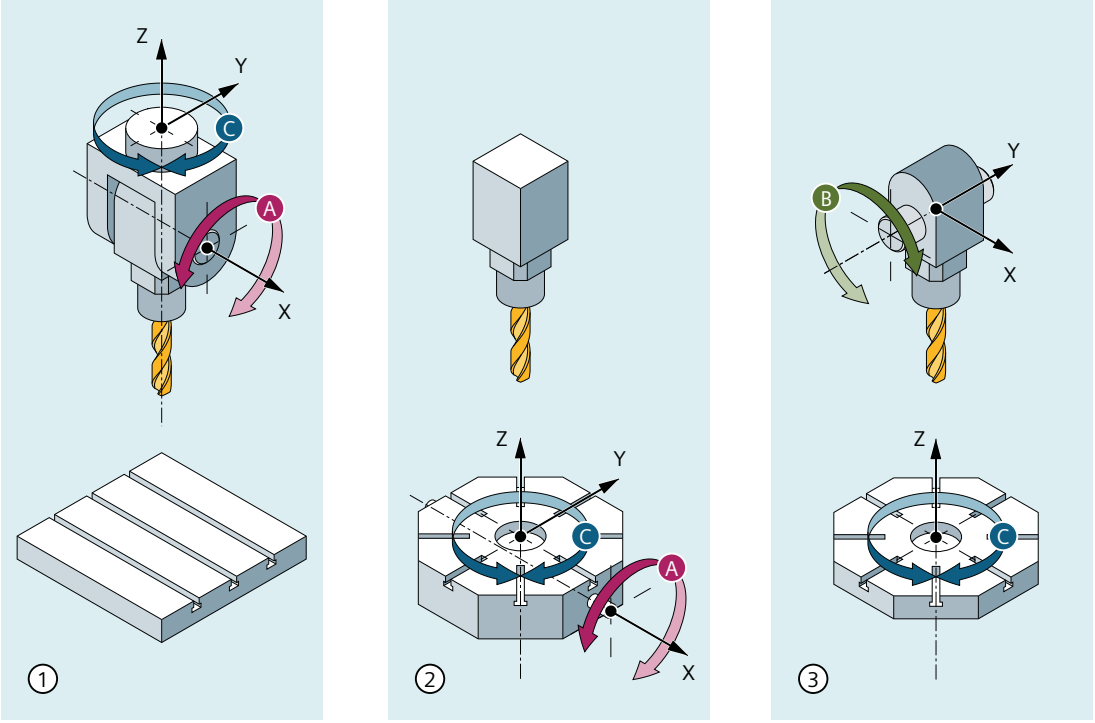
Změny poloh kruhových os podílejících se na transformaci orientace mají za následek kompenzační pohyby zbývajících strojních os. Poloha špičky nástroje zůstává přitom nezměněna.

Orientační pohyby nástroje mohou být v závislosti na použití naprogramovány prostřednictvím identifikátorů virtuálních kruhových os A..., B..., C... buď prostřednictvím zadání Eulerova úhlu nebo úhlu RPY nebo směrových vektorů nebo normálových vektorů plochy, normovaných vektorů pro osu kužele nebo pro pomocnou orientaci na kuželovou plášťovou plochu.

Při kinematické transformaci pomocí příkazů TRANSMIT, TRACYL a TRAANG řídící systém transformuje posuvové pohyby naprogramované v kartézském souřadném systému na posuvové pohyby reálných os stroje.

Kinematika stroje při transformace ve třech, čtyřech a pěti osách (TRAORI)

Otočný může být buď nástroj nebo nástrojový stůl s až maximálně dvěma kruhovými osami. Možné jsou také kombinace naklápěcí hlavičky s jednou osou a otočného stolu.



① Typ stroje 1 s posloupnosti os CA
- Držák nástroje otočný kolem osy X a kolem osy Z
- Obrobkový stůl pevný

② Typ stroje 2 s posloupnosti os AC
- Držák nástroje pevný
- Obrobkový stůl otočný kolem osy X a kolem osy Z

③ Typ stroje 3 s posloupnosti os BC
- Držák nástroje otočný kolem osy Y
- Obrobkový stůl otočný kolem osy Z

Typ stroje	Programování orientace
Tříosá transformace u strojů typu 1 a 2	Programování orientace nástroje jen v rovině, která je kolmá na osu otáčení Existují **dvě** translační osy (lineární osy) a **jedna** rotační osa (kruhová osa).
Čtyřosá transformace u strojů typu 1 a 2	Programování orientace nástroje jen v rovině, která je kolmá na osu otáčení Existují **tři** translační osy (lineární osy) a **jedna** rotační osa (kruhová osa).
Pětiosá transformace pro typ stroje 3 Naklápěcí hlavička s jednou osou a otočný stůl s jednou osou	Programování transformace orientace. Kinematika se **třemi** lineárními osami a **dvěma** ortogonálními kruhovými osami. Kruhové osy jsou rovnoběžné se dvěma ze tří lineárních os. První kruhová osa se pohybuje dvěma kartézskými lineárními osami. Otáčí třetí lineární osu s nástrojem. Druhá kruhová osa otáčí obrobkem.

Generické 5/6-osé transformace

Typ stroje	Programování transformace orientace
Generická Pěti-/šestiosá transformace, typ stroje 4 Naklápěcí hlavička se dvěma osami a s nástrojem, který se může otáčet okolo sebe sama, a otočný stůl s jednou osou	Programování transformace orientace. Kinematika se **třemi** lineárními osami a **třemi** ortogonálními kruhovými osami. Kruhové osy jsou rovnoběžné se dvěma ze tří lineárních os. První kruhová osa se pohybuje dvěma kartézskými lineárními osami. Otáčí třetí lineární osu s nástrojem. Druhá kruhová osa otáčí obrobkem. Základní orientace nástroje může být naprogramována pomocí dodatečného otočení nástroje okolo jeho osy o úhel THETA.

Při vyvolávání "generické tří-, čtyř- a pěti-/šestiosé transformace" může být navíc ještě předávána také základní orientace nástroje. Omezení týkající se směrů u kruhových os už neplatí. Jestliže kruhové osy nejsou uspořádány přesně kolmo na sebe nebo pokud se stávající kruhové osy nenacházejí přesně rovnoběžně s lineárními osami, je možné pomocí "generické pěti-/šestiosé transformace" dosáhnout lepších výsledků orientace nástroje.

Kinematické transformace TRANSMIT, TRACYL a TRAANG

Pro frézovací práce na soustruzích nebo pro šikmo nastavitelnou osu při broušení platí v závislosti na transformaci ve standardním případě pro osy následující požadavky:

TRANSMIT	Aktivování polární transformace
Obrábění na čelní ploše při upnutí pro soustružení	Kruhová osa Přísuvná osa kolmá ke kruhové ose Podélná osa rovnoběžná s otočnou osou

TRACYL	Aktivování transformace válcového pláště
Obrábění drážek libovolného průběhu na válcovém tělese	Kruhová osa Přísuvná osa kolmá ke kruhové ose Podélná osa rovnoběžná s otočnou osou

TRAANG	Aktivování transformace šikmé osy
Obrábění s šikmou přísuvnou osou	Kruhová osa Přísuvná osa s úhlem, který může být nastaven pomocí parametru Podélná osa rovnoběžná s otočnou osou

Posuv PTP v kartézských souřadnicích

Pohyb stroje se uskutečňuje pomocí souřadnic stroje a pro jeho programování se používá:

TRAORI	Aktivování transformace
PTP - posuv od bodu k bodu	Poloha v kartézském souřadném systému (MCS)
CP	Pohyb po dráze kartézskými osami (BCS)
STAT	Nastavení polohy kloubu závisí na transformaci
TU	O jaký úhel mají osy najíždět po nejkratší dráze

Najíždění PTP při generické 5/6-osé transformaci

Pohyb stroje se uskutečňuje v souřadnicích stroje a orientaci nástroje je možné naprogramovat jak pomocí poloh kruhových os, tak také pomocí Eulerova vektoru, příp. úhlu RPY nezávislých na kinematické stroje nebo pomocí směrových vektorů.

Přitom jsou možné interpolace kruhových os, vektorové interpolace s využitím interpolace největší kružnice koule nebo interpolace vektoru orientace na kuželové válcové ploše.

Příklad tří- až pětiosé transformace u kardanové frézovací hlavičky

Obráběcí stroj má minimálně 5 os, z nichž jsou:

- tři osy lineární s přímkovými pohyby, které mohou najet na pracovní bod v jakékoli libovolné poloze v pracovním prostoru
- dvě otočné naklápěcí osy, které jsou uspořádány pod úhlem nastaveným v konfiguraci (většinou 45 stupňů) a které umožňují nastavování orientace nástroje v prostoru, které je v případě uspořádání pod úhlem 45 stupňů omezeno na polokouli

4.9.1.2 Přehled transformace orientace TRAORI

Možné druhy programování v souvislosti s transformací TRAORI

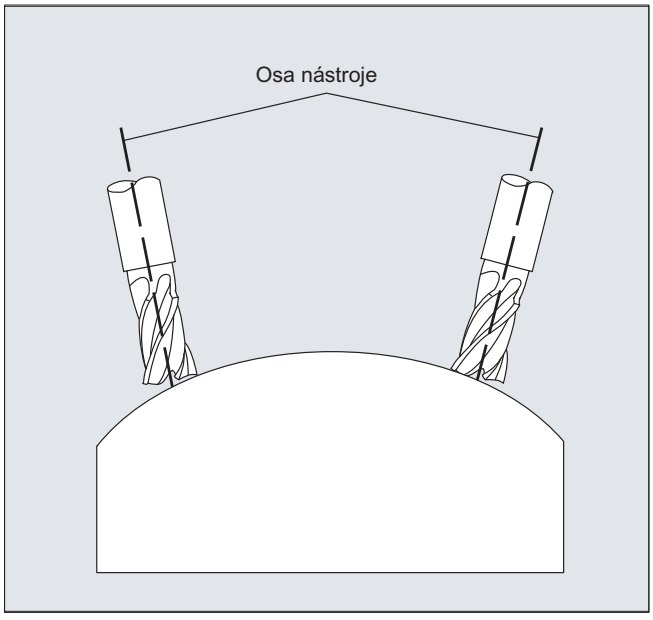
Typ stroje	Programování při aktivní transformaci TRAORI
Typy strojů 1, 2 nebo 3. Naklápěcí hlavička se dvěma osami nebo otočný stůl se dvěma osami nebo kombinace naklápěcí hlavičky a otočného stolu, které mají po jedné ose.	Posloupnost orientačních os a směrů orientace nástroje je buď **vztažena ke stroji** nastavuje se v konfiguraci pomocí strojních parametrů v závislosti na kinematice stroje nebo **vztaženo na souř. systém obrobku** s programovatelnou orientací nezávisle na kinematice stroje Směr otáčení orientačních os se ve vztažném systému naprogramuje pomocí příkazů - ORIMKS vztažný systém = souřadný systém stroje - ORIWKS vztažný systém = souřadný systém obrobku Základní nastavení je ORIWKS. Programování orientačních os zadáním těchto dat: A, B, C - polohy os stroje přímo A2, B2, C2 - programování úhlu virtuálních os pomocí - ORIEULER - pomocí Eulerova úhlu (standardní) - ORIRPY - pomocí úhlu RPY - ORIVIRT1 - pomocí virtuálních orientačních os (definice 1) - ORIVIRT2 - pomocí virtuálních orientačních os (definice 2) s rozlišením druhu interpolace: **lineární interpolace** - ORIAXES pro orientační osy nebo osy stroje **interpolace pomocí největší kružnice koule** (interpolace vektoru orientace) - ORIVECT pro orientační osy Programování orientačních os zadáním těchto dat: A3, B3, C3 - vektorové složky (směr/normála k ploše) Programování výsledné orientace nástroje A4, B4, C4 - normálové vektory plochy na začátku bloku A5, B5, C5 - normálové vektory plochy na konci bloku LEAD - předozadní úhel pro orientaci nástroje TILT - úhel bočního naklopení pro orientaci nástroje

Typ stroje	Programování při aktivní transformaci TRAORI
	Interpolace vektoru orientace na kuželové plášťové ploše Změny orientace na kuželové plášťové ploše nacházející se **libovolně v prostoru** prostřednictvím interpolace: - ORIPLANE v rovině (interpolace pomocí největší kružnice koule) - ORICONCW - po ploše pláště kužele ve směru hodinových ručiček - ORICONCCW - po ploše pláště kužele proti směru hodinových ručiček A6, B6, C6 - směrové vektory (osa otáčení kužele) -OICONIO - Interpolace na kuželové plášťové ploše pomocí: A7, B7, C7 - pomocné vektory (počáteční a koncová orientace) nebo - ORICONTO - na ploše pláště kuželu s tangenciálním přechodem Změny orientace vztahující se **na dráhu** s příkazy - ORICURVE - zadání pohybu přes dva uzlové body pomocí PO[XH]=(xe, x2, x3, x4, x5) - polynom orientace až do 5. stupně PO[YH]=(ye, y2, y3, y4, y5) - polynom orientace až do 5. stupně PO[ZH]=(ze, z2, z3, z4, z5) - polynom orientace až do 5. stupně - ORIPATHS - vyhlazování charakteristiky orientace A8, B8, C8 - Fáze změny orientace nástroje odpovídá: Směru a délce dráhy nástroje při pozvedávacím pohybu
Typy stroje 1 a 3 Další typy strojů, které umožňují otáčení nástroje okolo sebe sama, vyžadují 3. osu Transformace orientace, jako např. generická 6-osá transformace. Otáčení vektoru orientace.	Programování **otočení** orientace nástroje pomocí následujících parametrů: LEAD - předozadní úhel vzhledem k normálovému vektoru plochy PO[PHI] - programování polynomu až 5. stupně TILT - boční úhel otočení k tečně ke dráze (směr Z) PO[PSI] - programování polynomu až 5. stupně THETA - úhel otočení (otáčení okolo směru nástroje ve směru Z) THETA= hodnota, která bude dosažena na konci bloku THETA=AC(...) - blokové přepnutí na zadávání absolutních rozměrů THETA=AC(...) - blokové přepnutí na zadávání inkrementálních rozměrů THETA=Θ_e - interpolace naprogramovaného úhlu G90/G91 PO[THT]=(..) - programování polynomu až 5. stupně Programování vektoru otočení - ORIROTA - absolutní otočení - ORIROTR - relativní vektor otočení - ORIROTT - tangenciální vektor otočení
Orientace vztahující se k dráze pro popis změn orientace vzhledem ke dráze nebo otáčení vektoru tangenciálně k dráze	Změny orientace **vztažené k dráze** pomocí příkazů: - ORIPATH - Orientace nástroje vztažená na dráhu - ORIPATHS - navíc v případě skokové změny na charakteristice orientace Programování vektoru otočení - ORIROTC - tangenciální vektor otočení, otáčení k tečně dráhy

4.9.2 Transformace ve třech, čtyřech a pěti osách (TRAORI)

4.9.2.1 Všeobecné souvislosti v případě kardanové nástrojové hlavičky

Aby bylo dosaženo optimálních řezných podmínek při obrábění prostorově zakřivených ploch, musí být možné měnit úhel nastavení nástroje.

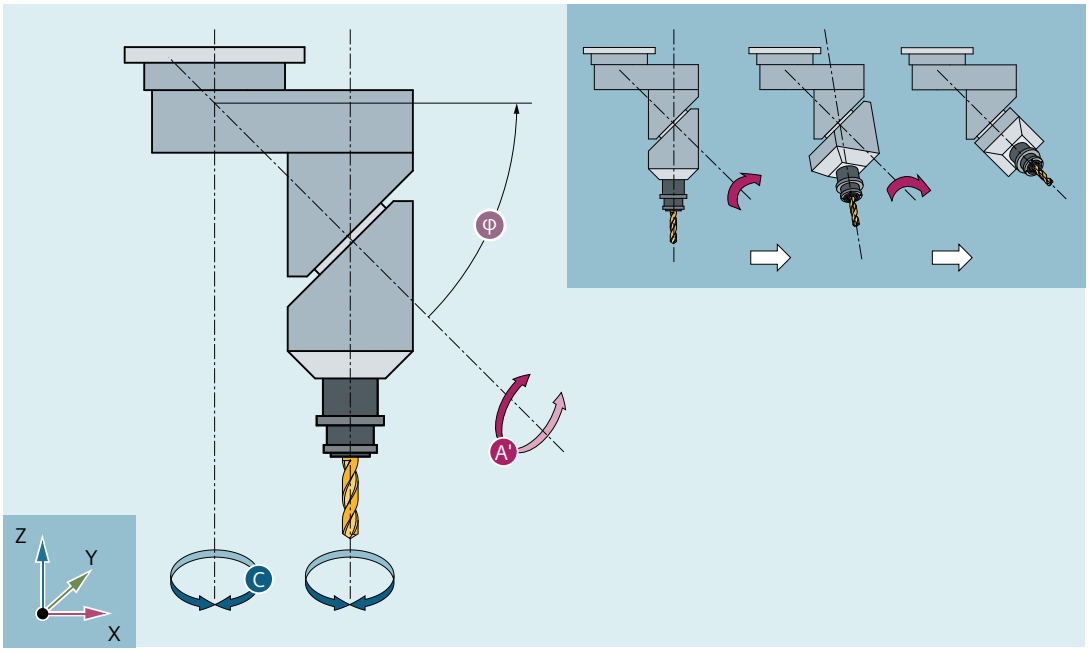


To, se kterými konstrukcemi stroje se toho dosáhne, je uloženo v parametrech os.

5-osá transformace

Kardanová nástrojová hlavička

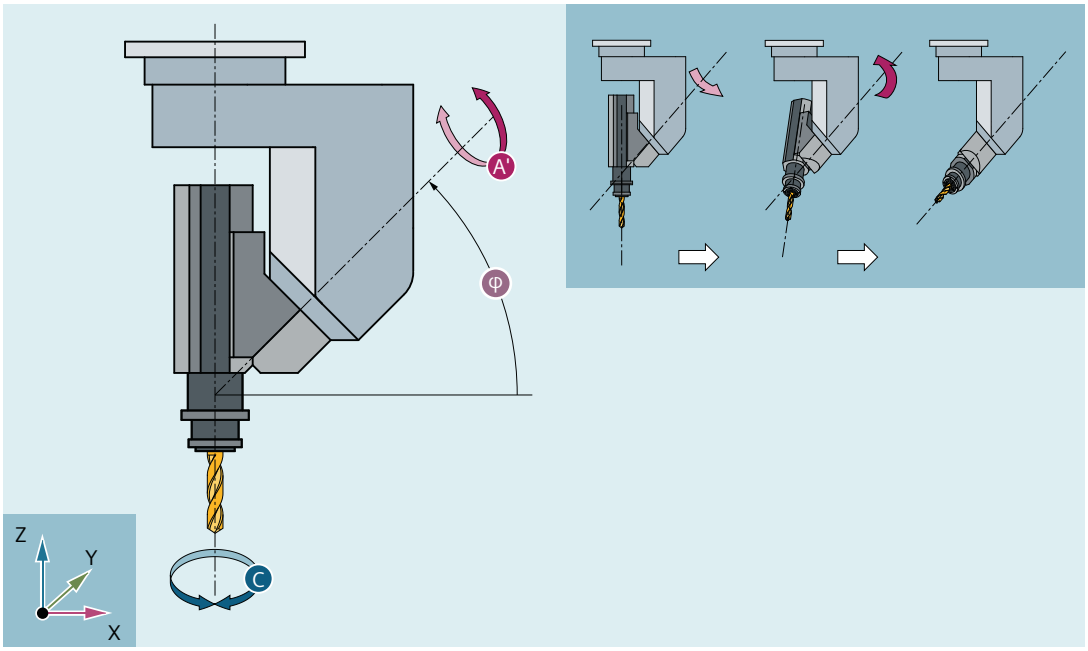
V tomto případě definují tři lineární osy (X, Y, Z) a dvě orientační osy (C, A) úhel nastavení a pracovní bod nástroje. Jedna z obou orientačních os je nastavena jako šikmá osa, zde v tomto příkladu A' - v mnoha případech jako uspořádání pod úhlem 45°.



Obrázek 4-3 Kardanová nástrojová hlavička, varianta 1

Ve zde uváděných příkladech vidíte uspořádání na případu kinematiky stroje CA s kardanovou nástrojovou hlavičkou!

Posloupnost orientačních os a směr orientace nástroje mohou být pomocí strojních parametrů nastaveny nezávisle na kinematice stroje (→ výrobce stroje).



V tomto příkladu leží A' pod úhlem φ vůči ose X

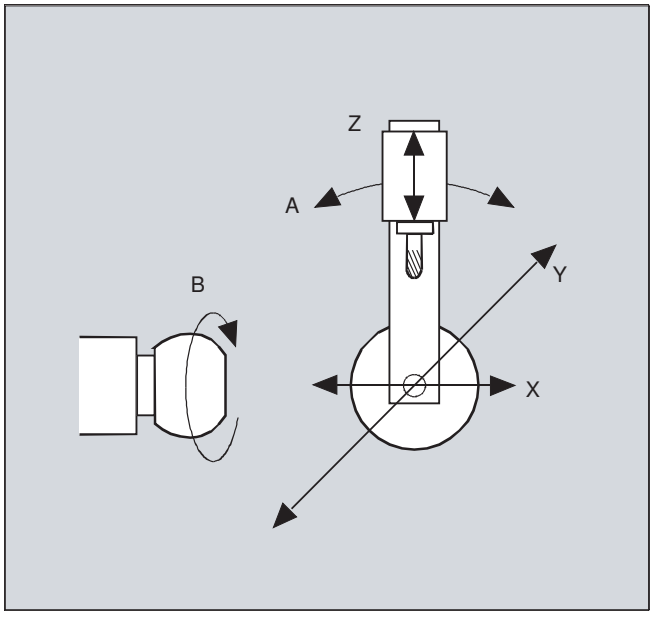
Obecně platí následující možné souvislosti:

A' leží pod úhlem φ vůči ose	Osa X
B' leží pod úhlem φ vůči ose	Osa Y
C' leží pod úhlem φ vůči ose	Osa Z

Úhel φ může být nastaven v konfiguraci pomocí strojních parametrů v rozsahu 0° až +89°.

S naklápěcí lineární osou

V tomto případě se jedná o uspořádání s pohyblivým obrobkem a pohyblivým nástrojem. Kinematika se skládá ze tří lineárních os (X, Y, Z) a tří kruhových os uspořádaných vůči sobě navzájem v pravých úhlech. První kruhová osa se pohybuje např. nad křížovým suportem dvou lineárních os, nástroj se nachází rovnoběžně se třetí lineární osou. Druhá kruhová osa otáčí obrobkem. Třetí lineární osa (naklápěcí osa) leží v rovině křížového suportu.



Posloupnost rotačních os a směr orientace nástroje mohou být pomocí strojních parametrů nastaveny nezávisle na kinematice stroje.

Platí následující možné souvislosti:

Osy:	Posloupnost os:
1. kruhová osa	A A B B C C
2. kruhová osa	B C A C A B
Naklopená lineární osa	Z Y Z X Y X

Pokud budete potřebovat další informace týkající se možností konfigurace posloupností os pro směr orientace nástroje, viz Příručka k funkcím, Transformace.

4.9.2.2 Transformace ve třech, čtyřech a pěti osách (TRAORI)

Uživatel může nastavit v konfiguraci dvě, příp. tři lineární osy a dvě osy kruhové. Transformace vychází z toho, že se kruhové osy nacházejí kolmo na orientační rovinu.

Orientace nástroje je možná pouze v rovině, která je kolmá na kruhovou osu. Tato transformace podporuje typy strojů s pohyblivým nástrojem a pohyblivým obrobkem.

Konfigurace a programování tří- a čtyřosé transformace se provádí analogicky k pětiosé transformaci.

Syntaxe

```
TRAORI(<n>)
TRAORI(<n>,<X>,<Y>,<Z>,<A>,<B>)
TRAFOOF
```

Význam

`TRAORI`:	Aktivuje první deklarovanou transformaci orientace
`TRAORI(<n>)`:	Aktivuje n-tou deklarovanou transformaci orientace
`<n>`:	Číslo transformace
	Hodnota: 1 nebo 2
	Příklad: TRAORI(1) aktivuje transformaci orientace 1
`<X>,<Y>,<Z>`:	Složky vektorů orientace, do kterých ukazuje nástroj
`<A>,<B>`:	Programovatelný offset pro kruhové osy
`TRAFOOF`:	Deaktivování transformace

Orientace nástroje

V závislosti na zvolených směrech orientace nástroje musí v NC programu nastavena aktivní pracovní rovina (G17, G18, G19) takovým způsobem, aby korekce délky nástroje působila ve směru orientace nástroje.

Poznámka

Po aktivování transformace jsou údaje o poloze (X, Y, Z) vztaženy vždy na špičku nástroje. Změna polohy kruhových os podílejících se na transformaci vede ke kompenzačním pohybům zbývajících os stroje, čímž se zajistí, že poloha špičky nástroje zůstane nezměněna.

Transformace orientace je vždy definována od špičky nástroje k jeho upevnění.

Offset pro orientační osy

Při aktivování transformace orientace je možné přímo naprogramovat doplňkový offset pro orientační osy.

Pokud je při programování dodržena správná posloupnost, smí být některé parametry vypuštěny.

Příklad:

`TRAORI(, , , ,A,B)` ; jestliže se mají zadat jen některé offsety.

Další možností vedle přímého naprogramování, jak mohou být doplňkové offsety pro orientační osy také zadány, je automatické převzetí z momentálně aktivního posunutí počátku. Tato funkce pro přebírání se nastavuje v konfiguraci pomocí strojních parametrů.

Příklady

```
TRAORI(1,0,0,1)          ; v základní orientaci je nástroj nasměrován v ose Z
TRAORI(1,0,1,0)          ; v základní orientaci je nástroj nasměrován v ose Z
TRAORI(1,0,1,1)          ; v základní orientaci nástroj ukazuje ve směru Y/Z (odpo-
                         vídá poloze -45°)
```

4.9.2.3 Varianty programování orientace a základního nastavení (ORIRESET)

Programování orientace nástroje pomocí funkce TRAORI

Ve spojení s programovatelnou transformací orientace TRAORI mohou být kromě lineárních os X, Y, Z pomocí identifikátorů kruhových os programovány také polohy os A..., B..., C... nebo virtuální osy s úhly nebo komponenty vektorů. Pro orientační osy a osy stroje jsou možné různé druhy interpolace. Nezávisle na tom, které polynomy orientace PO[úhel] a polynomy os PO[osa] jsou právě aktivní, může být naprogramován větší počet různých druhů polynomů, jako např. G1, G2, G3, CIP nebo POLY.

Změna orientace nástroje může být naprogramována také pomocí vektorů orientace. Orientace na konci každého bloku přitom může být zadávána buď přímým naprogramováním vektorů nebo naprogramováním poloh kruhových os.

Varianty programování orientace v případě tří- až pětiosých transformací

Následující varianty programování nastavení orientace se vzájemně vylučují.

`A, B, C`	Přímé zadávání poloh os stroje
`A2, B2, C2`	Programování úhlu virtuálních os pomocí Eulerova úhlu nebo úhlu RPY
`A3, B3, C3`	Zadání vektorových složek
`LEAD, TILT`	Zadávání předozadního a bočního úhlu vztažené na dráhu a plochu
`A4, B4, C4` `A5, B5, C5`	Normálový vektor plochy na začátku bloku a na konci bloku
`A6, B6, C6` `A7, B7, C7`	Interpolace vektoru orientace na kuželové plášťové ploše
`A8, B8, C8`	Změna orientace nástroje, směr a délka dráhy pohybu při pozvednutí

Najíždění na základní nastavení orientace nástroje (ORIRESET)

Prostřednictvím funkce `ORIRESET(...)` jsou osy příslušné kinematiky stroje pro nastavování orientace lineárně a synchronně přestaveny ze své aktuální polohy do naprogramované základní polohy. Jestliže pro nějakou osu není naprogramována žádná základní poloha, potom se použije poloha stanovená odpovídajícím strojním parametrem $MC_TRAFO5_ROT_AX_OFFSET_1/2.

Aktivní framy kruhových os nejsou nijak zohledňovány.

Příklady kinematiky stroje CA (názvy kanálových os C, A)

Příkazy	Popis
`ORIRESET(90, 45)`	`Osa C: 90°` `Osa A: 45°`
`ORIRESET(, 30)`	`Osa C: $MC_TRAFO5_ROT_AX_OFFSET_1/2[0]` `Osa A: 30°`
`ORIRESET( )`	`Osa C: $MC_TRAFO5_ROT_AX_OFFSET_1/2[0]` `Osa A: $MC_TRAFO5_ROT_AX_OFFSET_1/2[1]`

Příklady kinematiky stroje CAC (názvy kanálových os C, A, B)

Příkazy	Popis

`ORIRESET(90, 45, 90)`	`Osa C: 90°`
	`Osa A: 45°`
	`Osa B: 90°`

`ORIRESET( )`	`Osa C: $MC_TRAFO5_ROT_AX_OFFSET_1/2[0]`
	`Osa A: $MC_TRAFO5_ROT_AX_OFFSET_1/2[1]`
	`Osa B: $MC_TRAFO5_ROT_AX_OFFSET_1/2[2]`

Poznámka

Najíždění do základního nastavení orientace nástroje pomocí příkazu `ORIRESET(...)` se smí uskutečňovat, jen když je aktivní transformace orientace `TRAORI(...)`.

Programování otočení LEAD, TILT a THETA

Předozadní úhel LEAD a boční úhel TILT

Otočení orientace nástroje se v případě tří- až pětiosé transformace programují pomocí předozadního úhlu LEAD a pomocí bočního úhlu TILT.

Úhel otočení THETA

V případě transformace **s třetí kruhovou osou** může být naprogramováno nastavení orientace jak pomocí vektorových složek, tak také může být pomocí úhlů LEAD a TILT naprogramováno otočení nástroje okolo sebe sama s úhlem natočení THETA.

4.9.2.4 Programování orientace nástroje (A..., B..., C..., LEAD, TILT)

Pro programování orientace nástroje existují následující možnosti:

1. Přímé programování pohybu kruhových os. Změna orientace se vždy uskutečňuje v základním souřadném systému nebo v souřadném systému stroje. Orientační osy se pohybují jako synchronní osy.
2. Naprogramování Eulerových úhlů nebo úhlů RPY podle definice prostřednictvím `A2, B2, C2`.
3. Naprogramování směrových vektorů pomocí `A3, B3, C3`. Směrový vektoru ukazuje od špičky nástroje ve směru jeho unášeče (držáku).

4. Programování normálových vektorů plochy na začátku bloku pomocí `A4, B4, C4` na na konci bloku pomocí `A5, B5, C5` (frézování na čelní ploše).
5. Naprogramování pomocí předozadního úhlu `LEAD` a bočního úhlu `TILT`.
6. Naprogramování osy kužele jako normovaného vektoru pomocí `A6, B6, C6` nebo pomocné orientace na kuželové plášťové ploše pomocí `A7, B7, C7`,
   viz kapitola "Programování orientace podél kuželové plášťové plochy (ORIPLANE, ORICONxx)".
7. Programování změny orientace, směru a délky dráhy nástroje v průběhu pohybu při pozvednutí pomocí `A8, B8, C8`
   , viz kapitola "Vyhlazení průběhu orientace (ORIPATHS A8=, B8=, C8=)".

Poznámka

Ve všech případech je programování orientace přípustné jen tehdy, pokud je transformace orientace aktivována.

Výhoda: Tyto programy jsou přenositelné na jakoukoli kinematiku stroje.

Definice orientace nástroje pomocí příkazu G-funkce

Poznámka

Výrobce stroje

Pomocí strojního parametru je možné přepínat mezi Eulerovými úhly a úhly RPY. V případě odpovídajících nastavení strojních parametrů je přepínání možné jak nezávisle, tak i v závislosti na aktivním příkazu G-funkce ze skupiny 50. Můžete si vybrat z následujících možností nastavení:

1. Jestliže jsou oba strojní parametry pro definici orientačních os a definici orientačních úhlů nastaveny pomocí příkazů G-funkcí na nulu: Úhel naprogramovaný pomocí `A2, B2, C2` je **v závislosti na strojním parametru** pro definici úhlu pro programování orientace interpretován buď jako Eulerův úhel nebo jako úhel RPY.
2. Pokud je strojní parametr pro definici orientačních os nastaven pomocí příkazu G-funkce na jedna, provádí se přepínání
   v závislosti na aktivním příkazu G-funkce ze skupiny 50:
   Úhly naprogramované pomocí `A2, B2, C2` jsou interpretovány podle aktivního příkazu G-funkce `ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2, ORIAXPOS` a `ORIPY2`. Hodnoty naprogramované prostřednictvím orientačních os jsou interpretovány v souladu s aktivním příkazem G-funkce ze skupiny 50 také jako orientační úhel.
3. Jestliže je strojní parametr pro definici úhlu orientace prostřednictvím příkazu G-funkce nastaven na jedničku a strojní parametr pro definici orientačních os pomocí příkazu G-funkce je nastaven na nulu, provádí se přepínání
   nezávisle na aktivním příkazu G-funkce ze skupiny 50:
   Úhly naprogramované pomocí `A2, B2, C2` jsou interpretovány podle aktivního příkazu G-funkce `ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2 ORIAXPOS` a `ORIPY2`. Hodnoty naprogramované prostřednictvím orientačních os jsou interpretovány nezávisle na aktivním příkazu G-funkce ze skupiny 50 vždy jako polohy kruhových os.

Syntaxe

Polohy kruhových os

```
G1 X<hodnota> Y<hodnota> Z<hodnota> A<hodnota> B<hodnota> C<hodnota>
```

Eulerův úhel

```
G1 X<hodnota> Y<hodnota> Z<hodnota> A2=<hodnota> B2=<hodnota>
C2=<hodnota>
```

Směrový vektor

```
G1 X<hodnota> Y<hodnota> Z<hodnota> A3=<hodnota> B3=<hodnota>
C3=<hodnota>
```

Normálový vektor plochy na začátku bloku

```
G1 X<hodnota> Y<hodnota> Z<hodnota> A4=<hodnota> B4=<hodnota>
C4=<hodnota>
```

Normálový vektor plochy na konci bloku

```
G1 X<hodnota> Y<hodnota> Z<hodnota> A5=<hodnota> B5=<hodnota>
C5=<hodnota>
```

Předozadní úhel

```
LEAD=<hodnota>
```

Úhel bočního naklonění

```
TILT=<hodnota>
```

Význam

`G1:`	Lineární interpolace
`X, Y, Z:`	Polohy lineárních os
`A, B, C:`	Polohy kruhových os
`A2=, B2=, C2=:`	Naprogramovaný úhel (Eulerův úhel nebo úhel RPY)
`A3=, B3=, C3=:`	Směrový vektor v souřadnicích X, Y a Z ve WCS
`A4=, B4=, C4=:`	Normálový vektor plochy na začátku bloku v souřadnicích X, Y a Z ve WCS
`A5=, B5=, C5=:`	Normálový vektor plochy na konci bloku v souřadnicích X, Y a Z ve WCS
`LEAD= :`	Předozadní úhel [1)]
`TILT= :`	Boční úhel [1)]
1) Interpretace zadaných hodnot úhlu závisí na nastavení parametru MD21094 $MC_ORIPATH_MODE.	

Další informace

5-osé programy jsou zpravidla vytvářeny v systémech CAD/CAM a nejsou zadávány na řídícím systému. Z tohoto důvodu jsou následující vysvětlení určena především pro programátory postprocesorových systémů.

Pro programování orientace vám jsou k dispozici následující příkazy:

Příkaz	Význam
ORIEULER:	Eulerův úhel s posloupností při otáčení ZX'Z"
ORIRPY:	Úhel RPY s posloupnost při otáčení XY'Z"
ORIRPY2:	Úhel RPY s posloupnost při otáčení ZY'X"
ORIVIRT1:	Virtuální orientační osy s volně definovatelnou posloupností při otáčení pomocí parametru: MD21120 $MC_ORIAX_TURN_TAB_1
ORIVIRT2:	Virtuální orientační osy s volně definovatelnou posloupností při otáčení pomocí parametru: MD21130 $MC_ORIAX_TURN_TAB_2
ORIAXPOS:	Virtuální orientační osy v polohami kruhových os

Poznámka

Prostřednictvím strojních parametrů mohou být výrobcem stroje definovány různé varianty. Věnujte prosím pozornost informacím od výrobce stroje.

Programování pomocí Eulerových úhlů ORIEULER, posloupnost při otáčení Z X' Z"

Hodnoty naprogramované pomocí příkazu ORIEULER s hodnotami A2, B2, C2 při programování orientace jsou interpretovány jako Eulerův úhel (ve stupních).

Nový vektor orientace vznikne prostřednictvím následujících tří otočení původního vektoru orientace

1. pomocí kruhové osy A2 okolo souřadné osy Z
2. pomocí kruhové osy B2 okolo nové souřadné osy X'
3. pomocí kruhové osy C2 okolo nové souřadné osy Z"

V tomto případě nemá hodnota C2 (otočení okolo nové osy Z) žádný význam a nemusí být proto naprogramována.

Programování v úhlech RPY pomocí příkazu ORIRPY, posloupnost při otáčení X Y' Z"

Hodnoty naprogramované pomocí funkce ORIRPY s parametry A2, B2, C2 při programování orientace jsou interpretovány jako úhel RPY (ve stupních) s posloupností při otáčení X Y' Z".

Poznámka

Oproti programování pomocí příkazu ORIEULER mají při použití funkce ORIRPY na vektor orientace vliv všechny tři hodnoty.

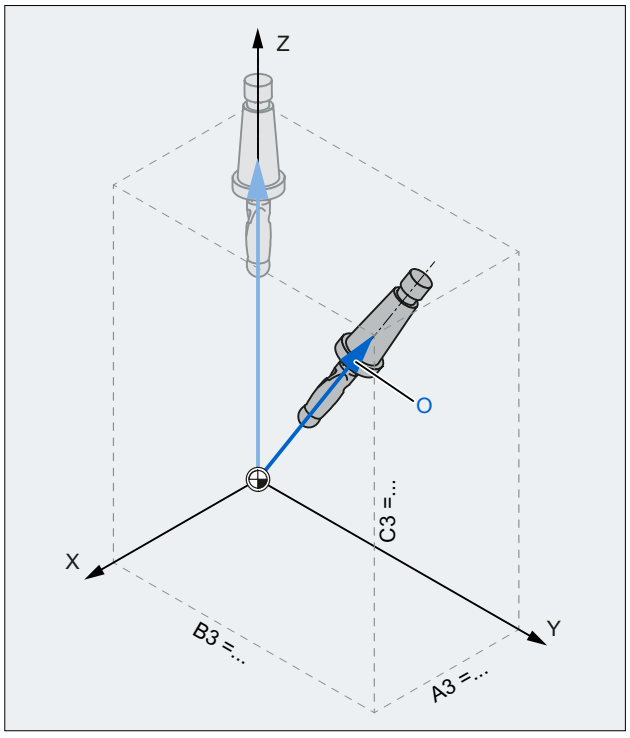
Nový vektor orientace vznikne prostřednictvím následujících tří otočení původního vektoru orientace

1. pomocí kruhové osy A2 okolo souřadné osy X
2. pomocí kruhové osy B2 okolo nové souřadné osy Y'
3. pomocí kruhové osy C2 okolo nové souřadné osy Z"

Programování směrových vektorů

Komponenty směrového vektoru jsou naprogramovány pomocí `A3, B3, C3`. Vektor ukazuje ve směru upnutí nástroje, délka vektoru přitom nemá žádný význam.

Nenaprogramovaným složkám vektoru je dosazena nulová hodnota.



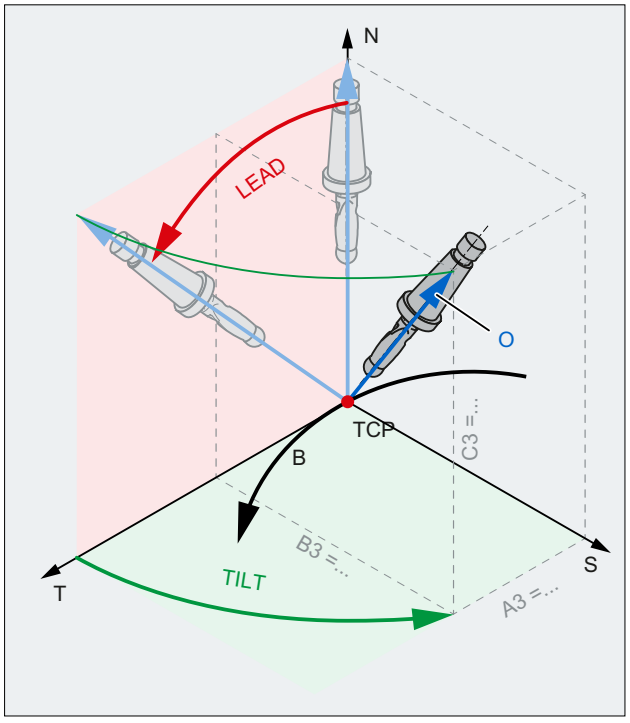
X, Y, Z	Souřadné osy WCS
A3, B3, C3	Složky směrového vektoru
O	Vektor orientace

Obrázek 4-4 Programování směrových vektorů

Programování orientace nástroje pomocí příkazů LEAD a TILT

Výsledná orientace nástroje je zjišťována z těchto údajů:

- Tečna ke dráze
- Normálový vektor plochy
  Na začátku bloku `A4, B4, C4` a na konci bloku `A5, B5, C5`
- Předozadní úhel `LEAD`
  Úhel v rovině definované tečnou ke dráze a normálovým vektorem plochy
- Úhel bočního naklopení `TILT` na konci bloku
  Úhel v rovině, která je kolmá na tečnu ke dráze vzhledem k normálovému vektoru plochy



T	Tečna ke dráze
S	Kolmo na tečnu ke dráze
N	Normálový vektor plochy
B	Dráha
TCP	Tool Center Point
O	Vektor orientace

Obrázek 4-5 Programování pomocí parametrů LEAD, TILT

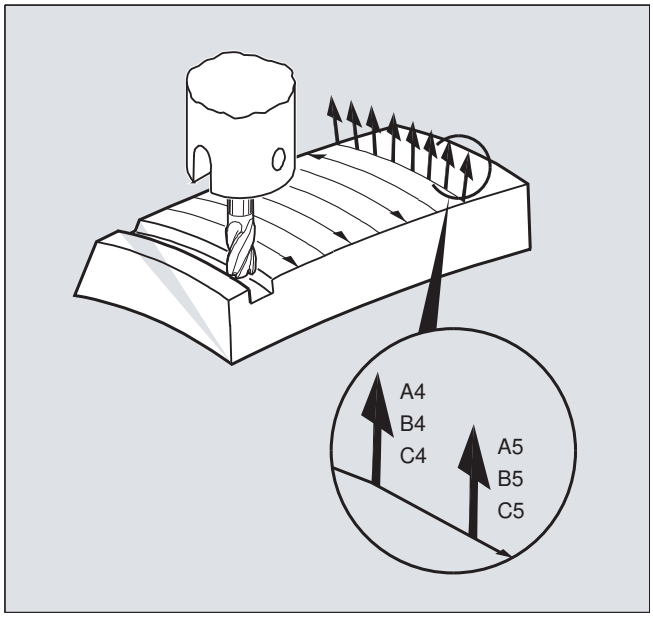
Poznámka

Chování na vnitřních rozích v případě 3D korekce nástroje

Jestliže je blok ve vnitřním rohu zkrácený, je i přesto naprogramovaná orientace nástroje nastavena na konci bloku.

4.9.2.5 Frézování na čelní ploše (A4, B4, C4, A5, B5, C5)

Frézování na čelní ploše slouží pro opracovávání libovolně zakřivených povrchů.



Pro tento druh 3D frézování potřebujete popis řádek po řádku jednotlivých 3D drah na povrchu obrobku.

Při výpočtech se obvykle uskutečňuje v systému CAM a zohledňuje se při něm tvar nástroje a rozměry nástroje. Nahotovo vypočítané NC bloky jsou potom prostřednictvím postprocesoru načteny do řídícího systému.

Programování zakřivení dráhy

Popis ploch

Pro popis zakřivení dráhy se využívá normálových vektorů plochy s následujícími složkami:

`A4, B4, C4` - počáteční vektor na začátku bloku

`A5, B5, C5` - koncový vektor na konci bloku

Jestliže se v bloku nachází pouze počáteční vektor, zůstává normálový vektor plochy po celý blok konstantní. Pokud se v bloku nachází pouze koncový vektor, uskuteční se interpolace pomocí největší kružnice koule z koncové hodnoty předcházejícího bloku do naprogramované koncové hodnoty.

Jestliže jsou naprogramovány jak počáteční, tak i koncový vektor, mezi oběma směry se rovněž uskuteční interpolace pomocí největší kružnice koule. Takto se dají vytvářet spojité hladké pohyby po dráze.

V základním nastavení ukazují normálové vektory plochy nezávisle na aktivní rovině G17 až G19 ve směru osy Z.

Délka vektoru nemá žádný význam.

Nenaprogramovaným složkám vektoru je dosazena nulová hodnota.

Když je aktivní příkaz ORIWKS (viz kapitola "Vztah orientačních os (ORIWKS, ORIMKS) (Strana 681)", jsou normálové vektory plochy vztaženy na aktivní frame a při otáčení framu se otáčejí také.

Výrobce stroje

Normálový vektor plochy musí být v rámci mezních hodnot nastavených pomocí strojních parametrů kolmý na tečnu ke dráze, jinak se aktivuje alarm.

4.9.2.6 Vztah orientačních os (ORIWKS, ORIMKS)

Při programování orientace v souřadném systému obrobku pomocí

- Eulerova úhlu nebo úhlu RPY nebo
- vektoru orientace

může být pomocí funkcí `ORIMKS`/`ORIWKS` nastaven průběh otáčivého pohybu.

Poznámka

Výrobce stroje

Druh interpolace pro orientaci je definován pomocí strojního parametru:

MD21104 $MC_ORI_IPO_WITH_G_CODE

= FALSE: Vztah určují příkazy G-funkcí ORIWKS a ORIMKS

= TRUE: Vztah určují příkazy G-funkcí z 51. Skupiny (ORIAXES, ORIVECT, ORIPLANE, ...)

Syntaxe

```
ORIMKS=...
ORIWKS=...
```

Význam

`ORIMKS`:	Otáčení v souřadném systému stroje
`ORIWKS`:	Otáčení v souřadném systému obrobku

Poznámka

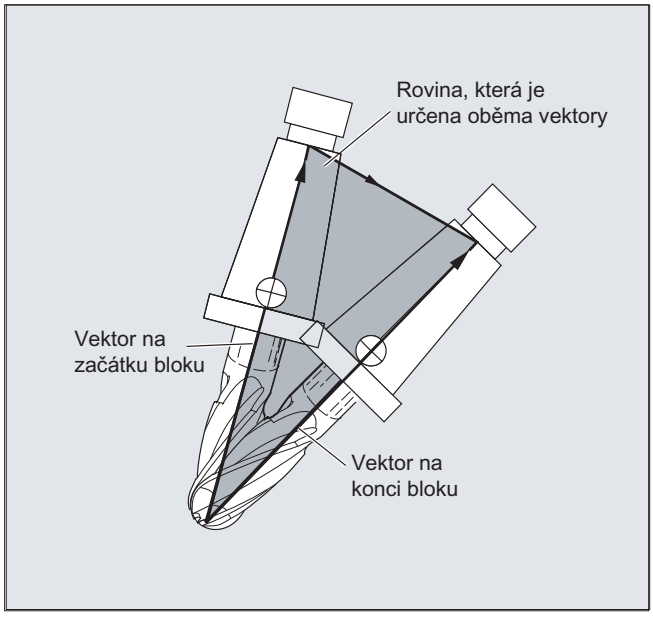
Základním nastavením je ORIWKS. Jestliže není u programu pracujícího s pěti osami předem jasné, na jakém stroji má být zpracováván, mělo by být v zásadě zvoleno ORIWKS. To, jaké pohyby stroje budou ve skutečnosti provedeny, závisí na kinematice stroje.

Pomocí `ORIMKS` mohou být programovány skutečné pohyby stroje, např. aby se zabránilo kolizi s různými zařízeními atd.

Další informace

V případě ORIMKS je prováděný pohyb nástroje **závislý** na kinematice stroje. Při změně orientace, kdy je špička nástroje na pevném místě v prostoru, se mezi dvěma polohami kruhové osy uskutečňuje lineární interpolace.

V případě ORIWKS je prováděný pohyb nástroje **nezávislý** na kinematice stroje. V případě změny orientace, při které špička nástroje zůstává pevně na jednom místě, se nástroj pohybuje v rovině definované počátečním a koncovým vektorem.



Singulární polohy

Poznámka

ORIWKS

Orientační pohyby v oblasti singulární polohy u stroje s pěti osami vyžadují velké pohyby os stroje. (Například v případě otočné naklápěcí hlavičky s osou C jako otočnou osou a osou A jako naklápěcí osou jsou singulární všechny polohy, kdy A=0.)

Výrobce stroje

Aby nebyly osy stroje příliš přetíženy, regulační systém v blízkosti singulárních poloh výrazně snižuje rychlost pohybu po dráze.

Pomocí strojních parametrů

$MC_TRAFO5_NON_POLE_LIMIT

$MC_TRAFO5_POLE_LIMIT

mohou být pro transformaci dosazeny takové parametry, aby se orientační pohyby v blízkosti pólů byly provedeny skrz pól a aby tak bylo umožněno plynulé opracování.

Singulární místa jsou ošetřena pouze strojním parametrem $MC_TRAFO5_POLE_LIMIT .

Další informace: Příručka k funkcím, Transformace

4.9.2.7 Programování orientačních os (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2)

Funkce "Orientační osy" popisuje orientaci nástroje v prostoru a daného nastavení se dosahuje naprogramováním offsetů kruhových os. Dalšího třetího stupně volnosti může být dosahováno dalším otáčením nástroje okolo sebe sama. Tyto orientace nástroje se libovolně v prostoru nastavují pomocí třetí kruhové osy a vyžadují transformaci se šesti osami. Vlastní otočení nástroje okolo sebe sama je definováno v závislosti na druhu interpolace vektorů otočení pomocí úhlu otočení THETA (viz "Otáčení orientace nástroje (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Strana 692)".

Orientační osy jsou programovány prostřednictvím identifikátorů os A2, B2, C2.

Syntaxe

```
N... ORIAXES/ORIVECT            ; Lineární interpolace nebo interpolace
                                pomocí největší kružnice koule
N... G1 X Y Z A B C

N... ORIPLANE                   ; Interpolace orientace roviny

N... ORIEULER/ORIRPY/ORIRPY2    ; Úhel orientace - Eulerův úhel / úhel RPY.
N... G1 X Y Z A2= B2= C2=       ; Programování úhlů virtuálních os

N... ORIVIRT1/ORIVIRT2          ; definice virtuálních orientačních os 1/2
N... G1 X Y Z A3= B3= C3=       ; programování směrového vektoru
```

Poznámka

Pro změny orientace podél kuželové plášťové plochy nacházející se v prostoru mohou být naprogramovány další offsety kruhových orientačních os (viz "Programování orientace podél kuželové plášťové plochy (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) (Strana 685)").

Význam

ORIAXES:	Lineární interpolace os stroje nebo orientačních os
ORIVECT:	Interpolace pomocí největší kružnice koule (identická s ORIPLANE)
ORIMKS: ORIWKS:	Otáčení v souřadném systému stroje Otáčení v souřadném systému obrobku Popis viz "Vztah orientačních os (ORIWKS, ORIMKS) (Strana 681)".
A= B= C=:	Programování poloh os stroje
ORIEULER:	Programování orientace pomocí Eulerova úhlu

ORIRPY:	Programování orientace pomocí úhlu RPY. Posloupnost při otáčení je X Y Z, přičemž platí: • A2 je úhel otočení okolo osy X • B2 je úhel otočení okolo osy Y • C2 je úhel otočení okolo osy Z
ORIRPY2:	Programování orientace pomocí úhlu RPY. Posloupnost při otáčení je Z Y X, přičemž platí: • A2 je úhel otočení okolo osy Z • B2 je úhel otočení okolo osy Y • C2 je úhel otočení okolo osy X
A2= B2= C2=:	Programování úhlů virtuálních os
ORIVIRT1/ORIVIRT2:	Programování orientace pomocí virtuálních orientačních os Definice 1: stanovení podle MD21120 $MC_ORIAX_TURN_TAB_1 Definice 2: stanovení podle MD21130 $MC_ORIAX_TURN_TAB_2
A3= B3= C3=:	Programování směrového vektoru směrové osy

Další informace

Výrobce stroje

Pomocí strojního parametru MD21102 $MC_ORI_DEF_WITH_G_CODE je stanoveno, jak jsou definovány naprogramované úhly A2, B2, C2:

Definice se uskutečňuje podle strojního parametru MD21100 $MC_ORIENTATION_IS_EULER (standardní) nebo se definice uskutečňuje podle G-funkcí ze skupiny 50 (ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2).

Pomocí strojního parametru MD21104 $MC_ORI_IPO_WITH_G_CODE se určuje, jaký druh interpolace se uplatňuje: ORIWKS/ORIMKS nebo ORIAXES/ORIVECT.

Provozní režim JOG

V tomto provozním režimu je úhel orientace interpolován vždy lineárně. Při spojitých a inkrementálních posuvech prostřednictvím tlačítek posuvů je možno pohybovat jen jednou orientační osou. Pomocí ručních koleček je možné pohybovat orientačními osami současně.

Za účelem manuálního ovládání orientačních os se používá kanálový spínač korekce (override) posuvu, příp. spínač pro korekci (override) rychlého posuvu.

Pomocí následujících strojních parametrů je možné oddělené nastavení rychlosti:

MD21160 $MC_JOG_VELO_RAPID_GEO

MD21165 $MC_JOG_VELO_GEO

MD21150 $MC_JOG_VELO_RAPID_ORI

MD21155 $MC_JOG_VELO_ORI

Poznámka

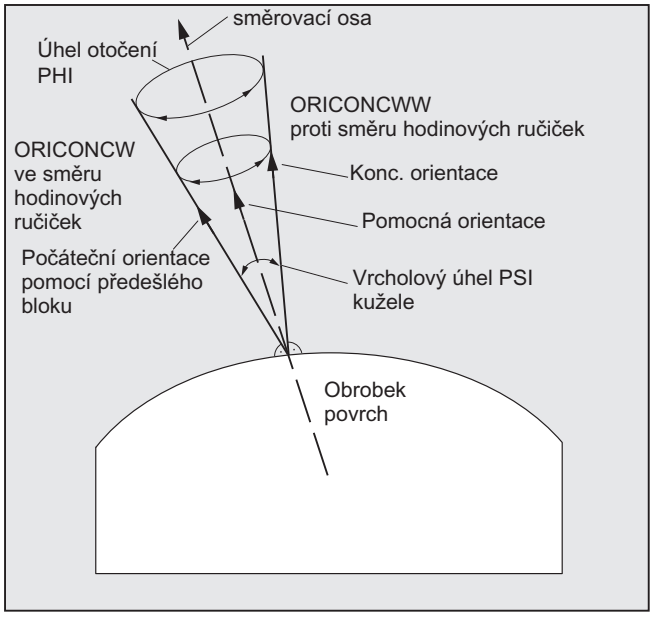
SINUMERIK 840D sl s "modulem pro práci s transformacemi"

Pomocí funkce "Kartézské manuální pohyby" je možné v provozním režimu JOG od sebe odděleně nastavovat posunutí geometrických os ve vztažných systémech MCS, WCS a TCS.

Další informace: Příručka k funkcím, Transformace

4.9.2.8 Programování orientace podél kuželové plášťové plochy (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO)

Pomocí rozšířené orientace je možné uskutečňovat změny orientace podél kuželové plášťové plochy nacházející se v prostoru. Interpolace vektorů orientace se na kuželové plášťové ploše uskutečňuje pomocí modálních příkazů ORICONxx. Pro interpolaci v rovině může být naprogramována koncová orientace ORIPLANE. Obecně je počáteční orientace určena na základě předcházejících bloků.



Programování

Koncová orientace se určuje buď prostřednictvím naprogramováním zadaného Eulerova úhlu nebo úhlu RPY pomocí proměnných A2, B2, C2 nebo naprogramováním poloh kruhových os A, B, C. Pro orientační osy podél kuželové plášťové plochy jsou zapotřebí další naprogramované údaje:

- Osa kužele jako vektor pomocí proměnných `A6, B6, C6`
- Úhel kužele `PSI` s identifikátorem `NUT`
- Pomocná orientace na plášti kužele pomocí proměnných `A7, B7, C7`

Poznámka

Programování směrového vektoru A6, B6, C6 pro osu kužele

Naprogramování koncové orientace není bezpodmínečně nutné. Jestliže koncová orientace není uvedena, potom se bude provádět interpolace na celém plášti kužele s úhlem 360 stupňů.

Programování úhlu kužele pomocí parametru NUT = úhel

Zadání koncové orientace je bezpodmínečně nutné.

Úplný plášť kužele s úhlem 360 stupňů není možné tímto způsobem interpolovat.

Programování pomocná orientace na plášti kužele pomocí proměnných A7, B7, C7

Zadání koncové orientace je bezpodmínečně nutné. Změna orientace a směr otáčení jsou jednoznačně určeny pomocí tří vektorů počáteční, koncové a pomocné orientace. Všechny tři vektory se přitom musí od sebe lišit. Jestliže je pomocná orientace rovnoběžná s počáteční nebo s koncovou orientací, potom se uskuteční lineární interpolace orientace pomocí největší kružnice koule v rovině, která je určena počátečním a koncovým vektorem.

Rozšířená interpolace orientace na kuželové plášťové ploše

```
N... ORICONCW nebo ORICONCCW
N... A6= B6= C6= A3= B3= C3=
nebo
N... ORICONTO
N... G1 X Y Z A6= B6= C6=
nebo
N... ORICONIO
N... G1 X Y Z A7= B7= C7=
N... PO[PHI]=(a2, a3, a4, a5)
N... PO[PSI]=(b2, b3, b4, b5)
```

Interpolace na plášti kuželu s těmito parametry

Směrový vektor kuželu ve/proti směru hodinových ručiček a koncová orientace nebo

tangenciální přechod a
údaj koncové orientace

nebo
údaj koncové orientace a údaj pomocné orientace na kuželovém plášti pomocí

polynomu pro úhel otočení a
polynomu pro vrcholový úhel

Parametr

`ORIPLANE`:	Interpolace v rovině (interpolace pomocí největší kružnice koule)
`ORICONCW`:	Interpolace po ploše pláště kužele ve směru hodinových ručiček
`ORICONCCW`:	Interpolace po ploše pláště kužele proti směru hodinových ručiček
`ORICONTO`:	Interpolace na ploše pláště kuželu s tangenciálním přechodem

A6= B6= C6=:	Programování osy kužele (normovaný vektor)
NUT=úhel:	Úhel kužele ve stupních
NUT=+179:	Úhel pohybu menší nebo roven 180 stupňům
NUT=-181:	Úhel pohybu větší nebo roven 180 stupňům
ORICONIO:	Interpolace na kuželové plášťové ploše
A7= B7= C7=:	Pomocná orientace (programuje se jako normovaný vektor)
PHI:	Úhel otočení orientace okolo směrové osy kužele
PSI:	Úhel kužele
možné polynomy PO[PHI]=(a2, a3, a4, a5) PO[PSI]=(b2, b3, b4, b5)	Kromě příslušných úhlů mohou být naprogramovány také polynomy maximálně 5. stupně

Příklad: Různé změny orientace

Programový kód	**Komentář**
…	
N10 G1 X0 Y0 F5000	
N20 TRAORI(1)	; aktivování transformace orientace
N30 ORIVECT	; interpolace orientace nástroje jako vektor
…	; orientace nástroje v rovině
N40 ORIPLANE	; aktivování interpolace pomocí největší kružnice koule
N50 A3=0 B3=0 C3=1	
N60 A3=0 B3=1 C3=1	; orientace v rovině Y/Z otočená o 45 stupňů, na konci bloku bude dosaženo orientace (0,1/√2,1/√2)
…	
N70 ORICONCW	; programování orientace na plášti kužele:
N80 A6=0 B6=0 C6=1 A3=0 B3=0 C3=1	; interpolace vektoru orientace na plášti kužele se směrem (0,0,1) až do orientace (1/√2,0,1/√2) ve směru hodinových ručiček, úhel přitom činí 270 stupňů
N90 A6=0 B6=0 C6=1	; orientace nástroje proběhne celou jednu otáčku po stejném plášti kuželu

Další informace

Jestliže mají být popsány změny orientace na kuželové plášťové ploše nacházející se libovolně v prostoru, potom musí být znám vektor, okolo kterého se má orientace nástroje otáčet. Kromě toho musí být zadána počáteční a koncová orientace. Počáteční orientace vyplývá z předcházejícího bloku a koncová orientace musí být buď naprogramována nebo musí být stanovena na základě dalších podmínek.

Programování v rovině ORIPLANE odpovídá funkci ORIVECT

Naprogramování interpolace pomocí největší kružnice koule spolu s polynomy popisujícími úhel odpovídá lineární a polynomické interpolaci kontur. Orientace nástroje je interpolována v

rovině, která je definována počáteční a koncovou orientací. Pokud je kromě toho naprogramován polynom, může se vektor orientace naklápět i mimo tuto rovinu.

Programování kruhových drah v rovině pomocí příkazů G2/G3, CIP a CT

Rozšířená orientace odpovídá interpolaci kruhových oblouků v rovině. Pokud budete potřebovat odpovídající informace o možnostech programování kruhových oblouků pomocí středu nebo rádiusu (G2/G3), pomocí vnitřního bodu CIP a kruhových oblouků s tangenciálním napojením, viz kapitola „Kruhová interpolace (Strana 190)".

Programování orientace

Interpolace vektoru orientace na kuželové plášťové ploše (ORICONxx)

Pro interpolaci orientace na kuželové plášťové ploše mohou být používány čtyři různé druhy interpolace ze skupiny příkazů G-funkcí č. 51.

1. Interpolace na plášti kužele ve směru hodinových ručiček ORICONCW se zadáním koncové orientace a směru kužele nebo úhlu kužele. Směrový vektor se programuje pomocí identifikátorů A6, B6, C6 a vrcholový úhel kužele se programuje pomocí identifikátoru NUT= s intervalem hodnot 0 až 180 stupňů.
2. Interpolace na plášti kužele proti směru hodinových ručiček ORICONCWW se zadáním koncové orientace a směru kužele nebo úhlu kužele. Směrový vektor se programuje pomocí identifikátorů A6, B6, C6 a vrcholový úhel kužele se programuje pomocí identifikátoru NUT= s intervalem hodnot 0 až 180 stupňů.
3. Interpolace na plášti kužele pomocí příkazu ORICONIO se zadáním koncové orientace a pomocné orientace, která se programuje pomocí identifikátorů A7, B7, C7.
4. Interpolace po ploše pláště kužele pomocí příkazu ORICONTO s tangenciálním přechodem a se zadáním koncové orientace. Směrový vektor se programuje pomocí identifikátorů A6, B6, C6.

4.9.2.9 Zadání orientace pomocí druhého styčného bodu (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=)

Programování změny orientace pomocí druhé prostorové křivky ORICURVE

Další možnost, jak naprogramovat změny orientace, spočítá v tom, že se kromě pohybu špičky nástroje podél prostorové křivky naprogramuje pomocí příkazu ORICURVE také pohyb druhého styčného bodu nástroje. Tímto způsobem mohou být jednoznačně definovány změny orientace nástroje stejně jako při naprogramování vektoru nástroje samotného.

Výrobce stroje

Věnujte prosím pozornost pokynům výrobce stroje, které se týkají identifikátorů osy, které jsou nastavitelné pomocí strojních parametrů a které slouží pro programování 2. dráhy pro popis orientace nástroje.

Programování

U tohoto druhu interpolace mohou být pro obě prostorové křivky naprogramovány body pomocí příkazů G1, příp. polynomy pomocí příkazu POLY. Kruhové oblouky a evolventy jsou nepřípustné. Kromě toho mohou být aktivovány splinová interpolace pomocí příkazu BSPLINE a funkce "Shrnutí krátkých splinových bloků"

Pokud budete potřebovat další informace týkající se funkce „Shrnutí krátkých splinových bloků“, viz Příručka k funkcím, Základní funkce.

Použití jiných druhů splinů ASPLINE a CSPLINE, stejně jako aktivování kompresoru pomocí příkazů COMP..., není přípustné.

Souřadnice pro pohyb dvou styčných bodů nástroje může být zadán naprogramováním polynomu orientace až maximálně 5. stupně.

Rozšířená interpolace orientace pomocí doplňkové prostorové křivky a polynomu pro souřadnice

```
N... ORICURVE                                    ; Zadání pohybu druhého styčného bodu nástroje a doplňkový polynom příslušné souřadnice
N... PO[XH]=(xe, x2, x3, x4, x5)
N... PO[YH]=(ye, y2, y3, y4, y5)
N... PO[ZH]=(ze, z2, z3, z4, z5)
```

Parametr

`ORICURVE`	Interpolace orientace s udáním pohybu dvou styčných bodů nástroje.
`XH YH ZH`	Identifikátory souřadnic druhého kontaktního bodu nástroje, které popisují doplňkovou konturu jako prostorovou křivku

`PO[XH]=(xe, x2, x3, x4, x5)`
`PO[YH]=(ye, y2, y3, y4, y5)`
`PO[ZH]=(ze, z2, z3, z4, z5)`
	Možné polynomy
	Kromě odpovídajících koncových bodů mohou být prostorové křivky naprogramovány také pomocí polynomů.

`xe, ye, ze`	Koncové body prostorové křivky
`xi, yi, zi`	Koeficienty polynomu maximálně 5. stupně

Poznámka

Identifikátory XH YH ZH pro programování 2. dráhy orientace

Tyto identifikátory musí být zvoleny tak, aby nemohlo dojít ke konfliktu s jinými identifikátory lineárních os

X Y Z - osy

a kruhovými osami, jako např.

A2 B2 C2 - Eulerův úhel, příp. úhel RPY

A3 B3 C3 - směrové vektory

A4 B4 C4 příp. A5 B5 C5 - normálové vektory plochy

A6 B6 C6 - vektory otáčení, příp. A7 B7 C7 souřadnice vnitřního bodu

nebo jinými interpolačními parametry.

4.9.3 Polynomický popis orientace (PO[úhel], PO[souřadnice])

Nezávisle na tom, která polynomická interpolace ze skupiny příkazů G-funkcí č.1 je momentálně aktivní, mohou být při tří- až pětiosé transformaci naprogramovány dva různé typy polynomů pro popis orientace až maximálně 5. stupně.

1. Polynom pro **úhel**: Předozadní úhel LEAD, boční úhel TILT
   vztažený na rovinu, která je definována počáteční a koncovou orientací.
2. Polynom pro **souřadnice**: XH, YH, ZH druhé prostorové křivky pro orientaci nástroje pomocí vztažného bodu na nástroji.

V případě transformace se šesti osami může být pro účely orientace nástroje naprogramováno navíc ještě i otáčení vektoru otočení THT pomocí polynomu maximálně 5. stupně pro otáčení nástroje okolo sebe sama.

Syntaxe

Polynom orientace pro typ 1 pro **úhel**

```
N... PO[PHI]=(a2, a3, a4, a5)
N... PO[PSI]=(b2, b3, b4, b5)
```

Tří- až pětiosá transformace

Polynom orientace pro typ 2 pro **souřadnice**

```
N... PO[XH]=(xe, x2, x3, x4, x5)
N... PO[YH]=(ye, y2, y3, y4, y5)
N... PO[ZH]=(ze, z2, z3, z4, z5)
```

Identifikátor pro souřadnice druhé dráhy orientace pro orientaci nástroje

Navíc může být v obou případech naprogramován polynom pro **otáčení** v případě transformace se šesti osami pomocí

`N... PO[THT]=(c2, c3, c4, c5)`	interpolace otáčení vztažená k dráze
nebo	absolutní, relativní a tangenciální interpolace pro změnu orientace
`N... PO[THT]=(d2, d3, d4, d5)`	

vektoru orientace. To je možné tehdy, pokud je podporována transformace vektoru otočení s offsetem, který může být naprogramován a interpolován pomocí úhlu otočení THETA.

Význam

`PO[PHI]`	Úhel v rovině mezi počáteční a koncovou orientací
`PO[PSI]`	Popisuje úhel naklopení orientace ven z roviny mezi počáteční a koncovou orientací
`PO[THT]`	Úhel otočení, který vzniká otočením vektoru naprogramovaného pomocí příkazu G-funkce THETA ze skupiny 54
`PHI`	Předozadní úhel LEAD
`PSI`	Úhel bočního naklopení TILT
`THETA`	Otočení okolo směru nástroje v ose Z
`PO[XH]`	Souřadnice X vztažného bodu na nástroji
`PO[YH]`	Souřadnice Y vztažného bodu na nástroji
`PO[ZH]`	Souřadnice Z vztažného bodu na nástroji

Další informace

Polynomy orientace nemusí být naprogramovány,

- pokud jsou aktivní splinové interpolace ASPLINE, BSPLINE, CSPLINE.
  Polynomy typu 1 pro úhel orientace jsou možné pro každý druh interpolace kromě splinů, tzn. při lineární interpolaci s rychlým posuvem G00, příp. s pracovním posuvem G01,
  při polynomické interpolaci POLY a
  při kruhové, příp. evolventní interpolaci s příkazy G02, G03, CIP, CT, INVCW a INCCCW.

  Polynom typu 2 pro souřadnice orientace jsou oproti tomu možné jen tehdy, pokud jsou aktivní
  lineární interpolace s rychlým posuvem G00, příp. s pracovním posuvem G01 nebo polynomická interpolace POLY.
- jestliže je orientace interpolována pomocí osové interpolace ORIAXES. V tomto případě mohou být přímo pomocí příkazů PO[A] a PO[B] naprogramovány polynomy pro orientační osy A a B.

Polynomy orientace typu 1 s příkazy ORIVECT, ORIPLANE a ORICONxx

V případě interpolace pomocí největší kružnice koule a interpolace plochy pláště kužele pomocí příkazů ORIVECT, ORIPLANE a ORICONxx je možné používat jedině polynomy orientace typu 1.

Polynom orientace pro typ 2 s příkazem ORICURVE

Jestliže je aktivní interpolace s doplňkovou prostorovou křivkou ORICURVE, jsou interpolovány kartézské složky vektoru orientace a tato operace je možná pouze s polynomem orientace typu 2.

4.9.4 Otáčení orientace nástroje (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA)

Jestliže má u typů obráběcích strojů s pohyblivým nástrojem existovat možnost měnit orientaci nástroje, potom se v každém bloku programuje koncová orientace. V závislosti na kinematice stroje mohou být naprogramovány buď směr orientace orientačních os nebo směr otáčení vektoru orientace THETA. Pro tyto vektory otočení mohou být naprogramovány různé druhy interpolace:

- ORIROTA: Úhel otočení vztažen ke směru otáčení zadanému absolutně.
- ORIROTR: Úhel otočení relativně vůči rovině mezi počáteční a koncovou orientací.
- ORIROTT: Úhel otočení relativně vůči změně vektoru orientace.
- ORIROTC: Tangenciální úhel otočení k tečně dráhy.

Syntaxe

Jen tehdy, pokud je aktivován druh interpolace ORIROTA, může být úhel otočení nebo vektor otočení naprogramován jedním z následujících čtyř možných způsobů:

1. Přímé polohování kruhových os `A, B, C`
2. Eulerův úhel (ve stupních) pomocí `A2, B2, C2`
3. Úhel RPY (ve stupních) pomocí `A2, B2, C2`
4. Směrový vektor pomocí `A3, B3, C3` (úhel otočení pomocí `THETA=<hodnota>`)

Pokud jsou aktivní funkce ORIOTR nebo ORIOTT, může být úhel otočení naprogramován ještě i přímo pomocí příkazu `THETA`.

Otočení může být naprogramováno také samotně v bloku, aniž by se uskutečňovala změna orientace. Příkazy ORIROTR a ORIROTT přitom nemají žádný význam. V tomto případě je úhel otočení vždy interpretován ve vztahu k absolutnímu směru (ORIROTA).

`N... ORIROTA`	Definice interpolace vektoru otočení
`N... ORIROTR`	
`N... ORIROTT`	
`N... ORIROTC`	
`N... A3= B3= C3= THETA=<hodnota>`	Definice otáčení vektoru orientace
`N... PO[THT]=(`d_2`, `d_3`, `d_4`, `d_5`)`	Interpolace úhlu otočení pomocí polynomu 5. stupně

Význam

ORIROTA:	Úhel otočení vztažen ke směru otáčení zadanému absolutně
ORIROTR:	Úhel otočení relativně vůči rovině mezi počáteční a koncovou orientací
ORIROTT:	Úhel otočení jako tangenciální vektor otočení ke změně orientace
ORIROTC:	Úhel otočení jako tangenciální vektor otočení vůči tečně ke dráze
THETA:	Otočení vektoru orientace
THETA=<hodnota>:	Úhel otočení ve stupních, který bude dosažen na konci bloku
THETA=Θe:	Úhel otočení s Eulerovým úhlem Θ_e tohoto vektoru
THETA=AC(...):	Blokové přepnutí na zadávání absolutních rozměrů
THETA=AC(...):	Blokové přepnutí na zadávání inkrementálních rozměrů
Θe:	Koncový úhel vektoru otočení buď jako absolutní rozměr, když je aktivní G90, nebo jako relativní rozměr (inkrementální), když je aktivní G91
PO[THT]=(....):	Polynom pro úhel otočení

Příklad: Otáčení orientace

Programový kód	**Komentář**
N10 TRAORI	; aktivování transformace orientace
N20 G1 X0 Y0 Z0 F5000	; orientace nástroje
N30 A3=0 B3=0 C3=1 THETA=0	; ve směru Z s úhlem otočení 0
N40 A3=1 B3=0 C3=0 THETA=90	; ve směru X a otočení o 90 stupňů
N50 A3=0 B3=1 C3=0 PO[THT]=(180,90)	; orientace
N60 A3=0 B3=1 C3=0 THETA=IC(-90)	; ve směru Y a otočení do polohy 180 stupňů
N70 ORIROTT	; zůstává konstantní a otočení do polohy 90 stupňů
N80 A3=1 B3=0 C3=0 THETA=30	; úhel otočení relativně vůči změně orientace
	; vektor otočení pod úhlem 30 stupňů vůči rovině X-Y

Při interpolaci bloku N40 se úhel otočení lineárně interpoluje od počáteční hodnoty 0 stupňů do koncové hodnoty 90 stupňů. V bloku N50 se mění úhel otočení o 90 stupňů do polohy 180 stupňů podle paraboly θ(u) = +90u2. V bloku N60 může být uvedeno také otočení, aniž by se konala nějaká změna orientace.

V bloku N80 se orientace nástroje otáčí ze směru Y do směru X. Změna orientace přitom leží v rovině X-Y a vektor otočení svírá s touto rovinou úhel 30 stupňů.

Další informace

ORIROTA

Úhel otočení THETA bude interpolován vzhledem k absolutně definovanému směru v prostoru. Základní směr otočení se zadává pomocí strojních parametrů.

ORIROTR

Úhel otočení THETA je interpretován vzhledem k rovině, která je definována počáteční a koncovou orientací.

ORIROTT

Úhel otočení THETA je interpretován vzhledem ke změně orientace- Pro případ THETA=0 je vektor otočení interpolován tangenciálně vůči změněn orientace a od situace s příkazem ORIROTR se liší jen tehdy, pokud byl pro orientaci naprogramován minimálně jeden polynom pro "Úhel naklopení PSI". Z toho pak vyplývá změna orientace, která neprobíhá v dané rovině. Prostřednictvím dále naprogramovaného úhlu otočení THETA může být potom interpolován např. vektor otočení tak, že neustále zaujímá určitou hodnotu vůči změně orientace.

ORIROTC

Vektor otočení je interpolován vzhledem k tečně ke dráze s offsetem naprogramovaným pomocí úhlu THETA. Pro úhel offsetu přitom může být naprogramovaný také polynom PO[THT]=(c2, c3, c4, c5) maximálně 5. stupně.

4.9.5 Orientace vzhledem k dráze

4.9.5.1 Druhy orientace vztažené k dráze

Pomocí této rozšiřovací funkce je dosahováno orientace vztažené nejen na konec bloku, nýbrž po celé dráze. Orientace dosažená v předcházejícím bloku se pomocí interpolace pomocí největší kružnice koule převádí do naprogramované koncové orientace. V zásadě existují dvě možnosti, jak naprogramovat požadovanou orientaci vzhledem k dráze:

1. Interpolace orientace nástroje také jako otáčení nástroje pomocí příkazů ORIPATH, ORPATHTS vzhledem k dráze.
2. Vektor orientace je naprogramován a interpolován stejně jako dříve. Pomocí příkazu ORIROTC se nastavuje otočení vektoru orientace vzhledem k tečně ke dráze.

Syntaxe

Druh interpolace orientace a otočení nástroje se programují pomocí těchto příkazů:

N... ORIPATH	Orientace vzhledem k dráze
N... ORIPATHS	Orientace vzhledem ke dráze s vyhlazováním charakteristiky orientace
N... ORIROTC	Interpolace vektoru otočení vzhledem ke dráze

Skoková změna orientace vyvolaná místem, kde se na dráze nachází roh, může být vyhlazena pomocí příkazu ORIPATHS. Směr a délka dráhy pohybu při pozvednutí nástroje se programuje pomocí vektoru se složkami A8 = X, B8 = Y a C8 = Z.

Pomocí příkazů ORIPATH/ORIPATHS mohou být naprogramovány různé vztahy vůči tečně ke dráze prostřednictvím tří úhlů

- LEAD = zadání předozadního úhlu vztaženého na dráhu a na povrch
- TILT = zadání úhlu bočního naklopení vztaženého na dráhu a na povrch
- THETA = úhel otočení

pro celou dráhu. Pro úhel otočení THETA je možné pomocí příkazu PO[THT] = (...) naprogramovat navíc ještě i polynom maximálně 5. stupně.

Poznámka

Výrobce stroje

Věnujte prosím pozornost informacím od výrobce stroje. Prostřednictvím konfigurace strojních a nastavovaných parametrů mohou být realizována ještě i další nastavení týkající se druhu orientace vztaženého k dráze.

Další informace: Příručka k funkcím, Transformace

Význam

Interpolace úhlů LEAD a TILT může být pomocí strojního parametru nastavena různě:

- Vztažný směr orientace nástroje naprogramovaný pomocí příkazů LEAD a TILT zůstává po celou dobu trvání bloku zachován.
- Předozadní úhel LEAD: Otočení okolo směru, který je kolmý na tečnu a normálový vektor TILT: Otočení orientace okolo normálového vektoru.
- Předozadní úhel LEAD: Otočení okolo směru, který je kolmý na tečnu a normálový vektor, úhel bočního naklopení TILT: Otočení orientace okolo směru tečny ke dráze.
- Úhel otočení THETA: Otočení nástroje okolo sebe sama s doplňkovou třetí kruhovou osou, která představuje orientační osu u transformace se šesti osami.

Poznámka

Nastavování orientace vztažené ke dráze je při použití příkazů OSC, OSS, OSSE, OSD, OST nepřípustné

Interpolaci orientace vztahující se ke dráze pomocí příkazů ORIPATH, příp. ORIPATHS a ORIOTC není možné naprogramovat spolu s vyhlazením průběhu orientace pomocí příkazů G-funkce ze skupiny 34. Pro tento účel musí být aktivován příkaz OSOF.

4.9.5.2 Otáčení orientace nástroje vzhledem k dráze (ORIPATH, ORIPATHS, úhel otočení)

V případě transformace se šesti osami může být orientace nástroje nastavována libovolně v prostoru a kromě toho se nástroj může pomocí třetí kruhové osy otáčet okolo sebe sama. V případě otáčení orientace nástroje pomocí příkazů ORIPATH, příp. ORIPATHS vztaženého ke dráze může být doplňkové otočení naprogramováno pomocí úhlu otočení THETA. Kromě toho existuje možnost naprogramovat pomocí příkazů LEAD a TILT vektor, který leží v rovině kolmé na směr nástroje.

Výrobce stroje

Věnujte prosím pozornost informacím od výrobce stroje. Pomocí strojního parametru může být interpolace úhlů LEAD a TILT nastavena odlišně.

Syntaxe

Otáčení orientace nástroje a nástroje samotného

Druh orientace nástroje vzhledem ke dráze se aktivuje pomocí příkazu ORIPATH nebo ORIPATHS.

`N... ORIPATH`	Aktivování druhu orientace vztaženého na dráhu
`N... ORIPATHS`	Aktivování druhu orientace vztaženého na dráhu s vyhlazením průběhu orientace

Aktivování tří možných úhlů způsobujících následující otočení:

`N... LEAD=`	Úhel pro naprogramovanou orientaci vzhledem k normálovému vektoru plochy
`N... TILT=`	Úhel pro naprogramovanou orientaci v rovině, která je kolmá na tečnu ke dráze vzhledem k normálovému vektoru plochy
`N... THETA=`	Úhel otočení třetí kruhové osy vzhledem ke změně orientace okolo směru nástroje

Hodnota úhlu na konci bloku se naprogramuje pomocí příkazů `LEAD=hodnota`, `TILT=hodnota`, příp. `THETA=hodnota`. Kromě konstantních úhlů mohou být pro všechny tři úhly naprogramovány polynomy maximálně 5. stupně.

`N... PO[PHI]=(a2, a3, a4, a5)`	Polynom pro předozadní úhel LEAD
`N... PO[PSI]=(b2, b3, b4, b5)`	Polynom pro úhel bočního naklopení TILT
`N... PO[THT]=(d2, d3, d4, d5)`	Polynom pro úhel otočení THETA

Při programování mohou být vypuštěny vyšší koeficienty polynomu, které jsou nulové. Příklad: `PO[PHI]=a2` udává pro předozadní úhel `LEAD` parabolu.

Význam

Orientace nástroje vztahující se ke dráze

`ORIPATH`:	Orientace nástroje vztažená na dráhu
`ORIPATHS`:	Orientace nástroje vztažená na dráhu, zlom v průběhu orientace se vyhladí
`LEAD`:	Úhel je vztažen k normálovému vektoru plochy, a svírají jej tečna dráhy a normálový vektor plochy v dané rovině
`TILT`:	Otočení orientace okolo směru Z, příp. otočení okolo tečny ke dráze
`THETA`:	Otočení okolo směru nástroje (Z)
`PO[PHI]`:	Polynom orientace pro předozadní úhel LEAD
`PO[PSI]`:	Polynom orientace pro úhel bočního naklopení TILT
`PO[THT]`:	Polynom orientace pro úhel otočení THETA

Poznámka

Úhel otočení THETA

Pro otočení nástroje okolo sebe sama pomocí třetí kruhové osy jako orientační osy je zapotřebí transformace se šesti osami.

4.9.5.3 Interpolace otočení nástroje vzhledem k dráze (ORIROTC, THETA)

Interpolace s vektory otočení

K otočení nástroje vzhledem k tečně ke dráze naprogramovanému pomocí příkazu ORIROTC může být vektor otočení interpolován s offsetem, který je možno naprogramovat pomocí úhlu otočení THETA. Pro offsetový úhel přitom může být pomocí příkazu PO[THT] naprogramován polynom maximálně 5. stupně.

Syntaxe

`N... ORIROTC`	Nastavení otočení nástroje vzhledem k tečně ke dráze
`N... A3= B3= C3= THETA=hodnota`	Definice otáčení vektoru orientace
`N... A3= B3= C3= PO[THT]=(c2, c3, c4, c5)`	Interpolace úhlu offsetu pomocí polynomu 5. stupně

Otočení může být naprogramováno také samotně v bloku, aniž by se uskutečňovala změna orientace.

Význam

Interpolace otočení nástroje vztažená k dráze v případě transformace se šesti osami

`ORIROTC`:	Nastavení tangenciálního vektoru otočení k tečně dráhy
`THETA=hodnota`:	Úhel otočení ve stupních, který bude dosažen na konci bloku
`THETA=θe`:	Úhel otočení s Eulerovým úhlem Θ_e tohoto vektoru
`THETA=AC(...)`:	Blokové přepnutí na zadávání absolutních rozměrů
`THETA=IC(...)`:	Blokové přepnutí na zadávání inkrementálních rozměrů
`PO[THT]=(c2, c3, c4, c5)`:	Interpolace offsetového úhlu pomocí polynomu 5. stupně

Poznámka

Interpolace vektoru otočení ORIROTC

Jestliže má být vedle směru orientace nástroje nastavováno také otočení nástroje vzhledem k tečně ke dráze, potom je pro tuto operaci nutné používat transformaci se šesti osami.

Když je aktivní příkaz ORIROTC

Vektor otočení ORIROTA nemůže být naprogramován. V případě naprogramování se aktivuje alarm 14128 "Absolutní programování otočení nástroje při aktivním příkazu ORIROTC".

Směr orientace nástroje v případě tří- až pětiosé transformace

Směr orientace nástroje může být naprogramován stejně, jak je obvyklé v případě tří- až pětiosé transformace, pomocí Eulerova úhlu, příp. pomocí úhlu RPY nebo pomocí směrových vektorů. Jsou možné také změny orientace nástroje v prostoru prostřednictvím naprogramování interpolace pomocí největší kružnice koule s příkazem ORIVECT, lineární interpolace orientačních os ORIAXES, všech interpolací na kuželové plášťové ploše ORICONxx, jakož i interpolace pomocí dvou styčných bodů nástroje pomocí příkazu ORICURVE s další prostorovou křivkou.

`G....:`	Zadání druhu pohybu kruhových os
`X, Y, Z:`	Zadání pro lineární osy
`ORIAXES:`	Lineární interpolace os stroje nebo orientačních os
`ORIVECT:`	Interpolace pomocí největší kružnice koule (identická s ORIPLANE)
`ORIMKS:` `ORIWKS:`	Otáčení v souřadném systému stroje Otáčení v souřadném systému obrobku Popis viz kapitola Otáčení orientace nástroje
`A= B= C=:`	Programování poloh os stroje
`ORIEULER:`	Programování orientace pomocí Eulerova úhlu
`ORIRPY:`	Programování orientace pomocí úhlu RPY.
`A2= B2= C2=:`	Programování úhlů virtuálních os
`ORIVIRT1:` `ORIVIRT2:`	Programování orientace pomocí virtuálních orientačních os (Definice 1), stanovení podle MD $MC_ORIAX_TURN_TAB_1 (Definice 2), stanovení podle MD $MC_ORIAX_TURN_TAB_2
`A3= B3= C3=:`	Programování směrového vektoru směrové osy
`ORIPLANE:`	Interpolace v rovině (interpolace pomocí největší kružnice koule)
`ORICONCW:`	Interpolace po ploše pláště kužele ve směru hodinových ručiček
`ORICONCCW:`	Interpolace po ploše pláště kužele proti směru hodinových ručiček
`ORICONTO:`	Interpolace na ploše pláště kuželu s tangenciálním přechodem
`A6= B6= C6=:`	Programování osy kužele (normovaný vektor)
`NUT=úhel`	Úhel kužele ve stupních
`NUT=+179`	Úhel pohybu menší nebo roven 180 stupňům
`NUT=-181`	Úhel pohybu větší nebo roven 180 stupňům
`ORICONIO:`	Interpolace na kuželové plášťové ploše

`A7= B7= C7=`:	Pomocná orientace (programuje se jako normovaný vektor)
`ORICURVE` `XH YH ZH z.B. s polynomem PO[XH]=(xe, x2, x3, x4, x5)`	Interpolace orientace s udáním pohybu dvou styčných bodů nástroje. Kromě odpovídajících koncových bodů mohou být další prostorové křivky naprogramovány pomocí polynomu.

Poznámka

Pokud je orientace nástroje s aktivním příkazem ORIAXES interpolována pro orientační osy, potom je nastavení úhlu otočení vztažené ke dráze splněno jen na konci bloku.

4.9.5.4 Vyhlazení průběhu orientace (ORIPATHS A8=, B8=, C8=)

V případě změn orientace s konstantním zrychlením na kontuře jsou přerušení pohybu po dráze, která se mohou vyskytnout zejména na rozích kontury, nežádoucí. Takto vznikající skoková změna na průběhu orientace může být vyhlazena vložením samostatného pomocného bloku. Jestliže je v průběhu změny orientace aktivní také příkaz ORIPATHS, změna orientace se potom uskutečňuje s konstantním zrychlením. V této fázi může být uskutečněno pozvednutí nástroje.

Výrobce stroje

Věnujte prosím pozornost informacím od výrobce stroje, pokud jde o případně se vyskytující předem definované strojní a nastavované parametry, s nimiž se tato funkce aktivuje.

Prostřednictvím strojního parametru může být nastaveno, jak je interpretován vektor pozvednutí.

1. V souřadném systému nástroje je souřadnice Z definována prostřednictvím směru nástroje.
2. V souřadném systému obrobku je souřadnice Z definována prostřednictvím aktivní roviny.

Pokud budete potřebovat další informace týkající se funkce „Orientace vzhledem k dráze“, viz Příručka k funkcím, Transformace.

Syntaxe

Aby byly orientace nástroje vztažené na celkovou dráhu spojité, jsou zapotřebí v rozích kontury další programové údaje. Směr a délka dráhy tohoto pohybu se programuje pomocí vektoru se složkami A8 = X, B8 = Y a C8 = Z:

```
N... ORIPATHS A8=X B8=Y C8=Z
```

Význam

`ORIPATHS`:	Orientace nástroje vztažená na dráhu, zlom v průběhu orientace se vyhladí.
`A8= B8= C8=`:	Vektorové složky pro směr a délku dráhy
`X, Y, Z`:	Pohyb pro pozvednutí ve směru nástroje

Poznámka

Programování směrového vektoru A8, B8, C8

Pokud je délka tohoto vektoru rovna nule, žádné pozvednutí nástroje se neuskutečňuje.

ORIPATHS

Orientace nástroje vztažená ke dráze se aktivuje pomocí příkazu ORIPATHS. Jinak je orientace interpolována od počáteční do koncové orientace prostřednictvím lineární interpolace pomocí největší kružnice koule.

4.9.6 Komprimování orientace (COMPON, COMPCURV, COMPCAD, COMPSURF)

NC-programy, ve kterých je aktivní transformace orientace (TRAORI) a orientace nástroje (jakéhokoli druhu) jsou naprogramovány, mohou být komprimovány, přičemž jsou dodrženy předem nastavené tolerance.

Programování

Orientace nástroje

Jestliže je aktivní transformace orientace (TRAORI), může být u strojů s 5 osami programována orientace nástroje následujícím způsobem (nezávisle na kinematice):

- Programování směrových **vektorů** prostřednictvím příkazů:
  `A3=<...> B3=<...> C3=<...>`
- Programování **Eulerova úhlu**, příp. **úhlu RPY** prostřednictvím příkazů:
  `A2=<...> B2=<...> C2=<...>`

Otočení nástroje

U strojů se **6 osami** je možné navíc ještě k orientaci nástroje naprogramovat také otočení nástroje.

Pro programování otočení nástroje se používá příkaz:

`THETA=<...>`

Viz „ Otáčení orientace nástroje (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Strana 692) ".

Poznámka

NC bloky, ve kterých je naprogramováno ještě i otočení, mohou být komprimovány jen tehdy, pokud se úhel otočení mění **lineárně**. Tzn. pro úhel otočení nesmí být naprogramován žádný polynom PO[THT] = (...).

Všeobecná forma komprimovatelných NC bloků

Z těchto důvodů mohou mít komprimovatelné NC bloky následující obecnou formu:

`N... X=<...> Y=<...> Z=<...> A3=<...> B3=<...> C3=<...> THETA=<...> F=<...>`

nebo

```
N... X=<...> Y=<...> Z=<...> A2=<...> B2=<...> C2=<...> THETA=<...> F=<...>
```

Poznámka

Hodnoty polohy mohou být zadávány buď přímo (např. X90) nebo nepřímo pomocí přiřazení parametrů (např. X=R1*(R2+R3)).

Programování orientace nástroje pomocí poloh kruhových os

Orientace nástroje může být zadávána také pomocí poloh kruhových os, např. ve formě:

```
N... X=<...> Y=<...> Z=<...> A=<...> B=<...> C=<...> THETA=<...> F=<...>
```

V tomto případě se komprimování uskutečňuje dvěma různými způsoby v závislosti na tom, zda se provádí interpolace pomocí největší kružnice koule nebo neprovádí. Jestliže se interpolace pomocí největší kružnice koule nekoná, potom je komprimovaná změna orientace reprezentována obvyklým způsobem pomocí axiálních polynomů pro kruhové osy.

Přesnost kontury

V závislosti na nastaveném režimu komprese (MD20482 $MC_COMPRESSOR_MODE) jsou pro geometrické osy a orientační osy při kompresi uplatňovány buď v konfiguraci nastavené tolerance pro specifické osy (MD33100 $MA_COMPRESS_POS_TOL) nebo následující kanálové tolerance nastavitelné pomocí strojních parametrů:

SD42475 $SC_COMPRESS_CONTUR_TOL (maximální odchylka od kontury)

SD42476 $SC_COMPRESS_ORI_TOL (maximální úhlová odchylka pro orientaci nástroje)

SD42477 $SC_COMPRESS_ORI_ROT_TOL (maximální úhlová odchylka pro úhel otočení nástroje (k dispozici jen u strojů se 6 osami))

Aktivování / deaktivování

Funkce kompresoru se aktivují promocí příkazů G-funkcí s modální platností COMPON, COMPCURV, COMPCAD, příp. COMPSURF.

Funkce kompresoru se příkazem COMPOF ukončí.

Viz „ Aktivování/deaktivování komprese NC bloků (COMPON, COMPCURV, COMPCAD, COMPSURF, COMPOF) (Strana 610) ".

Poznámka

Orientační pohyby budou komprimovány jen tehdy, když je aktivní interpolace pomocí největší kružnice koule (tzn. změna orientace nástroje se uskutečňuje v rovině, která je určena počáteční a koncovou orientací).

Interpolace pomocí největší kružnice koule se provádí, pokud jsou splněny následující podmínky:

- MD21104 $MC_ORI_IPO_WITH_G_CODE = 0,
  ORIWKS je aktivní a
  orientace je naprogramována pomocí vektorů (pomocí příkazů A3, B3, C3, příp. A2, B2, C2).
- MD21104 $MC_ORI_IPO_WITH_G_CODE = 1 a
  ORIVECT, příp. ORIPLANE je aktivní.
  Orientace nástroje může být naprogramována buď jako směrový vektor nebo pomocí poloh kruhových os. Jestliže je aktivní některý z příkazů G-funkce ORICONxx nebo ORICURVE nebo pokud jsou naprogramovány polynomy pro úhel orientace (PO[PHI] a PO[PSI]), interpolace pomocí největší kružnice koule se neprovádí.

Příklad

V následujícím příkladu programování bude komprimován kruh, který je aproximován polynomickou křivkou. Orientace nástroje se přitom pohybuje synchronně po plášti kužele. Přestože postupně za sebou naprogramované změny orientace mohou být nespojité, generuje funkce kompresoru hladký průběh orientace.

```
Programování                                  Komentář
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                  ; maximální odchylka od kontury = 0,05 mm
$SC_COMPRESS_ORI_TOL=5                        ; maximální odchylka orientace = 5 stupňů

TRAORI
COMPCURV                                      ; pohyb se bude uskutečňovat po kruhu, který
                                              se skládá z mnohoúhelníků Orientace se při-
                                              tom pohybuje po plášti kužele okolo osy Z a s
                                              vrcholovým úhlem 45 stupňů.
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
```

4.9.7 Aktivování/deaktivování vyhlazování charakteristiky orientace (ORISON, ORISOF):

Funkci „Vyhlazování průběhu orientace" je možno aktivovat/deaktivovat ve výrobním programu pomocí příkazu G-funkce ze skupiny 61. Příkazy mají modální platnost.

Předpoklady

- Systém s transformací s 5/6 osami.
- Funkce kompresoru COMPCAD je aktivní.

Syntaxe

```
ORISON
...
ORISOF
```

Význam

ORISON:	Aktivování vyhlazování průběhu orientace
ORISOF:	Deaktivování vyhlazování průběhu orientace

Příklad:

```
Programový kód                  Komentář
...
TRAORI()                        ; aktivování transformace orientace
COMPCAD                         ; aktivování funkce kompresoru COMPCAD
ORISON                          ; aktivování vyhlazování orientace
$SC_ORISON_TOL=1.0              ; maximální úhlová odchylka orientace nástroje
                                = 1,0 stupně
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                          ; deaktivování vyhlazování orientace
...
```

Orientace bude naklopena o 90 stupňů v rovině XZ od úhlu -45 stupňů do úhlu +45 stupňů. Prostřednictvím vyhlazování průběhu orientace není už dosahováno maximálních úhlových hodnot orientace -45 stupňů, příp. +45 stupňů.

4.9.8 Kinematická transformace

4.9.8.1 Aktivování transformace čelní plochy (TRANSMIT)

Transformace čelní plochy (TRANSMIT) se ve výrobním programu nebo v synchronní akci aktivuje prostřednictvím příkazu `TRANSMIT`.

Syntaxe

```
TRANSMIT
TRANSMIT(<n>)
```

Význam

`TRANSMIT`:	Aktivování transformace TRANSMIT s prvním datovým blokem TRANSMIT
`TRANSMIT(n)`:	Aktivování transformace TRANSMIT s <n>-tým datovým blokem TRANSMIT

Poznámka

Deaktivování aktivní transformace TRANSMIT v kanálu:

- Deaktivování transformace: `TRAFOOF`
- Aktivování jiné transformace: např. `TRACYL`, `TRAANG`, `TRAORI`

4.9.8.2 Aktivování transformace válcového pláště (TRACYL)

Transformace válcového pláště (TRACYL) se ve výrobním programu nebo v synchronní akci aktivuje prostřednictvím příkazu `TRACYL`.

Syntaxe

```
TRACYL(<d>)
TRACYL(<d>,<n>)
TRACYL(<d>,<n>,<k>)
```

Význam

TRACYL(<d>):	Aktivování transformace TRACYL s prvním datovým blokem TRACYL a pracovním průměrem <d>	
TRACYL (<d>,<n>):	Aktivování transformace TRACYL s <n>-tým datovým blokem TRACYL a pracovním průměrem <d>	
<d>:	Vztažný, příp. pracovní průměr Hodnota musí být větší než 1.	
<n>:	Číslo datového bloku TRACYL (nepovinné)	
	Rozsah hodnot:	1, 2
<k>:	Parametr <k> má smysl pouze pro typ transformace 514	
	k = 0:	bez korekce stěny drážky
	k = 1:	s korekcí stěny drážky
	Jestliže parametr není zadán, uplatňuje se základní nastavení zadané v parametru: $MC_TRACYL_DEFAULT_MODE_<n> kde <n> = číslo datového bloku transformace TRACYL	

Poznámka

Deaktivování aktivní transformace TRACYL v kanálu:

- Deaktivování transformace: TRAFOOF
- Aktivování jiné transformace: např. TRAANG, TRANSMIT, TRAORI

Příklad:

Programový kód	Komentář
...	
N40 TRACYL(40.)	; aktivování transformace TRACYL s prvním datovým blokem TRACYL a pracovním průměrem 40 mm
...	

Další informace

Struktura programu

Výrobní program pro frézování drážky pomocí transformace TRACYL 513 (TRACYL s korekcí stěny drážky) se zpravidla skládá z následujících kroků:

1. Volba nástroje.
2. Aktivování funkce TRACYL.
3. Aktivování vhodného posunutí počátku souřadného systému (FRAME).
4. Najíždění na pozici.
5. Naprogramování příkazu OFFN.
6. Aktivování korekce rádiusu nástroje.

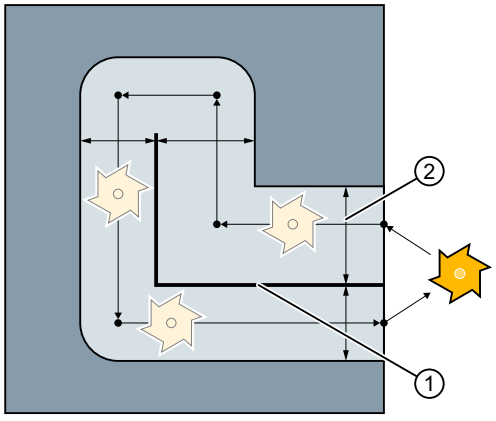
7. Nájížděcí blok (najíždění na korekci rádiusu nástroje a najíždění na stěnu drážky).
8. Kontura osy drážky.
9. Deaktivování korekce rádiusu nástroje.
10. Odjížděcí blok (odjíždění od korekce rádiusu nástroje a odjíždění od stěny drážky).
11. Najíždění na pozici.
12. TRAFOOF.
13. Opětovné aktivování původního posunutí počátku souřadného systému (FRAME).

Offset kontury (OFFN)

Abyste mohli pomocí transformace TRACYL 513 frézovat drážky, ve výrobním programu naprogramujte osu drážky a pomocí adresy OFFN **polovinu šířky drážky**.

Aby se zabránilo poškození stěny drážky, příkaz OFFN se uplatní teprve spolu s aktivovanou korekcí rádiusu nástroje.

Změny parametru OFFN v rámci výrobního programu jsou možné. Tím může být osa drážky posunuta ze středu:



① OFFN
② Naprogramovaná dráha

Poznámka

Parametr OFFN by měl být minimálně tak velký, jako rádius nástroje, aby se vyloučilo poškození protilehlé stěny drážky.

Poznámka

OFFN s funkcí TRACYL funguje jinak než bez funkce TRACYL. Protože se OFFN započítává i bez funkce TRACYL, když je aktivní korekce rádiusu nástroje, po příkazu TRAFOOF by měla být do OFFN znovu dosazena nula.

UPOZORNĚNÍ
Působení parametru OFFN závisí na typu transformace V případě transformace TRACYL 513 (TRACYL s korekcí stěny drážky) se do parametru OFFN naprogramuje polovina šířky drážky. V případě transformace TRACYL 512 (TRACYL bez korekce stěny drážky) se oproti tomu hodnota zadaná do parametru OFFN uplatňuje jako přídavek rozměru ke korekci rádiusu nástroje.

Korekce rádiusu nástroje (WRK)

V případě transformace TRACYL 513 se korekce rádiusu nástroje nezapočítává vzhledem ke stěně drážky, nýbrž vůči jejímu naprogramovanému středu. Aby se nástroj pohyboval vlevo od stěny drážky, je potřeba naprogramovat příkaz G42, nikoli G41 nebo hodnotu parametru OFFN zadat se záporným znaménkem.

Průměr nástroje

Pomocí transformace TRACYL a nástroje, jehož průměr je menší než šířka drážky, není vyrobena stejná geometrie stěny drážky jako s nástrojem, jehož průměr se rovná šířce drážky. Aby se zvýšila přesnost, doporučuje se zvolit nástroj, jehož průměr je jen o málo menší, než je šířka drážky.

Využití os

Poznámka

Následující osy nemohou být používány jako polohovací osy, příp. osy vykonávající kyvný pohyb:

- Geometrické osy v obvodovém směru plášťové plochy válce (osa Y)
- Doplňkové lineární osy při korekci stěny drážky (osa Z)

4.9.8.3 Aktivování transformace šikmého úhlu s naprogramovatelným úhlem (TRAANG)

Transformace šikmého úhlu s naprogramovatelným úhlem se ve výrobním programu nebo v synchronní akci aktivuje prostřednictvím příkazu `TRAANG`.

Omezení u kinematických řetězců

Při nastavování parametrů pomocí strojních parametrů je potřeba prostřednictvím příslušných strojních parametrů definovat úhel α, který šikmá osa svírá s odpovídající souřadnou osou v pravoúhlém souřadném systému. Úhel α může být při volání prostřednictvím příkazu TRAANG(<α>) změněn. Změna tohoto úhlu znamená také změnu kinematického modelu, v důsledku čehož mohou být tímto nastavením ovlivněny i jiné aplikace. Zadání tohoto úhlu je proto přípustné jen tehdy, jestliže je tento úhel identický s úhlem, který vyplývá z kinematického řetězce.

Poznámka

Definice úhlu

Úhel je definován jen tehdy, jestliže poloha šikmé osy vznikne otočením okolo jedné souřadné osy. Existuje tedy jen jedna geometrická osa, která není rovnoběžně s jednou z os souřadného systému a která leží v hlavní rovině.

Syntaxe

```
TRAANG
TRAANG()
TRAANG(,<n>)
TRAANG(<α>)
TRAANG(<α>,<n>)
```

Význam

TRAANG: TRAANG():	Aktivování transformace TRAANG s prvním datovým blokem TRAANG a s naposled platným úhlem <α>	
TRAANG(,<n>):	Aktivování transformace TRAANG s <n>-tým datovým blokem TRAANG a s naposled platným úhlem <α>	
TRAANG(<α>):	Aktivování transformace TRAANG s prvním datovým blokem TRAANG a s úhlem <α>	
TRAANG(<α>,<n>):	Aktivování transformace TRAANG s <n>-tým datovým blokem TRAANG a s <α>	
<α>:	Úhel šikmo nastavené osy (nepovinné)	
	Rozsah hodnot:	-90° < α < + 90°
	Pokud není zadán žádný úhel, použije se základní nastavení definované ve strojním parametru: MD2xxxx $MC_TRAANG_ANGLE_<n>	
<n>:	Číslo datového bloku TRAANG (nepovinné)	
	Rozsah hodnot:	1, 2

Poznámka

Transformace šikmého úhlu TRAANG, která je aktivní v daném kanálu, se deaktivuje pomocí některé z těchto možností_

- Deaktivování transformace: TRAFOOF
- Aktivování jiné transformace: např. TRACYL, TRANSMIT, TRAORI

Příklad

Programový kód	Komentář
N20 TRAANG(45)	; aktivování transformace TRAANG s prvním datovým blokem TRAANG a s úhlem 45°

4.9.8.4 Šikmé zapichování na bruskách (G5, G7)

Příkazy G-funkcí G7 a G5 slouží pro zjednodušení programování šikmých zápichů na bruskách s transformací "Šikmá osa (TRAANG)", takže se při výrobě zápichu pohybuje pouze příslušná šikmá osa.

Naprogramovat je přitom potřeba pouze požadovanou koncovou pozici zapichovacího pohybu ve směrech X a Z. Je-li aktivován příkaz G7, je odpovídající počáteční pozice, na kterou se najíždí, přitom vypočítána NC systémem, přičemž se vychází z aktuální pozice osy X, naprogramované koncové pozice a úhlu α.

Počáteční poloha vyplývá z průsečíku dvou přímek:

- Přímka rovnoběžná s osou Z ve vzdálenosti aktuální pozice osy X
- Přímka rovnoběžná s šikmou osou a procházející naprogramovanou koncovou pozicí

Pomocí následujícího příkazu G5 se šikmou osou najíždí na naprogramovanou koncovou pozici.

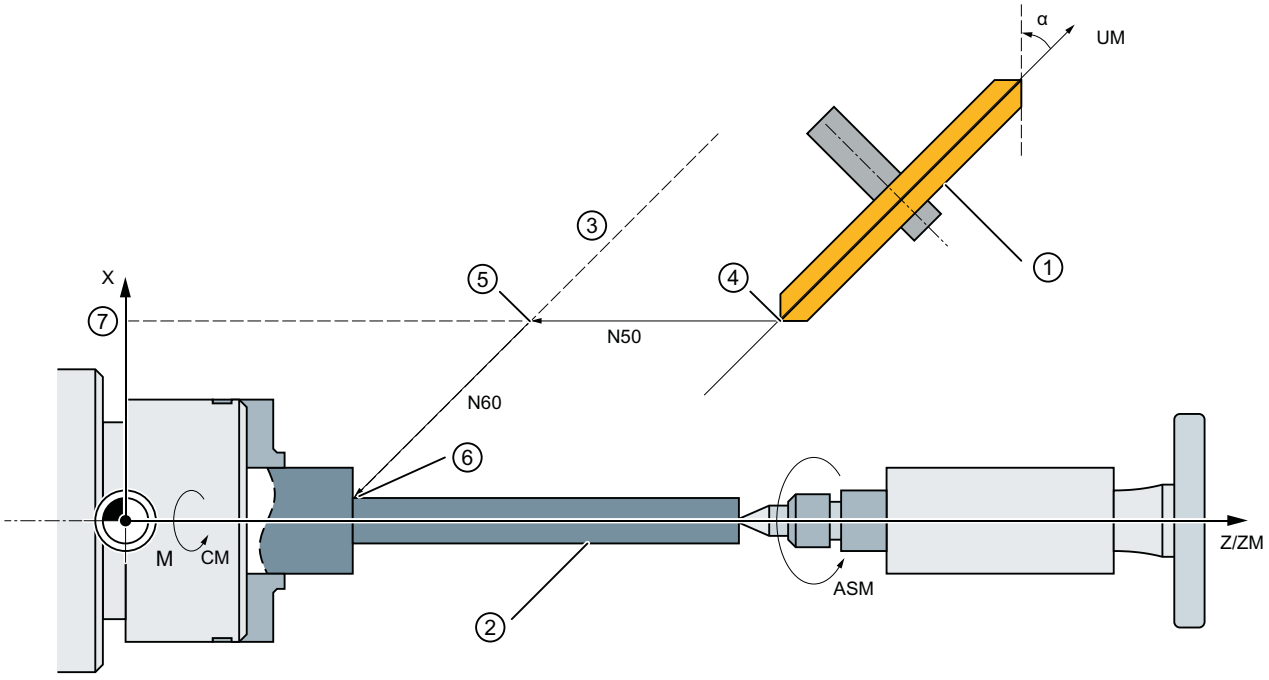
Syntaxe

```
G7 <Endpos_X> <Endpos_Z>
G5 <Endpos_X>
```

Význam

G7:	Výpočet počáteční pozice pro šikmé zapichování a najíždění na tuto pozici
G5:	Najíždění šikmou osou na naprogramovanou koncovou pozici
<Endpos_X>:	Koncová poloha osy X
<Endpos_Z>:	Koncová poloha osy Z

Příklad:



① Brusný kotouč
② Obrobek
③ Přímka rovnoběžná s šikmou osou a procházející naprogramovanou koncovou pozicí
④ Výchozí pozice
⑤ Výroba zápichu: Počáteční pozice
⑥ Výroba zápichu: Koncová pozice
⑦ Přímka rovnoběžná s osou Z ve vzdálenosti aktuální pozice osy X
X Geometrická osa
Z Geometrická osa
ZM Osa stroje
UM Osa stroje

Obrázek 4-6 Programování šikmé osy

Programový kód	**Komentář**
`N... G18`	`; aktivování roviny XZ`
`N40 TRAANG(45.0)`	`; aktivování transformace TRAANG, úhel = 45°`
`N50 G7 X40 Z70 F4000`	`; výpočet počáteční pozice a najíždění na tuto pozici`
`N60 G5 X40 F100`	`; najíždění šikmou osou na koncovou pozici`
`N70 ...`	

4.9.9 Aktivování zřetězené transformace (TRACON)

Zřetězené transformace definované v konfiguraci se ve výrobním programu nebo v synchronní akci aktivují prostřednictvím příkazu TRAFOON (Strana 733) nebo TRACON.

Aby bylo možné příkazem TRACON aktivovat transformaci, jejíž parametry jsou dány kinematickým řetězcem, musí být parametry této transformace nastaveny v systémové proměnné $NT_TRACON_CHAIN.

Syntaxe

```
TRACON(<Trafo_Nr>,<Par_1>,...,<Par_n>,<Par_n+1>)
...
TRAFOOF
```

Význam

TRACON:	Aktivování zřetězené transformace Jiná dříve aktivovaná transformace se příkazem TRACON() implicitně zruší.		
<Trafo_Nr>:	Číslo zřetězené transformace		
	Typ:	INT	
	Rozsah hodnot:	0 ... 2	
	Hodnota:	0, 1	první/jediná zřetězená transformace
		2	druhá zřetězená transformace
		žádný údaj	znamená totéž, co 0, příp. 1
	Poznámka: Jiné hodnoty než 0, 1, 2 budou mít za následek alarm.		
<Par_1>, ..., <Par_n>, <Par_n+1>:	Parametry pro vzájemně zřetězené transformace		
	<Par_1>, ..., <Par_n>	Parametry první transformace v řetězci Skutečný počet závisí na typu transformace: • TRAORI: 2 parametry • TRACYL: 1 až 2 parametry	
	<Par_n+1>	Parametry druhé transformace v řetězci (TRAANG) ≙ Úhel šikmé osy	
	Jestliže tyto parametry nejsou nastaveny, uplatní se předdefinovaná nastavení nebo naposled použité parametry. Jestliže mají být pro předtím uvedené parametry uplatněna nastavení předvoleb, musí být nastavením čárek zajištěno, aby byly zadané parametry vyhodnocovány v posloupnosti, ve které jsou očekávány. Zejména v případě zadání nejméně jednoho parametru musí být před tímto parametrem uvedena čárka, a to i když zadání parametru <Trafo_Nr> není zapotřebí, například TRACON(, 3.7).		
TRAFOOF:	Deaktivování naposled aktivované (zřetězené) transformace		

Příklad

Programový kód	Komentář
...	
N230 TRACON(1,45.)	; aktivování první zřetězené transformace
	; dříve aktivní transformace se automaticky deaktivuje
	; Úhel pro šikmou osu je 45°.
...	
N330 TRACON(2,40.)	; aktivování druhé zřetězené transformace
	; Úhel pro šikmou osu je 40°.
...	
N380 TRAFOOF	; deaktivování druhé zřetězené transformace
...	

4.9.10 Posuv PTP v kartézských souřadnicích

4.9.10.1 Aktivování/deaktivování posuvu PTP v kartézských souřadnicích (PTP, PTPG0, PTPWOC, CP)

Kartézské pohyby od bodu k bodu (PTP) je možno v NC programu aktivovat/deaktivovat pomocí příkazů z G-skupiny 49.

Příkazy mají modální platnost. Předdefinované nastavení odpovídá kartézskému pohybu po dráze (CP).

Na rozdíl od CP se při pohybech typu PTP uskutečňuje ještě navíc transformace kartézského cílového bodu a strojní osy se pohybují synchronně.

Aby bylo možno kartézský cílový bod jednoznačně přepočítat na hodnoty strojích os, jsou kromě údajů poloh a úhlů zapotřebí také informace, které popisují polohy os. Tyto údaje se zadávají prostřednictvím nastavitelných adres STAT (Strana 713) a TU (Strana 717).

Předpoklady

Transformace TRAORI_DYN, TRAORI_STAT, TRANSMIT_K, RCTRA nebo ROBX je aktivní (obvykle TRAORI a TRANSMIT).

RCTRA nebo ROBX nejsou k dispozici pro kinematické řetězce.

Syntaxe

```
PTP / PTPG0 / PTPWOC
...
CP
```

Význam

PTP:	Aktivování pohybu od bodu k bodu (PTP) Prostřednictvím synchronního pohybu os se najede na kartézskou pozici naprogramovanou v blocích s příkazy G0 a G1.
PTPG0:	Aktivování pohybu od bodu k bodu PTPG0 Prostřednictvím synchronního pohybu os se najede na kartézskou pozici, která byla naprogramována jen v blocích s příkazy G0. V blocích s G1 se přepíná na pohyb po dráze CP.
PTPWOC:	Aktivování pohybu od bodu k bodu PTPWOC (možné pouze tehdy, když je aktivní transformace orientace) Stejné jako PTP, avšak bez kompenzačních pohybů, které jsou způsobovány pohyby kruhových os a os nastavujících orientaci.
CP:	Zrušení pohybu typu od bodu k bodu a aktivování pohybu po dráze CP Prostřednictvím příkazu CP se uskutečňuje kartézský pohyb po dráze.

Poznámka

PTPWOC

V kombinaci s transformací RCTRA nebo ROBX nemá použití příkazu PTPWOC žádný smysl! RCTRA nebo ROBX nejsou k dispozici pro kinematické řetězce.

Příklady

Viz:

- Příklad 1: Pohyb PTP robota se 6 osami pomocí transformace ROBX (Strana 720)
- Příklad 2: Najíždění PTP při obecné 5-osé transformaci (Strana 721)
- Příklad 3: PTPG0 a TRANSMIT (Strana 722)

4.9.10.2 Zadávání polohy kloubu (STAT)

Údaje polohy pomocí kartézských souřadnic a údaje orientace nástroje nejsou pro jednoznačné stanovení polohy stroje postačující, protože při stejné orientaci nástroje je možný větší počet poloh kloubů. V závislosti na tom, o jakou kinematiku se jedná, existuje až 8 rozdílných poloh kloubů. Tyto odlišné polohy kloubů závisejí na konkrétní transformaci.

Aby se zabránilo nejednoznačnostem, zadává se poloha kloubu pomocí adresy STAT.

Poznámka

Řídící systém bere ohled na naprogramovanou polohu STAT jen při pohybech typu PTP. Při pohybech v režimu CP je ignorována, protože při pohybech s aktivní transformací není za normálních okolností změna polohy možná. Při pohybech s aktivní funkcí CP se pro cílový bod přebírá poloha platná pro počáteční bod.

Syntaxe

```
STAT=<hodnota>
```

Význam

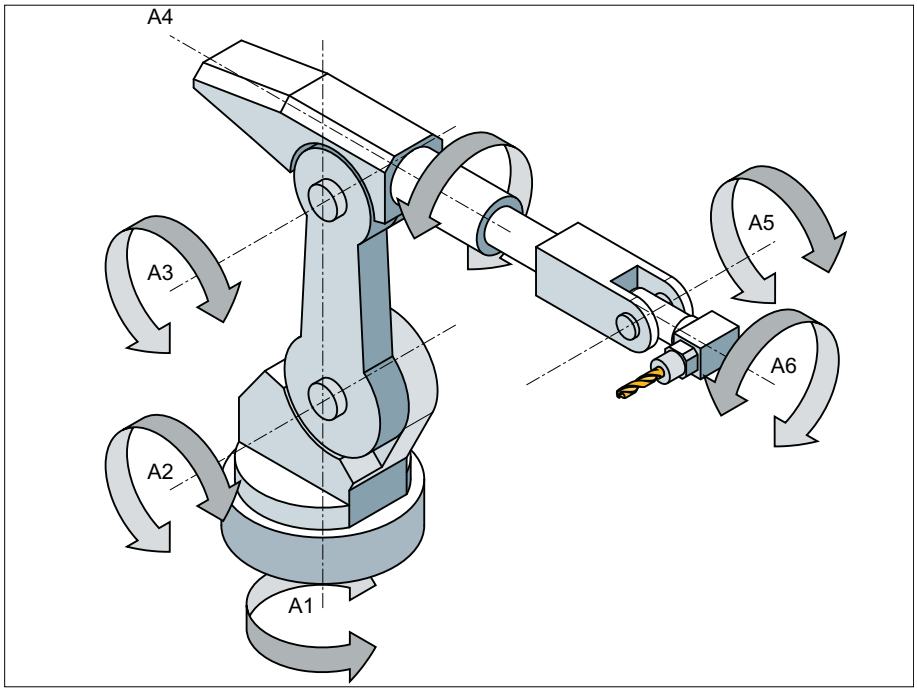
`STAT`:	Nastavitelná adresa pro zadání poloh kloubu
`<hodnota>`:	Binární nebo decimální hodnota Obsahuje pro každou z možných poloh jeden bit. Význam bitů je definován příslušnou transformací.

Použití příkazu STAT má být znázorněno na příkladu robota s kloubovým ramenem s frézovacím vřetenem a se 6 osami. Kinematická transformace se má uskutečňovat pomocí robotické transformace ROBX (předpoklad: Cyklus překladače „Rozšířená robotická transformace RMCC/ROBX je aktivní).

Poznámka

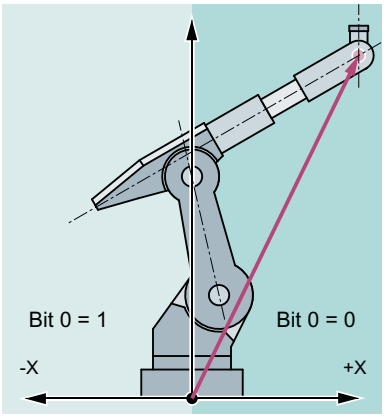
Příklad s transformací ROBX jen s konvenčním nastavením strojních parametrů

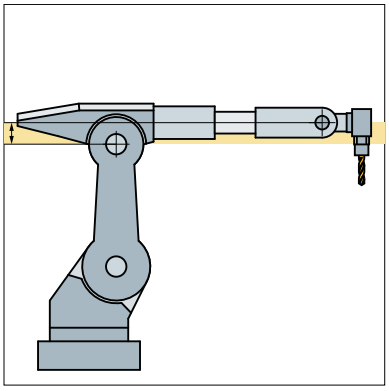
Tento příklad je možný pouze s nastavením strojních parametrů, nikoli s kinematickými řetězci.



Osy A1, A2 a A3 jsou hlavními osami robota s kloubovým ramenem. Prostřednictvím hlavních os jsou v pracovním prostoru nastavovány polohy os A4, A5 a A6, které jsou označeny jako osy hlavičky nebo manuální osy. Prostřednictvím doplňkových pohybových možností manuálních os může být frézovací vřeteno nastaveno v prostoru do orientace, která je pro obráběcí úlohu zapotřebí. Při stejné orientaci nástroje jsou přitom možná různá nastavení polohy kloubů.

Pro zvolení polohy kloubů, která se použije pro obrábění, se používá programování bitů 0 ... 2 nastavitelné adresy STAT:

<table>
<tr><td>Bit 0</td><td colspan="3">Poloha průsečíku manuálně nastavovaných os (A4 A5 A6)</td></tr>
<tr><td></td><td>= 0</td><td>Základní oblast (rameno vpravo)
Robot se nachází v základní oblasti, jestliže vzhledem k souřadnému systému A1 je hodnota X průsečíku manuálně ovládaných os kladná.</td><td rowspan="2"> 

Příklad: Průsečík manuálních os se nachází v základní oblasti</td></tr>
<tr><td></td><td>= 1</td><td>Obrácená oblast (rameno vlevo)
Robot se nachází v obrácené oblasti, jestliže vzhledem k souřadnému systému A1 je hodnota X průsečíku manuálně ovládaných os záporná.</td></tr>
<tr><td>Bit 1</td><td colspan="3">Poloha osy 3
Úhel, při kterém se hodnota bitu 1 změní, závisí na typu robota. Pro robota, jehož osa 3 a 4 se protínají, platí:</td></tr>
<tr><td></td><td>= 0</td><td colspan="2">A3 < 0° (loket dolů)</td></tr>
<tr><td></td><td>= 1</td><td colspan="2">A3 ≥ 0° (loket nahoru)</td></tr>
<tr><td></td><td colspan="2">Poznámka:
U robotů, které mají mezi osou 3 a osou 4 offset, je úhel, při kterém se hodnota bitu 1 změní, závislý na velikosti tohoto offsetu.</td><td> 
Offset mezi A3 a A4</td></tr>
<tr><td>Bit 2</td><td colspan="3">Poloha osy 5</td></tr>
<tr><td></td><td>= 0</td><td colspan="2">A5 ≥ 0° (žádné obrácení konce)</td></tr>
<tr><td></td><td>= 1</td><td colspan="2">A5 < 0° (obrácení konce)</td></tr>
</table>

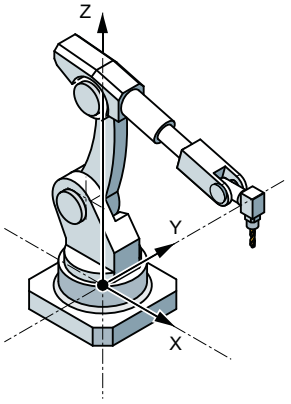
Příklad programování:

```
Programový kód                                    Komentář
...
N14 T="T8MILLD20" D1                              ; $TC_DP3[1,1]=132.95
N16 ORIMKS
```

Programový kód	**Komentář**
`N17 G1 PTP X1665.67 Y0 Z1377.405 A=0 B=0 C=0 **STAT=...** F2000`	`; hodnota STAT určuje polohy kloubů (viz dále)`
`...`	

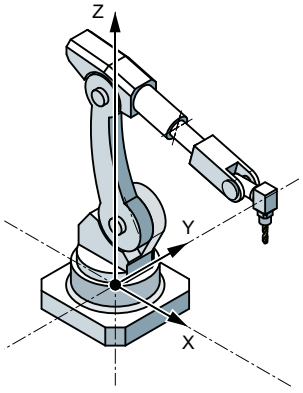
`STAT=1 ('B001')`

→ rameno vlevo
→ loket dolů
→ žádné obrácení konce



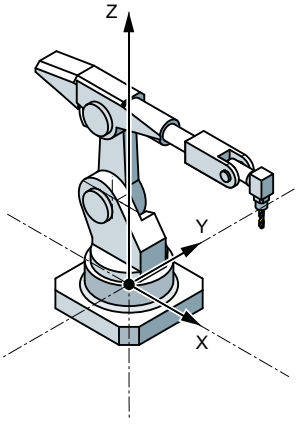
`STAT=2 ('B010')`

→ rameno vpravo
→ loket nahoru
→ žádné obrácení konce

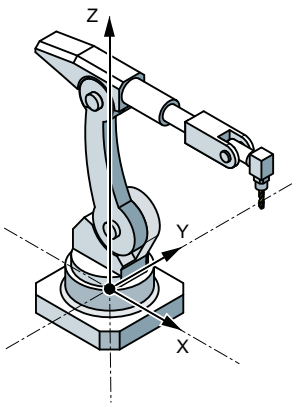


`STAT=5 ('B101')`

→ rameno vlevo
→ loket dolů
→ obrácení konce



`STAT=6 ('B110')` → rameno vpravo
→ loket nahoru
→ obrácení konce



TRANSMIT_K (konvenční transformace TRANSMIT)

U transformace TRANSMIT_K se adresa STAT používá pro odstranění nejednoznačnosti, pokud jde o polohu pólu.

Jestliže se kruhová osa musí otočit o 180°, příp. pokud by v režimu CP procházela kontura pólem, platí:

Bit 0	Má smysl jen tehdy, pokud $NT_POLE_SIDE_FIX[n] = 1, příp. 2 v případě nastavení strojních parametrů je $MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 1, příp. 2:	
	= 0	Kruhová osa se pohybuje o +180°, příp. se otáčí ve směru hodinových ručiček.
	= 1	Kruhová osa se pohybuje o -180°, příp. se otáčí proti směru hodinových ručiček.
Bit 1	Má smysl jen tehdy, pokud $NT_POLE_SIDE_FIX[n] = 0 v případě nastavení strojích parametrů je $MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 0:	
	= 0	Bude se projíždět pólem. Kruhová osa se neotáčí.
	= 1	Použije se otáčení okolo pólu. Přitom se vyhodnocuje bit 0 adresy STAT.

4.9.10.3 Zadání znaménka úhlu osy (TU)

Aby bylo možno u točivých os najíždět také na úhly os, které jsou větší než +180° nebo je menší než -180°, aniž by bylo potřeba definovat nějakou zvláštní strategii (např. pomocí vnitřního bodu), musíte prostřednictvím nastavitelné adresy TU zadat znaménko úhlu osy.

Poznámka

Řídící systém zohledňuje naprogramované hodnoty TU pouze při pohybech v režimu PTP. Při pohybech v režimu CP je tento parametr ignorován.

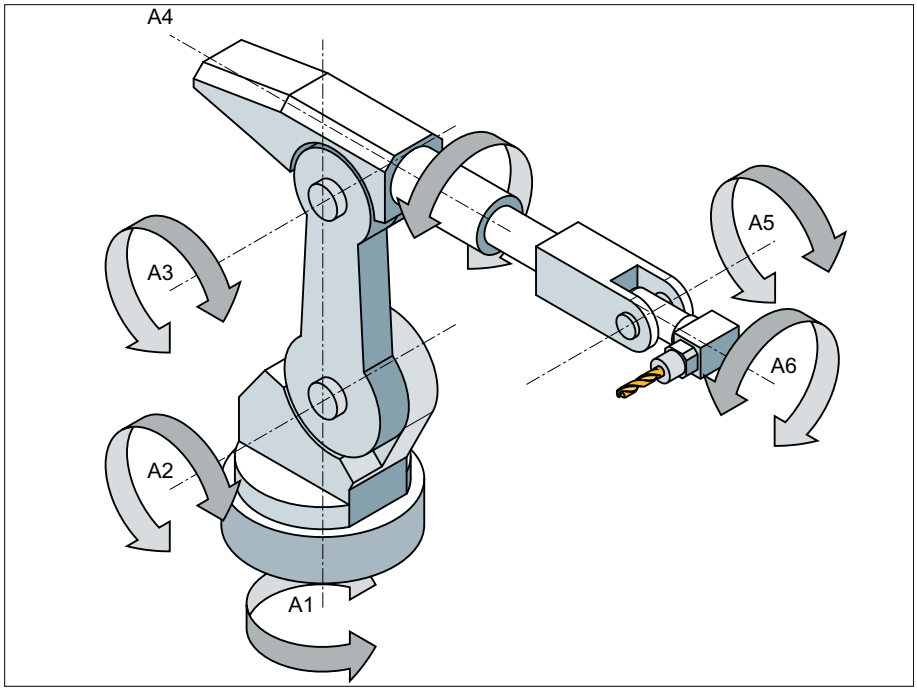
Syntaxe

`TU=<hodnota>`

Význam

TU:	Nastavitelná adresa pro zadání znaménka úhlu osy			
<hodnota>:	Binární nebo decimální hodnota Pro každou osu, která vstupuje do transformace, existuje jeden bit, který udává znaménko úhlu osy (θ), a tím pádem také směr pohybu.			
	Bit	= 0	Znaménko úhlu osy: +	Úhlový rozsah osy: 0° ≤ θ < 360°
		= 1	Znaménko úhlu osy: -	Úhlový rozsah osy: - 360° < θ < 0°

Příklad: Robot s kloubovým ramenem se 6 osami



Bit	Význam	Hodnota	Znaménko úhlu osy	Úhel osy
Bit 0 [1)]	Znaménko pro úhel osy A1	= 0	+	≥ 0°
		= 1	-	< 0°
Bit 1 [1)]	Znaménko pro úhel osy A2	= 0	+	≥ 0°
		= 1	-	< 0°
Bit 2 [1)]	Znaménko pro úhel osy A3	= 0	+	≥ 0°
		= 1	-	< 0°
Bit 3 [1)]	Znaménko pro úhel osy A4	= 0	+	≥ 0°
		= 1	-	< 0°
Bit 4 [1)]	Znaménko pro úhel osy A5	= 0	+	≥ 0°
		= 1	-	< 0°

Bit	Význam	Hodnota	Znaménko úhlu osy	Úhel osy
Bit 5 [1)]	Znaménko pro úhel osy A6	= 0	+	≥ 0°
		= 1	-	< 0°

[1)] Skutečná čísla bitů TU vyplývají z čísel kanálových os robota! V příkladu odpovídají osy robota (A1 až A6) prvním šesti osám v kanálu, v důsledku čehož se používají bity 0 ... 5 proměnné TU. V případě jiného přiřazení kanálových os osám robota by se čísla bitů proměnné TU os robota odpovídajícím způsobem změnila (např.: osy robota jsou 3. až 8. kanálová osa, tzn. pro osy robota se budou používat bity 2 ... 7 proměnné TU.)

TU=19 (odpovídá TU='B010011) by tedy znamenalo:

Bit	Hodnota		Úhel osy
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	⇒	$\theta_{A6} \geq 0°$

Poznámka

U os s rozsahem pohybu > ±360° se pohyb vždy uskutečňuje po nejkratší dráze, protože pomocí informace v parametru TU není možno polohu osy jednoznačně určit.

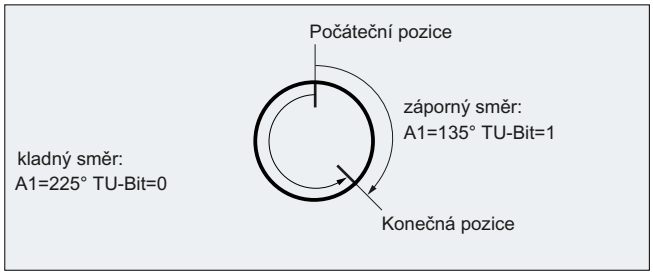
Jestliže u nějaké pozice není naprogramována žádné hodnota TU, bude se systém v závislosti na parametru MD30455 $MA_MISC_FUNCTION_MASK nastavovat po kratší nebo po delší dráze.

TRANSMIT_K (konvenční transformace TRANSMIT)

Při pohybu v režimu PTP a při aktivní transformaci TRANSMIT nemá adresa TU žádný význam!

Příklad

Na pozici kruhové osy uvedenou na následujícím obrázku je možné najet v záporném nebo v kladném směru. Pomocí adresy A1 je naprogramována úhlová poloha. Teprve prostřednictvím zadání adresy TU je směr pohybu jednoznačný.



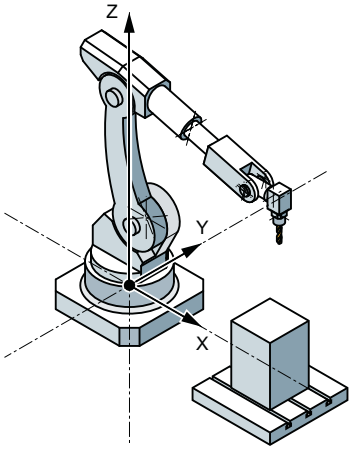
4.9.10.4 Příklad 1: Pohyb PTP robota se 6 osami pomocí transformace ROBX

V následujícím příkladu použití je vysvětlován kartézský pohyb PTP a s ním související příkazy NC jazyka.

Poznámka

Pouze s konvenčním nastavením strojních parametrů

S kinematickými řetězci tento příklad nefunguje.



Obrázek 4-7 Robot s kloubovým ramenem se 6 osami a s frézovacím vřetenem

```
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
```

4.9.10.5 Příklad 2: Najíždění PTP při obecné 5-osé transformaci

Předpoklad: Základem je pravoúhlá kinematika typu CA.

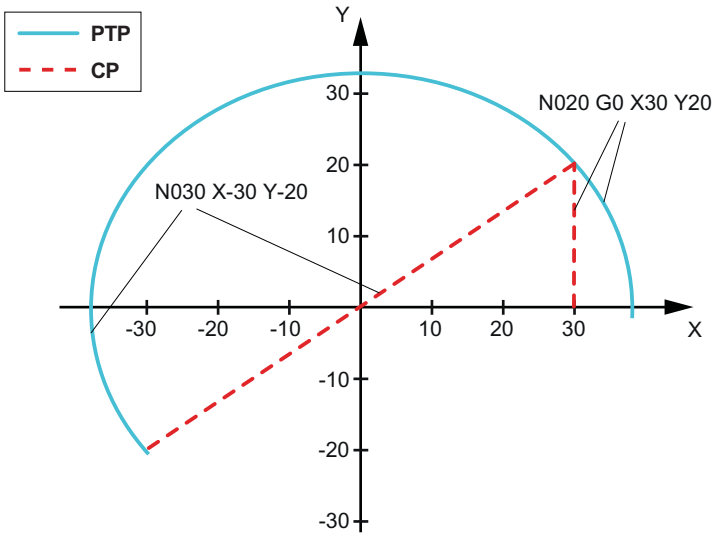
Programový kód	**Komentář**
TRAORI	; aktivování transformace kinematiky typu CA
PTP	; aktivování posuvu PTP
N10 A3=0 B3=0 C3=1	; polohy kruhových os C=0 A=0
N20 A3=1 B3=0 C3=1	; polohy kruhových os C=90 A=45
N30 A3=1 B3=0 C3=0	; polohy kruhových os C=90 A=90
N40 A3=1 B3=0 C3=1 **STAT=1**	; polohy kruhových os C=270 A=–45

Volba jednoznačně určené polohy kruhových os pro najíždění:

V bloku N40 v důsledku programového příkazu **STAT=1** přitom najíždějí kruhové osy ze svého počátečního bodu (C=90, A=90) do koncového bodu (C=270, A=–45) po delší dráze. Pokud by bylo naprogramováno **STAT=0**, kruhové osy by oproti tomu do koncového bodu (C=90, A=45) najížděly po kratší dráze.

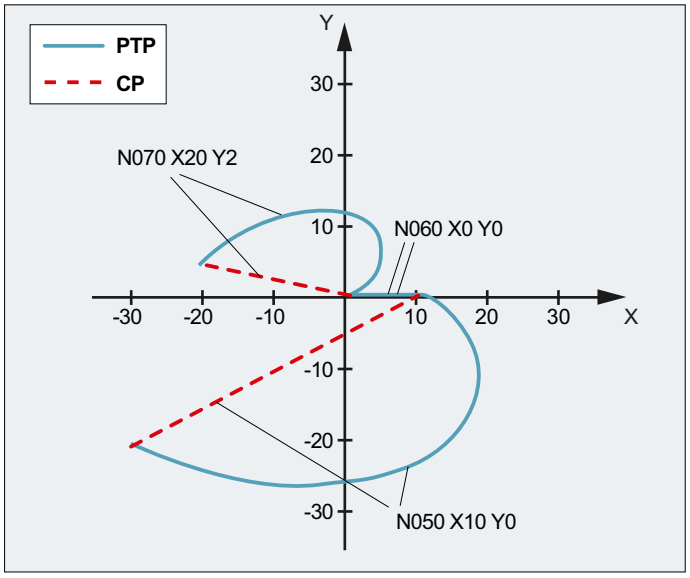
4.9.10.6 Příklad 3: PTPG0 a TRANSMIT

Objíždění pólu s funkcemi PTPG0 a TRANSMIT



Programový kód	**Komentář**
N001 G0 X30 Z0 F10000 T1 D1 G90	; výchozí pozice v absolutních rozměrech
N002 SPOS=0	
N003 **TRANSMIT**	; transformace TRANSMIT
N010 **PTPG0**	; ke každému bloku s G0 automaticky PTP a potom znovu CP
N020 G0 X30 Y20	
N030 X-30 Y-20	
N120 G1 X30 Y20	
N110 X30 Y0	
M30	

Odjíždění od pólu pomocí funkcí PTPG0 a TRANSMIT



Programování	**Komentář**
N001 G0 X90 Z0 F10000 T1 D1 G90	; výchozí nastavení
N002 SPOS=0	
N003 **TRANSMIT**	; transformace TRANSMIT
N010 **PTPG0**	; ke každému bloku s G0 automaticky PTP a potom znovu 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	

4.9.11 Okrajové podmínky při aktivování transformace

Funkce

Aktivování transformací je možné pomocí výrobního programu, příp. MDA. Přitom je potřeba mít na paměti následující:

- Pomocný pohybový blok se nevkládá (fasety/rádiusy).
- Posloupnost splinových bloků musí být uzavřena; pokud není, objeví se hlášení.

- Jemná korekce nástroje musí být deaktivována (FTOCOF); pokud není, objeví se hlášení.
- Korekce rádiusu nástroje musí být deaktivována (G40); pokud není, objeví se hlášení.
- Aktivovaná korekce délky nástroje je řídícím systémem přenášena do transformace.
- Aktuální frame, který byl před transformací aktivní, bude řídícím systémem deaktivován.
- Aktivní ohraničení pracovního pole jsou pro osy podílející se na transformaci řídícím systémem deaktivována (odpovídá příkazu WALOMOF).
- Monitorování chráněných oblastí je deaktivováno.
- Režim řízení pohybu po dráze a přechodová zaoblení se přeruší.
- Všechny osy uvedené ve strojním parametru musí být vzhledem k danému bloku synchronizovány.
- Vyměněné osy musí být vyměněny zpátky, pokud tomu tak není, objeví se hlášení.
- V případě závislých os se objeví hlášení.

Výměna nástroje

Výměna nástroje je přípustná, jen když je korekce rádiusu nástroje deaktivována.

Změna korekce délky nástroje a aktivování/deaktivování korekce rádiusu nástroje nesmí být naprogramovány ve stejném bloku.

Změna framu

Všechny příkazy, které se vztahují jen na základní souřadný systém, jsou přípustné (frame, korekce rádiusu nástroje). Změna framu, když je aktivní funkce G91 (inkrementální rozměry) ale není - narozdíl od situace, kdy transformace není aktivní - nijak zvlášť ošetřena. Inkrement, který se má urazit, se vyhodnocuje v souřadném systému obrobku nového framu - nezávisle na tom, který frame se uplatňoval v předcházejícím bloku.

Výjimky

Osy, které se podílejí na transformaci, se nemohou používat:

- jako osy s příkazem PRESET (alarm)
- pro najíždění na pevný bod (alarm)
- pro najíždění na referenční bod (alarm)

4.9.12 Deaktivování transformace (TRAFOOF)

Pomocí předem definované procedury TRAFOOF jsou všechny aktivní transformace a framy deaktivovány.

Poznámka

Pro deaktivování transformací platí stejné okrajové podmínky (Strana 723) jako pro jejich aktivování.

Framy, které jsou potom zapotřební, musí být novým naprogramováním přepnuty do aktivního stavu.

Syntaxe

```
...
TRAFOOF
```

Význam

`TRAFOOF`:	Deaktivování všech aktivních transformací/framů

4.10 Kinematické řetězce

4.10.1 Mazání komponent (DELOBJ)

Funkce `DELOBJ()` „vymaže" komponenty prostřednictvím nastavení přiřazených systémových proměnných na jejich původní předdefinované hodnoty:

- Prvky kinematických řetězců
- Chráněné oblasti, prvky chráněných oblastí a dvojice oblastí chráněných kvůli kolizi
- Data transformací

Syntaxe

```
[<RetVal>=] DELOBJ(<CompType>[,,,<NoAlarm>)])
[<RetVal>=] DELOBJ(<CompType>,<Index1>[,,<NoAlarm>])
[<RetVal>=] DELOBJ(<CompType>[,<Index1>][,<Index2>][,<NoAlarm>])
```

Význam

DELOBJ:	Mazání prvků z kinematických řetězců, chráněných oblastí, prvků chráněných oblastí, dvojic oblastí chráněných kvůli kolizi a dat transformací	
<CompType>:	Typ komponentu, který má být vymazán	
	Datový typ:	STRING
	Hodnota: "KIN_CHAIN_ELEM" **Význam**: Systémové proměnné pro všechny kinematické prvky: $NK_...	
	Hodnota: "KIN_CHAIN_SWITCH" **Význam**: Systémová proměnná $NK_SWITCH[<i>]	
	Hodnota: "KIN_CHAIN_ALL" **Význam:** Všechny kinematické prvky a přepínače. Stejný význam má postupné vyvolání funkce DELOBJ s hodnotami "KIN_CHAIN_ELEM" a "KIN_CHAIN_SWITCH"	
	Hodnota: "PROT_AREA" **Význam**: Systémové proměnné chráněných oblastí: • $NP_PROT_NAME • $NP_CHAIN_NAME • $NP_CHAIN_ELEM • $NP_1ST_PROT	
	Hodnota: "PROT_AREA_ELEM" **Význam**: Systémové proměnné prvků chráněných oblastí stroje a/nebo automatických chráněných oblastí nástroje: • $NP_NAME • $NP_NEXT • $NP_NEXTP • $NP_COLOR • $NP_D_LEVEL • $NP_USAGE • $NP_TYPE • $NP_FILENAME • $NP_PARA • $NP_OFF • $NP_DIR • $NP_ANG	
	Hodnota: "PROT_AREA_COLL_PAIRS" **Význam**: Systémové proměnné dvojic oblastí chráněných kvůli kolizi: • $NP_COLL_PAIR • $NP_SAFETY_DIST	
	Hodnota: "PROT_AREA_ALL" **Význam**: Všechny chráněné oblasti, prvky chráněných oblastí a dvojice oblastí chráněných kvůli kolizi (systémové proměnné $NP_...) Stejný význam má postupné vyvolání funkce DELOBJ s hodnotami "PROT_AREA", "PROT_AREA_ELEM" a "PROT_AREA_COLL_PAIRS"	
	Hodnota: "TRAFO_DATA" **Význam**: Systémové proměnné všech transformací $NT_...	

<Index1>:	Index prvního komponentu, který má být vymazán (**nepovinné**)	
	Datový typ:	INT
	Předdefinovaná hodnota:	-1
	Rozsah hodnot:	-1 ≤ x ≤ (maximální počet komponent nastavený v konfiguraci -1)
	Hodnota	**Význam**
	0, 1, 2, ...	Index komponentu, který má být vymazán.
	-1	Všechny komponenty uvedeného typu se vymažou. <Index2> se nevyhodnocuje.
<Index2>:	Index posledního komponentu, který má být vymazán (**nepovinné**) Pokud <Index2> není naprogramováno, jsou vymazány pouze systémové proměnné komponentů, na které se vztahuje parametr <Index1>.	
	Datový typ:	INT
	Předdefinovaná hodnota:	Jsou vymazány pouze systémové proměnné komponentů, na které se vztahuje parametr <Index1>.
	Rozsah hodnot:	<Index1> < x ≤ (max. počet komponent nastavený v konfiguraci -1)
<NoAlarm>:	Potlačení alarmů (**nepovinné**)	
	Datový typ:	BOOL
	Předdefinovaná hodnota:	FALSE
	Hodnota	**Význam**
	FALSE	V případě chyby (<RetVal> < 0) se zpracovávání programu pozastaví a aktivuje se alarm ..
	TRUE	V případě chyby se zpracovávání programu nepozastaví a neaktivuje se žádný alarm. Případ použití: Specifická uživatelská reakce v závislosti na výstupní hodnotě.
<RetVal>:	Výsledná hodnota funkce	
	Datový typ:	INT
	Rozsah hodnot:	0, -1, -2, ... -7
	Hodnota	**Význam**
	0	Nevyskytla se žádná chyba.
	-1	Volání funkce bez parametru. Musí být zadán alespoň parametr <CompType>.
	-2	<CompType> označuje neznámý komponent
	-3	<Index1> je menší než -1
	-4	<Index1> je větší než počet komponent nastavený v konfiguraci
	-5	<Index1> má při mazání **skupiny komponent** hodnotu nerovnající se -1
	-6	<Index2> je menší než <Index1>
	-7	<Index2> je větší než počet komponent nastavený v konfiguraci
	-8	Oprávnění k zápisu není dostatečné.

4.10.2 Zjišťování indexu podle názvu (NAMETOINT)

Do polí systémových proměnných typu STRING se ukládají specifické uživatelské názvy. Na základě identifikátorů systémových proměnných a názvů zjišťuje funkce `NAMETOINT()` hodnotu indexu, která patří k danému názvu a pod kterou je v poli systémových proměnných uložen.

Syntaxe

```
<RetVal> = NAMETOINT(<SysVar>,<Name>[,<NoAlarm>])
```

Význam

`NAMETOINT`:	Zjištění indexu systémové proměnné	
`<SysVar>`:	Název systémové proměnné typu STRING	
	Datový typ:	STRING
	Rozsah hodnot:	Názvy všech systémových proměnných NC systému typu STRING
`<Name>`:	Řetězec znaků, příp. název, ke kterému má být zjištěn index systémové proměnné.	
	Datový typ:	STRING
`<NoAlarm>`:	Potlačení alarmů (**nepovinné**)	
	Datový typ:	BOOL
	Předdefinovaná hodnota:	FALSE
	Hodnota	**Význam**
	TRUE	V případě chyby se zpracovávání programu nepozastaví a neaktivuje se žádný alarm. Případ použití: Specifická uživatelská reakce v závislosti na výstupní hodnotě.
	FALSE	V případě chyby (<RetVal> < 0) se zpracovávání programu pozastaví a aktivuje se alarm.
`<RetVal>`:	Index systémové proměnné nebo chybové hlášení	
	Datový typ:	INT
	Rozsah hodnot:	-1 ≤ x ≤ (max. počet komponent nastavený v konfiguraci -1)
	Hodnota	**Význam**
	≥ 0	Hledaný název byl nalezen pod uvedeným indexem systémové proměnné.
	-1	Hledaný název nebyl nalezen, příp. se vyskytla chyba.

Příklad

Programový kód	**Komentář**
`DEF INT INDEX`	
`$NP_PROT_NAME[27] = "Abdeckung"`	
`...`	

Programový kód	Komentář
INDEX = NAMETOINT("$NP_PROT_NAME","Abdeckung")	; INDEX == 27

4.11 Zabránění vzniku kolizí u kinematických řetězců

Poznámka

Chráněné oblasti

Chráněné oblasti zmiňované v následujících kapitolách se vztahují na funkci "Geometrické modelování stroje".

Pokud budete potřebovat další informace o této funkci, viz Příručka k funkcím, Monitorování a kompenzace.

4.11.1 Kontrola dvojice oblastí chráněných kvůli kolizi (COLLPAIR)

4.11.2 Vyvolání nového výpočtu modelu stroje pro prevenci kolizí (PROTA)

Jestliže jsou ve výrobním programu zapsány systémové proměnné kinematického řetězce $NK_..., proměnné pro geometrické modelování stroje nebo pro prevenci kolizí $NP_..., musí být potom vyvolána procedura PROTA, aby se změna interního modelu stroje pro prevenci kolizí uvnitř NC systému vstoupila v platnost.

Syntaxe

```
PROTA[(<Par>)]
```

Význam

PROTA:	Vyvolání nového výpočtu modelu stroje pro prevenci kolizí		
	• Spouští zastavení předběžného zpracování. • Musí se nacházet v samostatném bloku.		
<Par>:	Parametr (**volitelný**)		
	Datový typ:	STRING	
	Hodnota:	---	Bez parametru- Uskuteční se nový výpočet modelu stroje. Stavy chráněných oblastí zůstávají zachovány.
		"R"	Uskuteční se nový výpočet modelu stroje. Chráněné oblasti se nastaví do svého stavu po inicializaci v souladu s parametrem $NP_INIT_STAT.

Okrajové podmínky

Simulace

Procedura `PROTA` se ve výrobních programech nesmí používat v souvislosti se simulací (simNC).

Příklad: Zabránění volání příkazu `PROTA`, když je aktivní simulace.

Programový kód	**Komentář**
`...`	
`IF $P_SIM == FALSE`	`; IF simulace není aktivní`
`   PROTA`	`;  THEN nově vypočítat protikolizní model`
`ENDIF`	`; ENDIF`
`...`	

Viz také

Nastavení stavu chráněné oblasti (PROTS) (Strana 730)

4.11.3 Nastavení stavu chráněné oblasti (PROTS)

Procedura `PROTS(...)` nastavuje stav chráněných oblastí na zadanou hodnotu.

Syntaxe

```
PROTS(<State>[,<Name_1>,...,<Name_n>])
```

Význam

`PROTS`:	Nastavení stavu chráněných oblastí		
	• Musí se nacházet v samostatném bloku.		
`<State>`:	Status, do kterého má být zadaná chráněná oblast nastavena		
	Datový typ:	CHAR	
	Hodnota:	"A" nebo "a"	Status: Aktivní
		"I" nebo "i"	Status: Neaktivní
		"P" nebo "p"	Status: Předběžně aktivováno, příp. řízeno PLC [1)]
		"R" nebo "r"	Status: Interní hodnota stavu po inicializaci v NC systému [2)]

<Name_1> ... <Name_n>:	Název jedné nebo více chráněných oblastí, které mají být nastaveny do zadaného stavu (**volitelné**) Pokud není zadán žádný název, bude zadaný status nastaven pro všechny definované chráněné oblasti.	
	Datový typ:	STRING
	Rozsah hodnot:	Názvy chráněných oblastí, které mohou být nastaveny v parametrech
	Upozornění: Maximální počet chráněných oblastí, které mohou být zadány jako parametr, závisí pouze na maximálním možném počtu znaků na jeden řádek programu.	
[1] Aktivování / deaktivování se uskutečňuje prostřednictvím: DB10.DBX234.0 - DBX241.7 [2] Status se nastavuje na interní hodnotu NC systému po inicializaci, tzn. na hodnotu, kterou měla systémová proměnná $NP_INIT_STAT v okamžiku posledního vyvolání funkce PROTA(...) (Strana 729).		

4.11.4 Stanovení vzdálenosti mezi dvěma chráněnými oblastmi (PROTD)

Funkce PROTD(...) vypočítává vzdálenost dvou chráněných oblastí.

Vlastnosti funkce:

- Výpočet vzdálenosti se uskutečňuje nezávisle na stavu chráněné oblasti (aktivovaná, deaktivovaná, předběžně aktivovaná).
- Pro výpočet vzdálenosti dvou chráněných oblastí se používají jen takové prvky chráněných oblastí, pro které je přiřazeno $NP_USAGE = „C“ nebo „A“. Prvky chráněných oblastí, které mají přiřazeno $NP_USAGE = „V“, nejsou uvažovány.
- Chráněné oblasti, jejichž všechny prvky mají přiřazeno $NP_USAGE = „V“, nemohou být do výpočtu vzdálenosti zahrnuty.
- Výpočet vzdálenosti se uskutečňuje s polohou, která byla platná na konci předcházejícího bloku.
- Superponované pohyby, které jsou ve zpracování hlavní větve započítávány, např. posunutí DRF nebo externí posunutí počátku, jsou do výpočtu vzdálenosti začleněny, přičemž se berou hodnoty, které jsou platné v **okamžiku interpretace** funkce.

Poznámka

Synchronizace

Při použití funkce PROTD(...) je výlučnou odpovědností uživatele, aby v případě nutnosti prostřednictvím zastavení předběžného zpracování pomocí příkazu STOPRE zajistil synchronizaci hlavního a předběžného zpracování.

Kolize

Pokud mezi zadanými chráněnými oblastmi existuje kolize, je výsledkem funkce vzdálenost 0,0. Kolize existuje, jestliže se obě chráněné oblasti dotýkají nebo existuje jejich průnik.

Bezpečnostní vzdálenost pro kontrolu kolize (MD10622 $MN_COLLISION_SAFETY_DIST) se při výpočtu vzdálenosti nebere v úvahu.

Syntaxe

```
[<RetVal> =] PROTD([<Name_1>],[<Name_2>],VAR <Vector>[,<System>])
```

Význam

PROTD:	Výpočet vzdálenosti mezi dvěma zadanými chráněnými oblastmi		
	• Musí se nacházet v samostatném bloku.		
<RetVal>:	Výsledná hodnota funkce: Absolutní hodnota vzdálenosti dvou chráněných oblastí nebo v případě kolize 0,0 (viz výše: přesah při kolizi)		
	Datový typ:	REAL	
	Rozsah hodnot:	0,0 ≤ x ≤ +max. hodnota REAL	
<Name_1>, <Name_2>:	Názvy obou chráněných oblastí, jejichž vzdálenost vůči sobě má být vypočítána (**nepovinné**).		
	Datový typ:	STRING	
	Rozsah hodnot:	Názvy chráněných oblastí, které mohou být nastaveny v parametrech	
	Předdefinovaná hodnota:	„“ (prázdný řetězec) Pokud nejsou uvedeny žádné chráněné oblasti, funkce vypočítá aktuální nejmenší vzdálenost ze všech aktivovaných a deaktivovaných chráněných oblastí, které jsou obsaženy v protikolizním modelu.	
<Vector>:	Výsledná hodnota: 3-rozměrný vektor vzdálenosti od chráněné oblasti <Name_2> ke chráněné oblasti <Name_1>, přičemž: • <Vector>[0]: souřadnice X v globálním souřadném systému • <Vector>[1]: Souřadnice Y v globálním souřadném systému • <Vector>[2]: Souřadnice Z v globálním souřadném systému V případě kolize: <Vector> == nulový vektor		
	Datový typ:	VAR REAL [3]	
	Rozsah hodnot:	<Vector> [n]: 0,0 ≤ x ≤ ±max. hodnota typu REAL	
<System>:	Systém měřicích jednotek (palce/metrický systém) pro vzdálenost a vektor vzdálenosti (**nepovinný**)		
	Datový typ:	BOOL	
	Hodnota:	FALSE (předdefinované nastavení)	Systém měřicích jednotek v souladu s momentálně aktivním příkazem G-funkce ze skupiny 13 (G70, G71, G700, G710)).
		TRUE	Systém měřicích jednotek odpovídá nastavenému základnímu systému: MD52806 $MN_ISO_SCALING_SYSTEM

4.12 Transformace pomocí kinematických řetězců

4.12.1 Aktivování transformace (TRAFOON)

Transformace definovaná prostřednictvím kinematického řetězce se aktivuje pomocí předem definované procedury TRAFOON. Volání se musí nacházet v samostatném bloku.

Poznámka

Další možností, kterou můžete využít, je aktivovat transformaci definovanou pomocí kinematického řetězce pomocí konvenčního příkazu jazyka, jako např. TRAORI nebo TRANSMIT. Za tím účelem musí být v systémové proměnné $NT_TRAFO_INDEX uložena odpovídající hodnota nerovnající se nule.

Pokud budete potřebovat další informace týkající se proměnné $NT_TRAFO_INDEX, viz „Příručka pro sestavy, Systémové proměnné".

Syntaxe

```
TRAFOON(<Trafoname>,<Diameter>,<k>)
```

Význam

TRAFOON:	Procedura pro aktivování transformace definované prostřednictvím kinematického řetězce		
<Trafoname>:	Název datového bloku transformace		
	Datový typ:	STRING	
	Rozsah hodnot:	Všechny názvy datových bloků transformace definované pomocí proměnné $NT_NAME	
	Poznámka: Název datového bloku transformace musí být jedinečný. V proměnné $NT_NAME se smí vyskytnout jen jednou.		
<Diameter>:	Vztažný, příp. pracovní průměr(pouze TRACYL)		
	Datový typ:	REAL	
	Hodnota musí být > 1.		
<k>:	Definice použití korekce stěny drážky (jen TRACYL).		
	Datový typ:	BOOL	
	Hodnota:	FALSE	bez korekce stěny drážky
		TRUE	s korekcí stěny drážky
	Odpovídá typu transformace TRACYL 514 (programovatelná korekce stěny drážky). Pokud parametr <k> není uveden, uplatňuje se nastavení v bitu 10 parametru $NT_CNTRL[<n>].		

Příklad

Programový kód	Komentář
`TRAFOON["Trans_1"]`	`Aktivování transformace s názvem Trans_1.`

4.12.2 Modifikace transformace orientace po měření stroje (CORRTRAFO)

U strojů s transformacemi orientace, které byly definovány prostřednictvím kinematických řetězců, může uživatel používat předem definovanou funkci CORRTRAFO, která po měření stroje upravuje offsetové vektory nebo směrové vektory orientačních os v kinematickém modelu stroje.

Poznámka

Hodnoty korekcí, které byly zapsány pomocí funkce CORRTRAFO, nejsou v transformaci v platnosti okamžitě. Začnou se uplatňovat až po deaktivování transformace, vyvolání příkazu NEWCONF a po opětovném aktivování transformace.

Syntaxe

```
<Corr_Status> = CORRTRAFO(<Corr_Vect>, <Corr_Index>, <Corr_Mode>,
[ <No_Alarm>])
```

Význam

CORRTRAFO:	Volání funkce

<Corr_Status>:	Výsledná hodnota funkce		
	Datový typ:	INT	
	Hodnoty:	0	Funkce byla zpracována bezchybně.
		1	Není aktivní žádná transformace.
		2	Momentálně aktivní transformace není transformací orientace.
		3	Aktivní transformace orientace nebyla definována prostřednictvím kinematického řetězce.
		10	Parametr <Corr_Index> sloužící při volání je záporný.
		11	Parametr <Corr_Mode> sloužící při volání je záporný.
		12	Neplatný odkaz na dílčí úsek v řetězci znaků (místo jednotek v parametru <Corr_Index>). Hodnota nesmí být větší než počet orientačních os v dílčím řetězci.
		13	Neplatný odkaz na orientační osu v dílčím řetězci (místo jednotek v parametru <Corr_Index>). Tato hodnota musí být menší než počet orientačních os v dílčím řetězci.
		14	Neplatný odkaz na řetězec znaků (místo desítek v parametru <Corr_Index>). Přípustné jsou pouze hodnoty 0 a 1 (odkaz na řetězec součásti nebo nástroje). Toto číslo chyby se vyskytne také tehdy, když dílčí řetězec, na který odkazuje parametr <Corr_Index>, neexistuje.
		15	V úseku, na který odkazuje parametr <Corr_Index>, není definován žádný prvek korekce ($NT_CORR_ELEM_P, příp. $NT_CORR_ELEM_T).
		20	Neplatný režim korekce (místo jednotek v parametru <Corr_Mode>). Přípustné jsou pouze hodnoty 0 a 1.
		21	Neplatný režim korekce (místo desítek a/nebo místo stovek v parametru <Corr_Mode>). Při zápisu směru osy se smí pouze místo jednotek nerovnat nule.
		30	Místo stovek parametru <Corr_Mode> je neplatné. Přípustné jsou pouze hodnoty 0 a 1.
		31	Místo tisíců parametru <Corr_Mode> je neplatné. Přípustné jsou pouze hodnoty 0 a 1.
		40	Směrový vektor, který se má převzít jako směr osy, je nulový vektor. Tato chyba se může vyskytnout jen tehdy, pokud je místo tisíců v parametru <Corr_Mode> rovno nule. Jestliže se místo tisíců v tomto parametru rovná 1 (monitorování maximální korekce je deaktivováno), je možno zapsat i nulový vektor.
		41	U korekce offsetového vektoru je odchylka oproti aktuální hodnotě u minimálně jedné souřadnice větší, než je předem stanovená maximální hodnota definovaná v nastavovaném parametru SD41610 $SN_CORR_TRAFO_LIN_MAX. Parametr <Corr_Vect> se přepisuje chybovým vektorem. To platí i tehdy, když je zpracování přerušeno s alarmem (viz parametr <No_Alarm>). Chybový vektor obsahuje složky, jejichž hodnota korekce překračuje přípustnou mezní hodnotu, tedy rozdíl mezi zjištěnou hodnotou korekce a mezní hodnotou se správným znaménkem. Obsah složek, které svou mezní hodnotu nepřekračují, je nulový.
		42	U korekce směrového vektoru je úhlová odchylka oproti aktuálnímu směru větší, než je předem stanovená maximální hodnota

			definovaná v nastavovaném parametru SD41611 $SN_CORR_TRAFO_DIR_MAX.
		43	Pokus o zápis do systémové proměnné byl kvůli chybějícímu oprávnění k zápisu zablokován.
<Corr_Vect>:	Vektor korekce Obsah vektoru korekce je definován prostřednictvím následujících parametrů <Corr_Index> a <Corr_Mode>. Pokud je <Corr_Status> = 41, obsah vektoru se přepisuje (viz výše.).		
	Datový typ:	REAL	
<Corr_Index>:	Úsek, jehož prvek korekce má být modifikován / Index orientační osy, jejíž směrový vektor má být modifikován.		
	Datový typ:	INT	
	U parametru <Corr_Index> se používá desítkové kódování (místo jednotek až desítek):		
	Místo jednotek:	Obsahuje index úseku, příp. orientační osy v dílčím řetězci.	
	Místo desítek:	Odkazuje na dílčí řetězec.	
		0x	Řetězec obrobku
		1x	Řetězec nástroje

<Corr_Mode>:	Režim korekce		
	Datový typ:	INT	
	U parametru <Corr_Mode> se používá desítkové kódování (místo jednotek až tisíců):		
	Místo jednotek:	Určuje, u kterého prvku má být korekce provedena.	
		xxx**0**	Korekce lineárního offsetového vektoru
		xxx**1**	Korekce směrového vektoru orientační osy
	Místo desítek:	Určuje, jak má být prvek korekce, na který odkazuje obsah parametru <Corr_Index>, modifikován.	
		xx**0**x	Vektor korekce se okamžitě zapisuje do prvku korekce. Tato varianta se může použít, aby se prvek korekce zapisoval okamžitě, aniž by musel být znám index <n> příslušného systémového parametru ($NK_OFF_DIR[<n>, ...]).
		xx**1**x	Stejně jako u hodnoty 0, ale s tím rozdílem, že předávaná hodnota korekce je interpretována v globálních souřadnicích. Rozdíl mezi variantou 0 a 1může vzniknout vždy, když kinematický řetězec v základním nastavení (pozice všech orientačních os se rovnají 0) obsahuje další otočení.
		xx**2**x	Stejné jako 1, ale s tím rozdílem, že hodnota korekce se vztahuje na celý úsek, tzn. že v prvku korekce je uložena taková hodnota, že celý úsek dosahuje délky definované hodnotou korekce.
		Poznámka: Hodnoty 1 a 2 jsou při zápisu směrového vektoru orientační osy nepřípustné.	
	Místo stovek:	Určuje, jak má být obsah parametru <Corr_Vect> interpretován.	
		x**0**xx	Předávaný vektor korekce <Corr_Vect> obsahuje celkovou novou délku prvku korekce, příp. úseku, na který odkazuje parametr <Corr_Index> ve spojení s místem desítek parametru <Corr_Mode> (absolutní korekce).
		x**1**xx	Předávaný vektor korekce <Corr_Vect> obsahuje jen rozdíl oproti aktuální délce prvku korekce, příp. úseku, na který odkazuje parametr <Corr_Index> ve spojení s místem desítek parametru <Corr_Mode> (inkrementální korekce).
		Poznámka: U korekce směrového vektoru orientační osy musí místo stovek obsahovat hodnotu 0.	
	Místo tisíců:	Určuje, zda má být korekce omezena prostřednictvím následující maximální hodnoty: • SD41610 $SN_CORR_TRAFO_LIN_MAX nebo • SD41611 $SN_CORR_TRAFO_DIR_MAX	
		0xxx	Monitorování maximální korekce je aktivní.
		1xxx	Monitorování maximální korekce není aktivní.

<No_Alarm>:	Chování v případě chyby (vracená hodnota > 0) (**volitelné**)		
	Datový typ:	BOOL	
	Hodnota:	FALSE (předdefinované)	Pokud se vyskytne chyba, zpracování programu se pozastaví a aktivuje se alarm 14103.
		TRUE	V případě chyby se zpracovávání programu nepozastaví a neaktivuje se žádný alarm. Případ použití: Specifická uživatelská reakce v závislosti na výstupní hodnotě.

Poznámka

Jestliže se při volání funkce vyskytne chyba, buď se aktivuje alarm nebo se na výstupu objeví číslo chyby (viz parametr <No_Alarm>), takže uživatel může sám vhodným způsobem na chybový stav reagovat. Příčinu chyby blíže popisuje alarmový parametr. Číslo chyby vracené místo alarmu je identické s alarmovým parametrem.

Další informace k funkci CORRTRAFO

Kinematická struktura stroje s transformací orientace je popsána prostřednictvím jednoho nebo dvou kinematických řetězců (dílčí řetězce), které vycházejí z počátku globálního souřadného systému. Jeden z těchto dvou řetězců, **řetězec nástroje**, končí ve vztažném bodě nástroje, zatímco ten druhý, **řetězec obrobku**, končí v počátku základního souřadného systému.

Funkce CORRTRAFO zapisuje do speciálních korekčních prvků délky ramen pák a směry os u strojů s transformací orientace. Kinematický řetězec je mimo jiné popsán prvky typu OFFSE, které jsou definovány prostřednictvím proměnné $NK_TYPE.

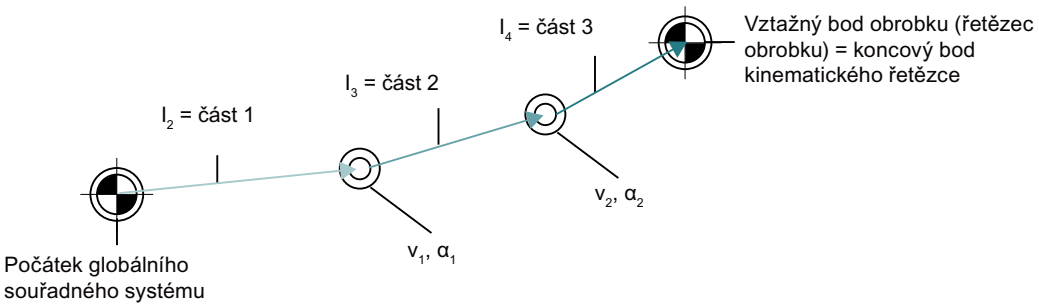
Funkce CORRTRAFO pracuje s úseky

Oba dílčí řetězce mohou být rozděleny na maximálně čtyři úseky:

- Úsek 1 začíná v počátečním bodě řetězce a končí v první ose orientace.
- Úsek 2 je vzdálenost mezi orientační osou 1 a orientační osou 2.
- Úsek 3 je vzdálenost mezi orientační osou 2 a orientační osou 3.
- Úsek 4 je vzdálenost mezi orientační osou 3 a koncem nástrojového nebo obrobkového řetězce.

Úseky mohou obsahovat příslušné konstantní prvky řetězce typu OFFSET nebo ROT_CONST.

Následující obrázek znázorňuje transformaci orientace se 2 orientačními osami.



Obrázek 4-8 CORRTRAFO - příklad

Úseky jsou definovány jednoznačně: Když budete procházet kinematickým řetězcem od jeho počátečního bodu do koncového bodu, má první úsek index 0, následující úsek index 1 atd. Index posledního úseku je pak vždy roven počtu orientačních os.

Prvky korekce

Na kterýkoli konstantní prvek kinematického řetězce (prvek řetězce typu $NK_TYPE[<n>] = "OFFSET“) v každém tomto úseku je možno se odkázat pomocí systémových proměnných $NT_CORR_ELEM_T[<n>, 0 ... 3], příp. $NT_CORR_ELEM_P[<n>, 0 ... 3]. Do takto označených prvků je možno pomocí funkce CORRTRAFO zapsat hodnoty korekcí, které byly zjištěny při měření stroje.

Příklad s indexem transformace = 1:

- $NT_CORR_ELEM_T[1,0] = „C_AXIS_OFFSET“; Offset osy C (orientační osa 1) v úseku 1 je definován jako korekční prvek.
- $NT_CORR_ELEM_T[1,1] = "B_AXIS_OFFSET"; Offset osy B (orientační osa 2) v úseku 2 je definován jako korekční prvek.
- $NT_CORR_ELEM_T[1,2] = "BASE_TOOL_OFFSET"; Offset osy B až po vztažný bod nástroje v úseku 3 je definován jako korekční prvek.

Posloupnost odkazů v proměnné $NT_CORR_ELEM_T/P[<n>, 0 ... 3] musí korespondovat s výše popsanými úseky, tzn. v proměnné $NT_CORR_ELEM_T/P [<n>, 0] se může nacházet jen prvek řetězce, který leží před první orientační osou atd.

Funkce CORRTRAFO zapisuje do takto definovaných korekčních prvků hodnoty, které byly stanoveny prostřednictvím měření stroje. Úpravy hodnot korekcí jsou ve funkci CORRTRAFO definovány prostřednictvím parametru <Corr_Mode>.

Zavření řetězce

Jestliže je v systémové proměnné $NT_CNTRL[<n>] nastaven bit 7 nebo bit 8, jsou na konci řetězce obrobku (bit 7), příp. před počátečním bodem řetězce nástroje (bit 8) interně automaticky vkládány doplňkové konstantní prvky řetězce, které vytvářejí spojení od koncového bodu řetězce k počátku souřadného systému stroje („zavření řetězce“).

Do těchto automaticky vkládaných prvků není možné externě nic zapisovat, lze je pouze číst (podrobnosti viz systémové proměnné $AC_TRAFO_CORR_ELEM_P/T).

Bod pro uzavření řetězce nástroje

Jestliže systémová proměnná $NT_CLOSE_CHAIN_T není prázdná, zavírá se řetězec nástroje nikoli v koncovém bodě řetězce, ale v koncovém bodě popisovaného prvku řetězce. Další prvky řetězce, které se nacházejí za tímto bodem, mají potom při aktivování transformace za následek odpovídající posunutí počátku.

Index orientační osy

Kromě konstantních offsetů mezi orientačními osami je možno pomocí funkce CORRTRAFO popsat také směrové vektory orientačních os. Pod pojmem index orientační osy se v tomto případě rozumí index, který je přiřazen, když se prochází kinematickým řetězcem od jeho počátku až do konce, přičemž počítání začíná nulou. Index orientační osy je proto vždy roven indexu předcházejícího úseku.

Index orientační osy je možno také zjistit pomocí systémové proměnné $AC_TRAFO_ORIAX_LOC.

Maximální přípustná změna prvku řetězce

Maximální přípustná změna prvku řetězce může být omezena pomocí dvou nastavovaných parametrů, a to SD41610 $SN_CORR_TRAFO_LIN_MAX pro offsetové vektory a SD41611 $SN_CORR_TRAFO_DIR_MAX pro směrové vektory orientačních os. SD41610 $SN_CORR_TRAFO_LIN_MAX udává absolutní hodnotu, o kterou se každá jednotlivá složka vektoru může maximálně změnit oproti své vztažné hodnotě. SD41611 $SN_CORR_TRAFO_DIR_MAX udává, o jaký úhel se smí maximálně změnit směr vektoru osy oproti své vztažné hodnotě. Vztažná hodnota je přitom vždy odpovídající hodnota, která je v platnosti při volání aktivní transformace s funkcí CORRTRAFO. To znamená, že obsah dat kinematiky v datové paměti, který byl eventuálně změněn po aktivování transformace, nemá žádný vliv na způsob, jakým se funkce CORRTRAFO chová.

4.13 Korekční parametry nástroje

4.13.1 Paměť korekcí

Struktura paměti korekčních parametrů

Každé datové pole může být vyvoláno pomocí T- a D-čísel a tato pole obsahují vedle geometrických údajů nástroje ještě i další záznamy, např. typ nástroje.

Uživatelské parametry břitu

Uživatelské parametry břitu mohou být konfigurovány pomocí strojního parametru. Věnujte prosím pozornost informacím od výrobce stroje.

Parametry nástroje

Poznámka

Jednotlivé hodnoty v paměti korekčních parametrů

Jednotlivé hodnoty v paměti korekčních parametrů P1 až P25 je možné číst a zapisovat z programu pomocí systémových proměnných. Všechny zbývající parametry jsou rezervovány.

Korekční parametry nástroje $TC_DP6 až $TC_DP8, $TC_DP10 a $TC_DP11, jakož i $TC_DP15 až $TC_DP17, $TC_DP19 a $TC_DP20 mají v závislosti na typu nástroje odlišný význam.

Číslo parametrů nástroje (DP)	Význam systémových proměnných	Poznámka
$TC_DP1	Typ nástroje	Přehled viz seznam
$TC_DP2	Poloha břitu	jen pro soustružnické nástroje
Geometrie	**Korekce délky**	
$TC_DP3	Délka 1	Započítává se podle
$TC_DP4	Délka 2	typu a roviny
$TC_DP5	Délka 3	
Geometrie	**Rádius**	
$TC_DP6 [1)] $TC_DP6 [2)]	Rádius 1 / délka 1 průměr d	Frézovací/soustružnické/brusné nástroje Drážková pila
$TC_DP7 [1)] $TC_DP7 [2)]	Délka 2 / Rohový rádius kuželové frézy, šířka drážky b, rohový rádius	Frézovací nástroje Drážková pila
$TC_DP8 [1)] $TC_DP8 [2)]	Rádius zaoblení 1 pro frézovací nástroje Přesah k	Frézovací nástroje Drážková pila
$TC_DP9 [1) 3)]	Rádius zaoblení 2	rezervováno
$TC_DP10 [1)]	Úhel 1 čelní strany nástroje	Kuželové frézovací nástroje
$TC_DP11 [1)]	Úhel 2 podélné osy nástroje	Kuželové frézovací nástroje
Opotřebení	**Korekce délky a rádiusu**	
$TC_DP12	Délka 1	
$TC_DP13	Délka 2	
$TC_DP14	Délka 3	
$TC_DP15 [1)] $TC_DP15 [2)]	Rádius 1 / délka 1 průměr d	Frézovací/soustružnické/brusné nástroje Drážková pila
$TC_DP16 [1)] $TC_DP16 [3)]	Délka 2 / Rohový rádius kuželové frézy, šířka drážky b, rohový rádius	Frézovací nástroje Drážková pila
$TC_DP17 [1)] $TC_DP17 [2)]	Rádius zaoblení 1 pro frézovací nástroje Přesah k	Frézování / 3D frézování na čelní ploše Drážková pila
$TC_DP18 [1) 3)]	Rádius zaoblení 2	rezervováno
$TC_DP19[1)]	Úhel 1 čelní strany nástroje	Kuželové frézovací nástroje
$TC_DP20[1)]	Úhel 2 podélné osy nástroje	Kuželové frézovací nástroje
Základní rozměr / adaptér	**Korekce délky**	

Číslo parametrů nástroje (DP)	Význam systémových proměnných	Poznámka
$TC_DP21	Délka 1	
$TC_DP22	Délka 2	
$TC_DP23	Délka 3	
Technologie		
$TC_DP24	Úhel volného řezání	jen pro soustružnické nástroje
$TC_DP25		rezervováno

1) Platí i u frézovacích nástrojů pro 3D frézování na čelní ploše

2) U nástroje typu drážková pila

3) rezervováno

Poznámky:

Pro geometrické veličiny (např. Délka 1 nebo Rádius) existuje několik vstupních komponent. Tyto komponenty se aditivně započítávají k výsledné veličině (např. celková délka 1, celkový rádius), která se potom uplatňuje při obrábění.

Do nepotřebných parametrů nástroje je nutno dosadit nulu.

Parametry nástroje $TC-DP1 až $TC-DP23 u nástrojů pro obrábění kontur

Poznámka

Parametry nástroje, které v tabulce nejsou uvedeny, jako např. $TC_DP7, nejsou vyhodnocovány, tzn. jejich obsah nemá žádný význam.

Číslo parametrů nástroje (DP)	Význam	Břity Dn		Poznámka
$TC_DP1	Typ nástroje			400 až 599
$TC_DP2	Poloha břitu			
Geometrie	**Korekce délky**			
$TC_DP3	Délka 1			
$TC_DP4	Délka 2			
$TC_DP5	Délka 3			
Geometrie	**Rádius**			
$TC_DP6	Rádius			
Geometrie	**Mezní úhel**			
$TC_DP10	Minimální mezní úhel			
$TC_DP11	maximální mezní úhel			
Opotřebení	**Korekce délky a rádiusu**			
$TC_DP12	Opotřebení – délka 1			
$TC_DP13	Opotřebení – délka 2			
$TC_DP14	Opotřebení – délka 3			
$TC_DP15	Rádius opotřebení			

Číslo parametrů nástroje (DP)	Význam	Břity Dn		Poznámka
Opotřebení	**Mezní úhel**			
$TC_DP19	Opotřebení - minimální mezní úhel			
$TC_DP20	Opotřebení - maximální mezní úhel			
Základní rozměr / adaptér	**Korekce délky**			
$TC_DP21	Délka 1			
$TC_DP22	Délka 2			
$TC_DP23	Délka 3			

Základní hodnota a hodnota opotřebení

Výsledné veličiny jsou vždy vypočítávány jako součet základní hodnoty a hodnoty opotřebení (např. $TC_DP6 + $TC_DP15 pro rádius). K délce prvního břitu nástroje se kromě toho přičítá ještě základní rozměr ($TC_DP21 – $TC_DP23). Kromě toho se u těchto délek nástroje uplatňují ještě i další veličiny, které i v případě běžných nástrojů mohou ovlivňovat jejich efektivní délku (adaptér, orientovatelný držák nástroje, nastavované parametry).

Mezní úhel 1 a 2

Mezní úhel 1, příp. mezní úhel 2, se vztahují vždy na vektor od středu břitu ke vztažnému bodu břitu a měří se proti směru hodinových ručiček.

4.13.2 Aditivní korekce

4.13.2.1 Aktivování aditivních korekcí (DL)

Na aditivní korekce je možné pohlížet jako na procesní korekce programovatelné v průběhu obrábění. Vztahují se na geometrické údaje o břitu a tvoří tedy součást údajů o břitech nástroje.

Údaje aditivních korekcí jsou adresována pomocí DL-čísla (DL: Location Dependent; korekce vztažená na určité místo použití) a zadávají se prostřednictvím uživatelského rozhraní.

Aplikace

Prostřednictvím aditivních korekcí mohou být kompenzovány chyby rozměrů podmíněné místem použití.

Syntaxe

```
DL=<číslo>
```

Význam

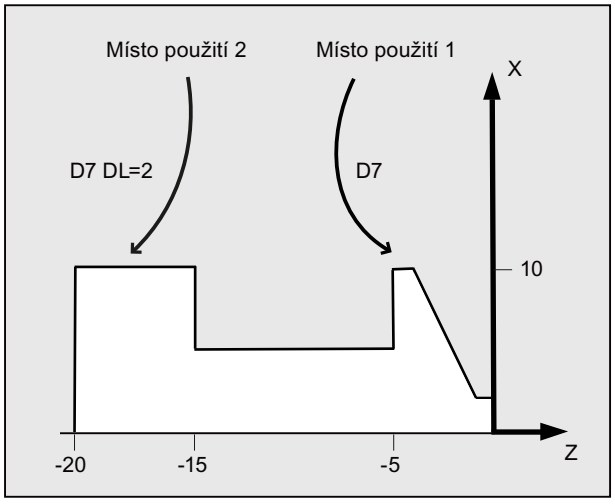
`DL`:	Příkaz pro aktivování aditivní korekce
`<číslo>`:	Prostřednictvím parametru `<číslo>` se zadává blok aditivních korekčních parametrů nástroje, který má být aktivován.

Poznámka

Definice počtu a aktivování aditivních korekcí se uskutečňuje prostřednictvím strojních parametrů (--> Věnujte prosím pozornost informacím od výrobce stroje!).

Příklad:

Pro 2 ložisková sedla se použije stejný břit:



Programový kód	**Komentář**
`N110 T7 D7`	`; revolver se nastaví na místo č. 7 Aktivují se D7 a DL=1 a v následujícím bloku se použijí.`
`N120 G0 X10 Z1`	
`N130 G1 Z-6`	
`N140 G0 DL=2 Z-14`	`; aditivně k D7 se aktivuje DL=2 a v následujícím bloku se použijí`
`N150 G1 Z-21`	
`N160 G0 X200 Z200`	`; najíždění na bod pro výměnu nástroje`
`...`	

4.13.2.2 Definice opotřebení a seřizovacích hodnot ($TC_SCPxy[t,d], $TC_ECPxy[t,d])

4.13.2.3 Vymazání aditivních korekcí (DELDL)

Příkazem `DELDL` se provádí mazání aditivních korekcí břitů nástroje (vymazání z paměti). Přitom se vymažou jak definované hodnoty opotřebení, tak i seřizovací hodnoty.

Syntaxe

```
DELDL[<t>,<d>]
```

```
DELDL[<t>]
DELDL
<Status>=DELDL[<t>,<d>]
```

Význam

DELDL:	Příkaz pro vymazání aditivních korekcí	
<t>:	T-číslo nástroje	
<d>:	D-číslo břitu nástroje	
DELDL[<t>,<d>]:	Vymažou se všechny aditivní korekce břitu <d> nástroje <t>.	
DELDL[<t>]:	Vymažou se všechny aditivní korekce všech břitů nástroje <t>	
DELDL:	Budou vymazány všechny aditivní korekce všech břitů všech nástrojů jednotky TO (pro kanál, ve kterém je příkaz naprogramován).	
<Status>:	Status vymazání	
	Hodnota:	Význam:
	0	Vymazání bylo úspěšně provedeno.
	-	Vymazání nebylo provedeno (pokud nastavení parametrů označuje přesně jeden břit) nebo vymazání nebylo provedeno úplně (pokud nastavení parametrů popisuje více břitů).

Poznámka

Hodnoty opotřebení a seřizovací hodnoty aktivního nástroje není možné vymazat (analogicky k chování při mazání D-čísel a parametrů nástroje).

4.13.3 Korekce nástroje - speciální zacházení

Pomocí nastavovaných parametrů SD 42900 až SD 42960 je možné ovládat vyhodnocování znaménka pro délku a opotřebení nástroje.

To platí jednak pro chování složek opotřebení při zrcadlovém převracení geometrických os, jednak při změně roviny obrábění, ale také pro teplotní kompenzaci ve směru nástroje.

Hodnoty opotřebení

Jestliže jsou v následujících odstavcích zmiňovány hodnoty opotřebení, je tím v každém případě míněn součet vlastních hodnot opotřebení ($TC_DP12 až $TC_DP20) a součtových korekcí s hodnotami opotřebení ($SCPX3 až $SCPX11) a seřizovacími hodnotami ($ECPX3 až $ECPX11).

Pokud budete potřebovat informace o součtových korekčních parametrech, viz Příručka k funkcím, Správa nástrojů.

Nastavované parametry

SD42900 $SC_MIRROR_TOOL_LENGTH	Zrcadlové převracení složek délky nástroje a složek základních rozměrů.
SD42910 $SC_MIRROR_TOOL_WEAR	Zrcadlové převrácení hodnot opotřebení složky délky nástroje.
SD42920 $SC_WEAR_SIGN_CUTPOS	Vyhodnocování znaménka složky opotřebení v závislosti na poloze břitu.
SD42930 $SC_WEAR_SIGN	Invertování znaménka rozměru opotřebení.
SD42935 $SC_WEAR_TRANSFORM	Transformace hodnot opotřebení.
SD42940 $SC_TOOL_LENGTH_CONST	Přiřazení složky délky nástroje geometrické ose.
SD42950 $SC_TOOL_LENGTH_TYPE	Přiřazení složky délky nástroje nezávisle na typu nástroje.
SD42960 $SC_TOOL_TEMP_COMP	Hodnota teplotní kompenzace ve směru nástroje. Funguje, i když je programována orientace nástroje.

Další informace

Aktivování změněných nastavovaných parametrů

Nové vyhodnocení složek nástroje při změně popisovaných nastavovaných parametrů se provádí až tehdy, když je následně aktivován břit nástroje. Pokud je nějaký nástroj už aktivní a vyhodnocování změněných údajů tohoto nástroje se má stát platným, je nutné tento nástroj znovu vyvolat.

Analogický postup se použije v případě, kdy se změní výsledná délka nástroje, protože se změnil stav zrcadlového převrácení osy. Po příkazu zrcadlového převrácení musí být nástroj znovu aktivován, aby se uplatnily změněné složky délky nástroje.

Orientovatelný držák nástroje a nové nastavované parametry

Nastavované parametry SD 42900 až SD 42940 nepůsobí na složky případného aktivního orientovatelného držáku nástroje. Avšak výpočet s orientovatelným držákem nástroje vždy umožňuje pracovat s nástrojem v celé jeho výsledné délce (délka nástroje + opotřebení + rozměr základny nástroje). Při výpočtu výsledné celkové délky se zohledňují všechny změny, jež byly způsobeny nastavovanými parametry; tzn. vektory orientovatelného držáku nástroje jsou nezávislé na rovině opracování.

Poznámka

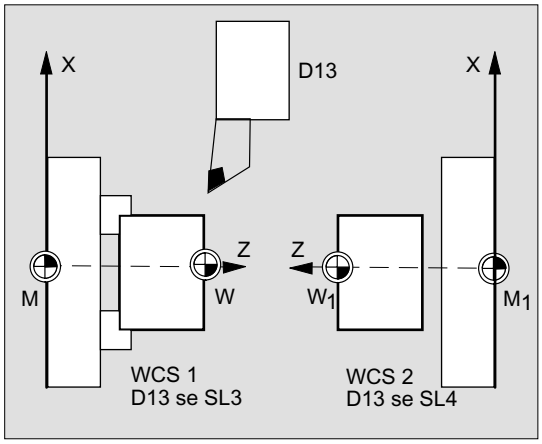
Při použití orientovatelného držáku nástroje je často užitečné definovat všechny nástroje pro základní systém bez zrcadlového převrácení a pak také ty, které se používají jenom při obrábění v zrcadlovém převrácení. Při obrábění se zrcadlově převrácenými osami se pak držák nástroje otočí tak, aby skutečná poloha nástroje byla popisována správně. Všechny složky délky nástroje jsou potom automaticky přiřazeny správným směrům, takže řídící systém si může ušetřit vyhodnocování jednotlivých složek pomocí nastavovaných parametrů v závislosti na stavu zrcadlového převrácení jednotlivých os.

Další možnosti použití

Použití funkce orientovatelného držáku nástroje může být užitečné i tehdy, pokud stroj fyzicky nenabízí žádnou možnost nástroje otáčet do různých poloh, místo toho jsou však pevně instalovány nástroje s různými orientacemi. Při určování rozměrů nástrojů je potom možno pracovat s jednotnou základní orientací a rozměr, který je zapotřebí obrobit, vznikne příslušným otočením virtuálního držáku nástroje.

4.13.3.1 Zrcadlové převrácení délek nástroje

Když je nastavovaným parametrům SD42900 $SC_MIRROR_TOOL_LENGTH a SD42910 $SC_MIRROR_TOOL_WEAR přiřazena nenulová hodnota, můžete provádět zrcadlové převrácení složek délky nástroje a složek základního rozměru odpovídajících příslušným osám.



SD42900 $SC_MIRROR_TOOL_LENGTH

Nastavovaný parametr se **nerovná** nule:

Zrcadlové převrácení – prostřednictvím změny znaménka – se bude týkat složek délky nástroje ($TC_DP3, $TC_DP4 a $TC_DP5) a složek základních rozměrů ($TCDP21, $TC_DP22 a $TC_DP23), jejichž osy jsou zrcadlově převraceny.

Zrcadlové převrácení se **netýká** hodnot opotřebení. Pokud mají být převraceny také, musí být aktivován nastavovaný parametr SD42910 $SC_MIRROR_TOOL_WEAR.

SD42910 $SC_MIRROR_TOOL_WEAR

Nastavovaný parametr se **nerovná** nule:

Hodnoty opotřebení složek délky nástroje, pro něž jsou příslušné osy zrcadlově převraceny, se budou změnou znaménka zrcadlově převracet také.

4.13.3.2 Vyhodnocování znaménka opotřebení

Když je nastavovaným parametrům SD42920 $SC_WEAR_SIGN_CUTPOS a SD42930 $SC_WEAR_SIGN přiřazena nenulová hodnota, můžete invertovat vyhodnocování znaménka u složek opotřebení.

SD42920 $SC_WEAR_SIGN_CUTPOS

Nastavovaný parametr se **nerovná** nule:

U nástrojů s relevantní polohou břitu (soustružnické a brusné nástroje – typy nástrojů 400) závisí vyhodnocování znaménka složek opotřebení v rovině obrábění na poloze břitu. U typů nástrojů bez relevantní polohy břitu nemá tento nastavovaný parametr žádný význam.

V následující tabulce jsou značkou X značeny rozměry, jejichž znaménko je nastavovaným parametrem SD42920 (různým od 0) invertováno.

Poloha břitu	Délka 1	Délka 2
1		
2		X
3	X	X
4	X	
5		
6		
7		X
8	X	
9		

Poznámka

Vyhodnocování znaménka prostřednictvím parametrů SD42920 a SD42910 jsou na sobě nezávislá. Jestliže se např. změní znaménko některého rozměru pomocí obou nastavovaných parametrů, zůstane výsledné znaménko nezměněno.

SD42930 $SC_WEAR_SIGN

Nastavovaný parametr se **nerovná** nule:

Znaménka všech rozměrů opotřebení jsou invertována. Tato změna se týká jak délky nástroje, tak i zbývajících veličin, jako jsou rádius nástroje, rádius zaoblení atd.

Jestliže je zadán kladný rozměr opotřebení, nástroj se stane „kratším“ a „tenčím“, viz "Korekční parametry nástroje – zvláštní zacházení"; kapitola „Aktivování změněných nastavovaných parametrů“.

4.13.3.3 Souřadný systém aktivního obrábění (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)

V závislosti na kinematice stroje nebo možnosti použití orientovatelného držáku nástroje se hodnoty opotřebení změřené v jednom z těchto souřadných systémů převádějí nebo transformují do vhodného souřadného systému.

Souřadné systémy momentálně probíhajícího obrábění

Korekce délky nástroje mohou vycházet z následujících souřadných systémů, které se mohou používat pro započítávání složky délky nástroje „opotřebení" do aktivního nástroje pomocí odpovídající skupiny příkazů G-funkcí č. 56.

- Souřadný systém stroje (MCS)
- Základní souřadný systém (BCS)
- Souřadný systém obrobku (WCS)
- Souřadný systém nástroje (TCS)
- Souřadný systém obrobku kinematické transformace (KCS)

Syntaxe

```
TOWSTD
TOWMCS
TOWWCS
TOWBCS
TOWTCS
TOWKCS
```

Význam

TOWSTD:	Počáteční nastavení pro korekce hodnoty opotřebení v délce nástroje
TOWMCS:	Korekce délky nástroje v MCS
TOWWCS:	Korekce délky nástroje ve WCS
TOWBCS:	Korekce délky nástroje v BCS
TOWTCS:	Korekce délky nástroje na vztažný bod držáku nástroje (orientovatelný držák nástroje)
TOWKCS:	Korekce délky nástroje na hlavu nástroje (kinematická transformace)

Další informace

Rozlišovací charakteristiky

V následující tabulce jsou shrnuty nejdůležitější rozlišovací charakteristiky:

Příkaz G-funkce	Hodnota opotřebení	Aktivní orientovatelný držák nástroje
TOWSTD	Hodnota základního nastavení, délka nástroje	Hodnoty opotřebení podléhají otáčení.
TOWMCS	Hodnoty opotřebení v MCS. TOWMCS a TOWSTD jsou identické, pokud není aktivní žádný orientovatelný držák nástroje.	Otáčí se pouze vektor výsledné délky nástroje bez ohledu na opotřebení.
TOWWCS	Hodnota opotřebení ve WCS se přepočítává na MCS.	Vektor nástroje se počítá bez ohledu na opotřebení stejně jako u TOWMCS.

TOWBCS	Hodnota opotřebení v BCS se přepočítává na MCS.	Vektor nástroje se počítá bez ohledu na opotřebení stejně jako u TOWMCS.
TOWTCS	Hodnota opotřebení v souřadném systému nástroje se přepočítává na MCS.	Vektor nástroje se počítá bez ohledu na opotřebení stejně jako u TOWMCS.

TOWWCS, TOWBCS, TOWTCS: Vektor opotřebení se přičítá k vektoru nástroje.

Lineární transformace

Délka nástroje je v MCS smysluplně definovatelná jen tehdy, pokud MCS vychází z BCS prostřednictvím lineární transformace.

Nelineární transformace

Pokud je např. pomocí TRANSMIT aktivována nelineární transformace, pak bude-li jako požadovaný souřadný systém zadán MCS, automaticky se použije BCS.

Žádná kinematická transformace a žádný orientovatelný držák nástroje

Pokud nejsou aktivní ani kinematická transformace a ani orientovatelný držák nástroje, potom s výjimkou WCS jsou všechny čtyři zbývající souřadné systémy shodné. V důsledku toho se pouze WCS odlišuje od zbývajících. Protože se mají vyhodnocovat pouze délky nástroje, nemají posunutí mezi jednotlivými souřadnými systémy žádný význam.

Započítávání hodnot opotřebení

Nastavovaný parametr **SD42935 $SC_WEAR_TRANSFORM** definuje, které z následujících tří složek opotřebení:

- Opotřebení
- Součtová korekce jemná
- Součtová korekce hrubá

mají podléhat rotaci pomocí transformace adaptéru nebo orientovatelného držáku nástroje, pokud je aktivní některý z následujících příkazů G-funkcí.

- TOWSTD
  Základní nastavení. Pro korekce hodnoty ve směru délky nástroje.
- TOWMCS
  Hodnoty opotřebení v souřadném systému stroje (MCS).
- TOWWCS
  Hodnoty opotřebení v souřadném systému obrobku (WCS).
- TOWBCS
  Hodnoty opotřebení v základním souřadném systému (BCS).

- TOWTCS
  Hodnoty opotřebení v souřadném systému nástroje na držáku nástroje (vztažný bod na držáku nástroje T).
- TOWKCS
  Hodnoty opotřebení v souřadném systému hlavy nástroje při kinetické transformaci.

Poznámka

Vyhodnocování jednotlivých složek opotřebení (přiřazení geometrickým osám, vyhodnocení znaménka) je ovlivňováno následujícími faktory:

- Aktivní rovina
- Transformace adaptéru
- Nastavované parametry:
  - SD42910 $SC_MIRROR_TOOL_WEAR
  - SD42920 $SC_WEAR_SIGN_CUTPOS
  - SD42930 $SC_WEAR_SIGN
  - SD42940 $SC_TOOL_LENGTH_CONST
  - SD42950 $SC_TOOL_LENGTH_TYPE

4.13.3.4 Délka nástroje a změna roviny

Když je nastavovanému parametru SD42940 $SC_TOOL_LENGTH_CONST přiřazena nenulová hodnota, můžete jednotlivé složky délky nástroje, jako jsou délka, opotřebení a základní rozměr, přiřadit při změně roviny geometrickým osám pro soustružnické a brusné nástroje.

SD42940 $SC_TOOL_LENGTH_CONST

Nastavovaný parametr se **nerovná** nule:

Přiřazení složek délky nástroje (délka, opotřebení a zkladní rozměr) geometrickým osám se při změně roviny obrábění (`G17` - `G19`) nemění.

Následující tabulka ukazuje přiřazení složek délky nástroje geometrickým osám pro soustružnické a brusné nástroje (typ nástroje 400 – 599):

Obsah	Délka 1	Délka 2	Délka 3
17	Y	X	Z
*)	X	Z	Y
19	Z	Y	X
-17	X	Y	Z
-18	Z	X	Y
-19	Y	Z	X

*) Každá hodnota, která se nerovná nule a ani některé z těchto šesti uváděných hodnot, bude vyhodnocena jako hodnota 18.

Následující tabulka ukazuje přiřazení složek délky nástroje geometrickým osám pro všechny ostatní nástroje (typ nástroje < 400, příp. > 599):

Rovina obrábění	Délka 1	Délka 2	Délka 3
*)	Z	Y	X

18	Y	X	Z
19	X	Z	Y
-17	Z	X	Y
-18	Y	Z	X
-19	X	Y	Z

*) Každá hodnota, která se nerovná nule a ani některé z těchto šesti uváděných hodnot, bude vyhodnocena jako hodnota 17.

Poznámka

Při zobrazování v tabulkách se vychází z toho, že geometrické osy až 3 jsou označeny X, Y, Z. Pro přiřazení korekce ose je určující ne identifikátor osy, ale posloupnost os.

4.13.4 On-line korekce nástroje

4.13.4.1 Definice polynomické funkce (FCTDEF)

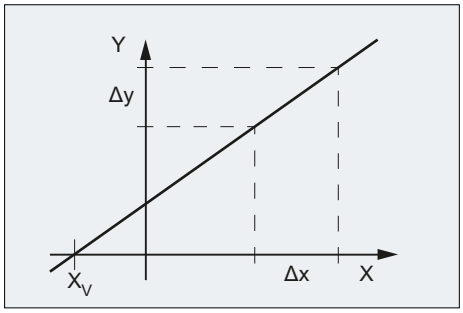
Určité strategie orovnávání (např. orovnávací kotouč) se vyznačují tím, že brusný kotouč je spojitě (lineárně) s určitou hodnotou přísuvu orovnávacího kotouče obroben na požadovaný rádius. Za tím účelem potřebujete lineární funkci mezi hodnotou přísuvu orovnávacího kotouče a zápisem hodnoty opotřebení příslušné délky. Definice této lineární funkce se provádí pomocí předem definované procedury FCTDEF(...) pro polynomické funkce maximálně třetího stupně.

Rovnice přímky

$y = f(x) = a_0 + a_1 * x_1$

a_1: Směrnice přímky, kde $a_1 = \Delta x / \Delta y$

a_0: Posunutí přímky podél osy X, kde $a_0 = -a1 * X_V$



Syntaxe

```
FCTDEF(<Func>,<LLimit>,<ULimit>,<a0>,<a1>,<a2>,<a3>)
```

Význam

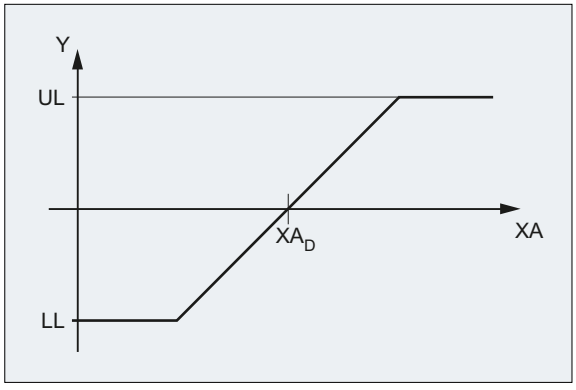
`FCTDEF(...)`:	Definice polynomické funkce pro PUTFTOCF(...): $y = f(x) = a_0 + a_1*x + a_2*x^2 + a_3*x^3$	
`<Func>`:	Číslo funkce	
	Datový typ:	INT
	Rozsah hodnot:	1, 2, 3
`<LLimit>`:	Spodní mezní hodnota	
	Datový typ:	REAL
`<ULimit>`:	Horní mezní hodnota	
	Datový typ:	REAL
`<a0>,<a1>,<a2>,<a3>`:	Koeficienty polynomické funkce	
	Datový typ:	REAL

Příklad

Definice

- Číslo funkce: 1
- Horní a dolní mezní hodnota: -100, 100
- Stoupání charakteristiky: $a_1 = 1$
- Pracovní bod by se měl nacházet uprostřed charakteristiky. Za tím účelem musí být charakteristika posunuta podle požadované polohy osy XA ve WCS v okamžiku definice funkce v NC programu v záporném směru osy Y: $a_0 = -a_1 * XA_D = -1 * \$AA_IW$
- $a_2 = a_3 = 0$

Charakteristika



UL Horní mezní hodnota

LL Spodní mezní hodnota

XA_D Požadovaná hodnota osy XA v okamžiku definice funkce v NC programu

Programování

Programový kód	Komentář
FCTDEF(1,-100,100,-$AA_IW[XA],1)	; definice funkce

4.13.4.2 Zápis on-line korekce nástroje, kontinuálně (PUTFTOCF)

Prostřednictvím předem definované procedury PUTFTOCF(...) se uskutečňuje on-line korekce nástroje podle polynomické funkce definované předtím pomocí příkazu FCTDEF(...) (Strana 753).

Poznámka

On-line korekce nástroje se může uskutečňovat také pomocí synchronní akce.

Pokud budete potřebovat další informace, viz Příručka k funkcím, Synchronní akce.

Syntaxe

```
PUTFTOCF(<Func>,<RefVal>,<ToolPar>,<Chan>,<Sp>)
```

Význam

PUTFTOCF(...):	Zápis on-line korekce nástroje, kontinuálně v bloku pomocí polynomické funkce definované pomocí příkazu FCTDEF(...)	
<Func>:	Číslo funkce, stanovené při definici funkce pomocí příkazu FCTDEF(...)	
	Datový typ:	INT
	Rozsah hodnot:	1, 2, 3
<RefVal>:	Vztažná hodnota, od které má být korekce odvozena (např. požadovaná hodnota osy).	
	Datový typ:	VAR REAL
<ToolPar>:	Číslo parametru opotřebení (Délka 1, 2 nebo 3), do kterého se má hodnota korekce započítat	
	Datový typ:	INT
<Chan>:	Číslo kanálu, v němž má být on-line korekce nástroje v platnosti **Poznámka**: Je zapotřebí jen tehdy, pokud se hodnota korekce nemá uplatňovat v aktivním kanálu.	
	Datový typ:	INT
<Sp>:	Číslo vřetena, pro které má být on-line korekce nástroje v platnosti **Poznámka:** Je zapotřebí jen tehdy, pokud má být korekce uplatňována na neaktivní brusný kotouč a nikoli na nástroj, který je momentálně používán.	
	Datový typ:	INT

4.13.4.3 Diskrétní zápis on-line korekce nástroje (PUTFTOC)

Funkce

Prostřednictvím předem definované procedury PUTFTOC(...) se uskutečňuje on-line korekce nástroje s pevně danou hodnotou korekce.

Syntaxe

```
PUTFTOC(<CorrVal>,<ToolPar>,<Chan>,<Sp>)
```

Význam

`PUTFTOC(...)`:	Zápis do on-line korekčních parametrů nástroje	
`<CorrVal>`:	Hodnota korekce, která se přičítá k parametru opotřebení	
	Datový typ:	REAL
`<ToolPar>`:	Číslo parametru opotřebení (Délka 1, 2 nebo 3), do kterého se má hodnota korekce započítat	
	Datový typ:	INT
`<Chan>`:	Číslo kanálu, v němž má být on-line korekce nástroje v platnosti **Poznámka:** Je zapotřebí jen tehdy, pokud se hodnota korekce nemá uplatňovat v aktivním kanálu.	
	Datový typ:	INT
`<Sp>`:	Číslo vřetena, pro které má být on-line korekce nástroje v platnosti **Poznámka:** Je zapotřebí jen tehdy, pokud má být korekce uplatňována na neaktivní brusný kotouč a nikoli na nástroj, který je momentálně používán.	
	Datový typ:	INT

4.13.4.4 Aktivování/deaktivování on-line korekce nástroje (FTOCON/FTOCOF)

On-line korekce nástroje se aktivuje, příp. deaktivuje pomocí příkazů G-funkcí FTOCON a FTOCOF.

Syntaxe

```
FTOCON
...
FTOCOF
```

Význam

FTOCON:	Aktivování on-line korekce nástroje Příkaz musí být naprogramován v kanálu, v němž má být on-line korekce nástroje aktivována.
FTOCOF:	Deaktivování on-line korekce nástroje Příkaz musí být naprogramován v kanálu, v němž má být on-line korekce nástroje deaktivována. **Poznámka:** Pomocí příkazu FTOCOF se další najíždění na korekci nástroje neuskutečňuje. V korekčních parametrech specifického břitu zůstává ale hodnota vypočítaná pomocí funkcí PUTFTOC/PUTFTOCF zachována. Aby byla on-line korekce nástroje s konečnou platností deaktivována, musí být po příkazu FTOCOF provedeno ještě aktivování/deaktivování nástroje (T...).

4.13.5 3D korekce rádiusu nástroje

4.13.5.1 Aktivování 3D korekce rádiusu nástroje pro 3D obvodové frézování (CUT3DC, CUT3DCD, ISD)

3D korekce rádiusu nástroje (3D-WRK) pro 3D obvodové frézování bez vyhodnocování omezujících ploch se aktivuje pomocí příkazů G-funkce CUT3DC, příp. CUT3DCD, s modální platností.

Pro vlastní aktivování se používají příkazy G41, příp. G42. Příkazem G40 se korekce rádiusu nástroje deaktivuje.

Syntaxe

```
G41/G42 ORIC/ORID ISD=... CUT3DC/CUT3DCD CDOF2 X... Y... Z...
...
G40 X... Y... Z...
```

Význam

CUT3DC:	3D-korekce rádiusu nástroje pro obvodové frézování (jen když je aktivní 5-osá transformace)	
CUT3DCD:	3D-korekce rádiusu nástroje pro obvodové frézování vztažená na diferenční nástroj (jen když je aktivní 5-osá transformace) Rozdíl rádiusu se určuje prostřednictvím parametru nástroje $TC_DP15.	
G41/G42 X... Y... Z...:	Aktivování korekce rádiusu nástroje	
	G41:	Korekce rádiusu nástroje vlevo od kontury
	G42:	Korekce rádiusu nástroje vpravo od kontury
	Poznámka: Aktivování se musí nacházet v bloku lineárního pohybu (G0/G1).	

CDOF2:	Deaktivování monitorování kolize pro 3D obvodové frézování	
ORIC/ORID:	Prostřednictvím příkazů G-funkcí ORIC a ORID se nastavuje chování při změnách orientace na vnějších rozích.	
	ORIC:	Změny orientace na vnějších rozích jsou superponovány s vkládaným kruhovým blokem.
	ORID:	Změny orientace na vnějších rozích jsou prováděny před vkládaným kruhovým blokem.
ISD=<hodnota>:	Prostřednictvím adresy ISD je možno při obvodovém frézování a při **aktivní** 3D korekci rádiusu nástroje měnit hloubku zajíždění nástroje do materiálu.	
	<hodnota>:	Údaj hloubky zajíždění nástroje
G40 X... Y... Z...:	Deaktivování korekce rádiusu nástroje **Poznámka:** Deaktivování se musí nacházet v lineárním bloku (G0/G1) s pohybem geometrických os.	

Poznámka

Příkazy G-funkcí pro aktivování 3D korekce rádiusu nástroje jsou vyhodnocovány v najížděcím bloku, tzn. typicky v bloku, který obsahuje příkaz G41 nebo G42.

Příkaz G41, příp. G42, může být naprogramován také v blocích, ve kterých geometrické osy, jichž se korekce týká, neuskutečňují žádné posuvy. V tomto případě je najížděcím blokem první blok s posuvy, který za tímto blokem následuje.

Změna varianty 3D korekce rádiusu nástroje, když je korekce rádiusu nástroje aktivní, je ignorována, přičemž se neaktivuje žádný alarm.

Příklad:

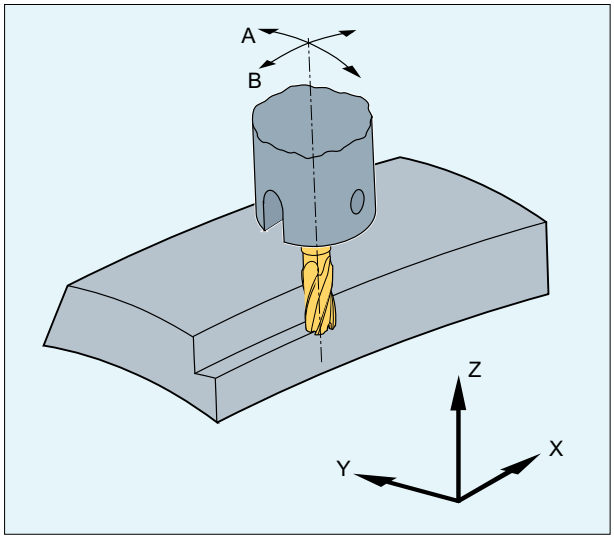
Programový kód	**Komentář**
	; Definice nástroje D1:
$TC_DP1[1,1]=120	; Typ (stopková fréza)
$TC_DP3[1,1]=20	; Vektor pro korekci délky
$TC_DP6[1,1]=8	; rádius
N10 X0 Y0 Z0 T1 D1 F12000	; Aktivování nástroje.
N20 TRAORI(1)	; Aktivování transformace.
N30 G42 ORIC ISD=10 **CUT3DC** G64 X30	; aktivování 3D obvodového frézování
	; Spojité změny orientace na vnějších rozích
	; Hloubka zajíždění nástroje: 10 mm
N40 ORIWKS A30 B15	; Změna orientace na rohu prostřednictvím zadání polohy osy.
N50 Y20 A3=1 C3=1	; Blok posuvu se změnou orientace.
	; Zadání orientace pomocí směrového vektoru.
N60 X50 Y30	; Blok posuvu s konstantní orientací.
N70 Y50 A3=0.5 B3=1 C3=5	; Blok posuvu se změnou orientace.
N80 M63	; Blok bez informací o posuvu.

Programový kód	Komentář
N90 X0 ISD=20	; blok posuvu se změnou hloubky zajíždění nástroje.
N100 G40 Y0	; Deaktivování korekce rádiusu nástroje.
N110 M30	

Další informace

Dráha a orientace

Zde využívaná varianta obvodového frézování je realizována zadáním dráhy (řídící křivky) a k ní odpovídající orientace. U tohoto druhu obrábění nemá tvar nástroje na dráze žádný význam. Rozhodující je pouze rádius v bodě záběru nástroje.



Obrázek 4-9 Obvodové frézování

Chování při najíždění

Pro chování při najíždění je u 3D variant korekce rádiusu nástroje vždy zvoleno NORM.

Chování na vnějších rozích

Na vnějších rozích se při obvodovém frézování s 3D korekcí rádiusu nástroje analogicky s 2½D korekcí rádiusu nástroje vyhodnocuje příkaz G-funkce ze skupiny 18 (chování korekce nástroje v rozích):

- G450: Přechodový prvek kruh (nástroj objíždí rohy obrobku po kruhové dráze)
  Na rozdíl od řešení při 2½D korekci rádiusu nástroje je vkládaným konturovým prvkem na vnějším rohu vždy kruh s rádiusem 0, na který se uplatňuje korekce rádiusu nástroje úplně stejně, jako na kteroukoli jinou naprogramovanou dráhu. Na místě tohoto kruhu není možné vložit kuželovité průchody nástroje. Adresa DISC proto v tomto případě nemá žádný význam a nevyhodnocuje se.
- G451: Průsečík ekvidistantních drah (nástroj řeže ostré rohy obrobku)
  Průsečík se určuje tak, že se offsetové křivky z obou na pohybu se podílejících bodů prodlouží a určí se jejich průsečík v rovině kolmé na orientaci nástroje v rohu.

Najíždění na průsečík (G451) se nepoužívá, jestliže je mezi podílejícími se bloky posuvů vložen minimálně jeden blok, který obsahuje změnu orientace nástroje. V takovém případě se vždy v rohu vkládá kruh.

Chování při změnách orientace na vnějších rozích

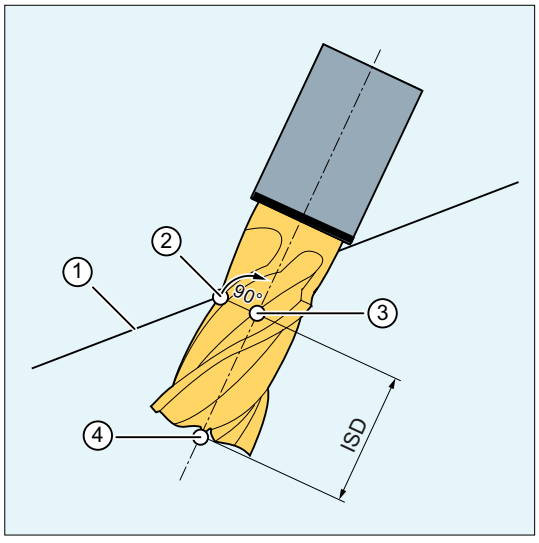
Pomocí příkazů G-funkcí ORIC a ORID může být definováno, zda se změny orientace, které byly naprogramovány mezi oběma bloky posuvů vytvářejících roh, uskuteční před zahájením zpracování vloženého bloku kruhového oblouku (ORID) nebo současně s ním (ORIC).

Hloubka zajíždění nástroje

Hloubka zajíždění frézy se rovná vzdálenosti mezi pomocným bodem frézy a špičkou nástroje.

Pomocný bod frézy je kolmá projekce obráběcího bodu frézy na naprogramovanou dráhu na podélné ose nástroje.

Prostřednictvím hloubky zajíždění nástroje je možné nastavovat polohu obráběcího bodu na ploše pláště nástroje.



① Naprogramovaná dráha
② Obráběcí bod frézy
③ Pomocný bod frézy
④ Špička frézy
ISD Hloubka zajíždění (InSertion Depth)

Obrázek 4-10 Hloubka zajíždění nástroje

Korekce nástroje vztažená na diferenční nástroj

3D korekce rádiusu nástroje pro obvodové frézování vztažená na diferenční nástroj se aktivuje prostřednictvím příkazu CUT3DCD. Je potřeba ji použít tehdy, jestliže je naprogramovaná kontura vztažena na dráhu středu normovaného nástroje a pokud se pro opracování používá jeden z nástrojů, které se od něj odlišují. Při výpočtu 3D korekce rádiusu nástroje se potom započítává pouze hodnota opotřebení rádiusu aktivního nástroje ($TC_DP15) a v případě potřeby naprogramovaný korekční offset nástroje OFFN a TOFFR/TOFFLR. Základní rádius ($TC_DP6) aktivního nástroje se **nezapočítává**.

Frézování kapes se šikmými bočními stěnami pro obvodové frézování s příkazem CUT3DC

Při této 3D korekci rádiusu nástroje se uskutečňuje kompenzace odchylky rádiusu frézy tím, že se provádí přísuv ve směru normálového vektoru obráběné plochy. Jestliže je ponechána hloubka zajíždění nástroje ISD na stejné hodnotě, zůstává rovina, ve které leží čelní plocha frézy, přitom nezměněna. Fréza s např. menším rádiusem, než jaký má normovaný nástroj, by potom nedosáhla na dno kapsy, které představuje také hraniční plochu. Aby se uskutečňoval automatický přísuv nástroje, musí být řídícímu systému tyto hraniční plochy známy, viz kapitola „3D obvodové frézování se zohledněním hraniční plochy (CUT3DCC, CUT3DCCD) (Strana 767)".

Advanced Surface / Top Surface

Poznámka

Při použití korekce rádiusu nástroje CUT3DCD ve spojení s funkcí „Advanced Surface" nebo „Top Surface", pro které je nezbytná licence, je zapotřebí dodržet doporučení pro nastavení parametrů, která se týkají funkcí „Advanced Surface" / „Top Surface"!

Pro kontrolu nastavených dat je k dispozici speciální zkušební program, který si lze stáhnout přes portál SIOS.

→ Zkušební program pro funkce Advanced Surface / Top Surface (https://support.industry.siemens.com/cs/ww/en/view/109738423)

4.13.5.2 Aktivování 3D korekce rádiusu nástroje pro 3D frézování na čelní ploše (CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD)

3D korekce rádiusu nástroje (3D-WRK) pro 3D frézování na čelní ploše se aktivuje pomocí příkazů G-funkcí CUT3DF, CUT3DFS, CUT3DFF, příp. CUT3DFD, s modální platností.

Pro vlastní aktivování se používají příkazy G41, příp. G42.

Aby bylo možné korekci rádiusu nástroje vypočítat, je při 3D frézování na čelní ploše zapotřebí definice normálového vektoru plochy pro rovinu, která má být opracována. To musí být uskutečněno v bloku s příkazem G41, příp. G42, pomocí adres A4, B4, C4 a A5, B5, C5.

Příkazem G40 se korekce rádiusu nástroje deaktivuje.

Syntaxe

```
G41/G42 ORIC/ORID CUT3DF/CUT3DFS/CUT3DFF/CUT3DFD X... Y... Z... A4=... B4=...
C4=... A5=... B5=... C5=...
...
G40 X... Y... Z...
```

Význam

CUT3DFS:	3D korekce rádiusu nástroje pro frézování na čelní ploše s konstantní orientací. Orientace nástroje je definována příkazy G17 - G19 a nemůže být ovlivňována pomocí framů.	
CUT3DFF:	3D korekce rádiusu nástroje pro frézování na čelní ploše s konstantní orientací. Orientace nástroje je definována příkazy G17 - G19 a v případě potřeby může být pomocí framu po-otočena do jiného směru.	
CUT3DF:	3D-korekce rádiusu nástroje pro frézování na čelní straně se změnami orientace (jen když je aktivní 5-osá transformace).	
CUT3DFD:	3D-korekce rádiusu nástroje pro frézování na čelní ploše vztažená na diferenční nástroj se změnami orientace (jen když je aktivní 5-osá transformace). Rozdíl rádiusu se určuje prostřednictvím parametru nástroje $TC_DP15. **Poznámka**: Příkaz CUT3DFD je možný pouze ve spojení s funkcí „Vyhlazení normálového vektoru plochy při 3D frézování na čelní ploše". Ta se aktivuje prostřednictvím volání funkce „Top Surface", pro kterou je nezbytná licence, pomocí cyklu CYCLE832(...).	
G41/G42 X... Y... Z...:	Aktivování korekce nástroje Chování v případě příkazů G41 a G42 je při 3D frézování na čelní ploše identické. **Poznámka:** Aktivování se musí nacházet v bloku lineárního pohybu (G0/G1).	
A4/5=... B4/5=... C4/5=...:	Definice normálového vektoru plochy pro opracovávanou rovinu	
	A4=... B4=... C4=...:	Definice makra na začátku bloku
	A5=... B5=... C5=...:	Definice makra na konci bloku
ORIC/ORID:	Prostřednictvím příkazů G-funkcí ORIC a ORID se nastavuje chování při změnách orientace na vnějších rozích.	
	ORIC:	Změny orientace na vnějších rozích jsou superponovány s vkládaným kruhovým blokem.
	ORID:	Změna orientace se bude provádět před kruhovým blokem.
G40 X... Y... Z...:	Deaktivování korekce rádiusu nástroje **Poznámka:** Deaktivování se musí nacházet v lineárním bloku (G0/G1) s pohybem geometrických os.	

Poznámka

Příkaz G41, příp. G42, může být naprogramován také v blocích, ve kterých geometrické osy, jichž se korekce týká, neuskutečňují žádné posuvy. V tomto případě je najížděcím blokem první blok s posuvy, který za tímto blokem následuje.

Změna varianty 3D korekce rádiusu nástroje, když je korekce rádiusu nástroje aktivní, je ignorována, přičemž se neaktivuje žádný alarm.

Příklady

Příklad 1: 3D frézování na čelní ploše s funkcí CUT3DF

Programový kód	Komentář
N10	; Definice nástroje D1:
N20 $TC_DP1[1,1]=121	; Typ nástroje (Torus fréza)
N30 $TC_DP3[1,1]=20	;-Korekce délky
N40 $TC_DP6[1,1]=5	; rádius
N50 $TC_DP7[1,1]=3	; Rádius zaoblení
N60	
N70	
N80 X0 Y0 Z0 A0 B0 C0 G17 T1 D1 F12000	; Aktivování nástroje.
N90 TRAORI(1)	; Aktivování transformace orientace
N100 B4=-1 C4=1	; Definice roviny.
N110 **G41** ORID **CUT3DF** G64 X10 Y0 Z0	; Aktivování korekce nástroje.
N120 X30	
N130 Y20 A4=1 C4=1	; Vnější rohy, nová definice roviny.
N140 B3=1 C3=5	; Změna orientace s funkcí ORID.
N150 B3=1 C3=1	; Změna orientace s funkcí ORID.
N160 X-10 A5=1 C5=2 ORIC	
N170 A3=-2 C3=1	; Změna orientace s funkcí ORIC.
N180 A3=-1 C3=1	; Změna orientace s funkcí ORIC.
N190 Y-10 A4=-1 C4=3	; Nová definice roviny.
N200 X-20 Y-20 Z10	; Vnitřní roh s předcházejícím blokem.
N210 X-30 Y10 A4=1 C4=1	; Vnitřní roh, nová definice roviny.
N220 A3=1 B3=0.5 C3=1.7	; Změna orientace s funkcí ORIC.
N230 X-20 Y30 A4=1 B4=-2 C4=3 ORID	
N240 A3 = 0.5 B3=-0.5 C3=1	; Změna orientace.
N250 X0 Y30 C4=1	; Pohyb po dráze, nová rovina.
	; Orientace s programováním relativních hodnot.
N260 BSPLINE X20 Z15	; Začátek splinu, relativní programování orientace
N270 X30 Y25 Z18	; zůstává během splinu aktivní.
N280 X40 Y20 Z13	
N290 X45 Y0 PW=2 Z8	
N300 Y-20	
N310 G2 ORIMKS A30 B45 I-20 X25 Y-40 Z0	; Spirála, orientace s programováním os.
N320 G1 X0 A3=-0.123 B3=0.456 C3=2.789 B4=-1 C4=5 B5=-1 C5=2	; Pohyb po dráze, orientace, rovina není konstantní.
N330 X-20 G40	; Deaktivování korekce rádiusu nástroje.
N340 M30	

Příklad 2: NC program vytvořený v systému CAD (pouze část) s funkcí CUT3DFD

Programový kód	Komentář
N01 G710	
N03 T="12"	

Programový kód	**Komentář**
N06 S5305 M03	
N07 G642	
; Najíždění na počáteční pozici v systému MCS při zohledňování délky nástroje.	
G00 G90 X-250.62787 Y-38.37944 A=DC(253.12719) B-12.49543	
G00 G90 Z251.80052	
; Konec nastavování polohy v MCS.	
;	
TRAORI(1)	; Aktivování transformace orientace
G500	
D1	
CYCLE832(0.01, _TOP_SURFACE_SMOOTH_ON + _ORI_FINISH, 1)	; Volání cyklu CYCLE832 s parametry: ; Tolerance kontury= 0,01 mm, ; Způsob opracování: Top Surface s vyhlazením, ; Obrábění načisto se zadáním tolerance orientace, ; Tolerance orientace = 1 stupeň
CUT3DFD	
N08 G90 G94	
N09 G00 X-269.21195 Y128.32027 Z1.18577 A3=-.216361688 B3=.934284397 C3=-.283373051	; Bloky N09 až N10 jsou rychlou částí najížděcího pohybu na obrobek s konstantní orientací.
N10 G00 X-251.90301 Y53.57752 Z23.85561	
N11 G01 X-247.57578 Y34.89183 Z29.52308 F50000.00000	; V blocích N11 až N21 je realizována pomalá část najížděcího pohybu. Nástroj se již nachází v blízkosti obrobku a kromě toho jsou z cyklu CYCLE832 (díky aktivnímu příkazu G01) již nyní aktivní nastavení pro výrobu forem (např. COMPSURF). Úsek dráhy této tak zvané natáčecí fáze pro správné chování při výrobě forem by měl odpovídat přibližně 1000-násobku tolerance kontury (v tomto případě tedy 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	
N21 X-248.05086 Y30.48986 Z10.01931 A5=-.060687572 B5=.974940255 C5=-.214029243 A3=-.215712821 B3=.932005189 C3=-.291263295	

Programový kód	**Komentář**
N22 **G41** X-248.06237 Y30.32400 Z9.36695 A5=-.060431854 B5=.973045457 C5=-.222554556 A3=-.214974689 B3=.929398552 C3=-.300007025 F1061.03295	; Od bloku N22 je normálový vektor plochy poprvé definován v rámci celého bloku (tzn. normálový vektor plochy na konci předcházejícího bloku N21 je již k dispozici, stejně jako normálový vektor plochy na začátku bloku N22 a normálový vektor plochy na konci boku N22). Tím je předpoklad pro aktivování korekce nástroje pomocí příkazů G41/ G42 splněn.
N23 X-248.07130 Y30.15119 Z8.71082 A5=-.060165696 B5=.971048883 C5=-.231179920 A3=-.214177198 B3=.926684940 C3=-.308841625	
N24 X-248.07829 Y29.97126 Z8.05094 A5=-.059884286 B5=.968941717 C5=-.239928784 A3=-.213318480 B3=.923853466 C3=-.317789237	
N25 X-248.08317 Y29.78487 Z7.38844 A5=-.059584206 B5=.966718449 C5=-.248807482 A3=-.212397895 B3=.920898045 C3=-.326854594	
N26 X-248.08578 Y29.59254 Z6.72679 A5=-.059263963 B5=.964380907 C5=-.257793037 A3=-.211418355 B3=.917822366 C3=-.336012474	
...	

Poznámka

Při 3D frézování na čelní ploše s funkcí CUT3DFD je pro aktivování korekce nástroje pomocí příkazů G41/G42 zapotřebí definice normálového vektoru plochy. Naprogramování příkazů G41/ G42 bez definice normálového vektoru plochy má za následek aktivování alarmu.

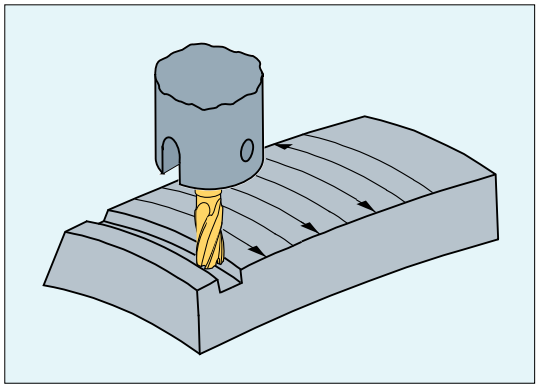
Viz také

CYCLE832 - Vysokorychlostní obrábění (Strana 1123)

Další informace

3D frézování na čelní ploše

Pro tento druh 3D frézování je zapotřebí popis řádek po řádku jednotlivých 3D drah na povrchu obrobku. Výpočty se obvykle uskutečňují v systému CAM a zohledňuje se při nich tvar nástroje a rozměry nástroje. Postprocesor zapisuje do výrobního programu - vedle NC bloků - orientace nástroje (když je aktivní 5-osá transformace) a příkazy G-funkcí pro požadovanou 3D korekci nástroje. V důsledku toho má obsluha stroje možnost - na rozdíl od případu, kdy se nástroj používá pro výpočet jeho drah v NC systému - používat nepatrně menší nástroje.



Obrázek 4-11 Čelní frézování

Chování při najíždění

Pro chování při najíždění je u 3D variant korekce rádiusu nástroje vždy zvoleno NORM.

Chování na vnějších rozích

S vnějšími rohy se při frézování na čelní ploše zachází jako s kruhovými oblouky s rádiusem 0, přičemž rovina tohoto kruhu je dána tečnou na konci prvního bloku a tečnou na začátku druhého bloku. Díky tomu je možná také změna orientace při přechodu na další blok. Na vnějších rozích se proto vždy vkládá kruh jako konturový prvek. Najíždění na průsečík není při frézování na čelní ploše možné.

Chování při změnách orientace na vnějších rozích

Pomocí příkazů G-funkcí ORIC a ORID může být definováno, zda se změny orientace, které byly naprogramovány mezi oběma bloky posuvů vytvářejících roh, uskuteční před zahájením zpracování vloženého bloku kruhového oblouku (ORID) nebo současně s ním (ORIC).

Korekce nástroje vztažená na diferenční nástroj

3D korekce rádiusu nástroje vztažená na diferenční nástroj se aktivuje prostřednictvím příkazu CUT3DFD. Je potřeba ji použít tehdy, jestliže je naprogramovaná kontura vztažena na dráhu středu normovaného nástroje a pokud se pro opracování používá jeden z nástrojů, které se od něj odlišují. Při výpočtu 3D korekce rádiusu nástroje se potom započítává pouze hodnota opotřebení rádiusu aktivního nástroje ($TC_DP15) a v případě potřeby naprogramovaný korekční offset nástroje OFFN a TOFFR/TOFFLR. Základní rádius ($TC_DP6) aktivního nástroje se **nezapočítává**.

3D frézování na čelní ploše s příkazem CUT3DFD je možné pouze ve spojení s funkcí „Vyhlazení normálového vektoru plochy při 3D frézování na čelní ploše“. Ta se aktivuje prostřednictvím volání funkce „Top Surface“, pro kterou je nezbytná licence, pomocí cyklu CYCLE832(...). Aktivování musí být uskutečněno **před** zapnutím korekce nástroje pomocí příkazů G41/G42, a sice nikoli přímo před zakousnutím nástroje, ale již o jeden úsek délky dráhy dříve, který odpovídá přibližně 1000-násobku tolerance kontury (např. 1000 x 0,01 mm = 10 mm). Deaktivování musí probíhat v opačném pořadí operací: Napřed deaktivujte korekci nástroje pomocí příkazu G40, potom, po jednom úseku dráhy, který odpovídá přibližně 1000-násobku tolerance kontury, funkci deaktivujte příkazem např. CUT2D (a podobným).

Aby bylo možno použít „Vyhlazení normálového vektoru plochy při 3D frézování na čelní ploše“, je zapotřebí navíc ještě odblokování funkce „Interpolace normálového vektoru plochy pomocí polynomů“:

MD28291 $MC_MM_SMOOTH_SURFACE_NORMALS = TRUE

Poznámka

Pro 3D frézování na čelní ploše s funkcí CUT3DFD ve spojení s funkcí „Top Surface" je zapotřebí dodržet doporučení pro nastavení parametrů, která se týkají funkce „Top Surface"!

Pro kontrolu nastavených dat je k dispozici speciální zkušební program, který si lze stáhnout přes portál SIOS.

→ Zkušební program pro funkci Top Surface (https://support.industry.siemens.com/cs/ww/en/view/109738423)

Viz také

CYCLE832 - Vysokorychlostní obrábění (Strana 1123)

4.13.5.3 3D obvodové frézování se zohledněním hraniční plochy (CUT3DCC, CUT3DCCD)

V případě 3D obvodového frézování se spojitou nebo konstantní změnou orientace nástroje je často programována dráha středu definovaného normovaného nástroje. Protože v praxi se nestává často, že by byl vyhovující normovaný nástroj k dispozici, může se použít nástroj, který se od normovaného nástroje přespříliš neliší (≤ 5%).

Pomocí příkazu CUT3DCCD je možné pro specifický reálný nástroj zohlednit omezující plochu, která by popisovala naprogramovaný normovaný nástroj. NC program popisuje dráhu středu normovaného nástroje.

Pomocí příkazu CUT3DCC je možné při použití válcových nástrojů zohlednit omezující plochu, na kterou by měl naprogramovaný normovaný nástroj dosáhnout. NC program popisuje konturu obráběné plochy.

Normálový vektor omezující plochy se zadává stejně jako při 3D frézování na čelní ploše pomocí příkazů A4, B4, C4 a A5, B5, C5.

Syntaxe

```
G41/G42 CUT3DCCD/CUT3DCC CDOF2 X... Y... Z... A4=... B4=... C4=... A5=... B5=...
C5=...
...
G40 X... Y... Z...
```

Význam

CUT3DCCD:	3D korekce rádiusu nástroje pro obvodové frézování se zohledňováním omezovací plochy s diferenčním nástrojem na dráze středu nástroje: Přísuv k omezující ploše.
CUT3DCC:	3D korekce rádiusu nástroje pro obvodové frézování se zohledňováním omezovací plochy s 3D korekcí rádiusu: Kontura na obráběné ploše

G41/G42 X... Y... Z...:	Aktivování korekce rádiusu nástroje	
	G41:	Korekce rádiusu nástroje vlevo od kontury
	G42:	Korekce rádiusu nástroje vpravo od kontury
	Poznámka: Aktivování se musí nacházet v bloku lineárního pohybu (G0/G1).	
CDOF2:	Deaktivování monitorování kolize pro 3D obvodové frézování	
A4/5=... B4/5=... C4/5=...:	Definice normálového vektoru plochy pro omezovací rovinu	
	A4=... B4=... C4=...:	Definice makra na začátku bloku
	A5=... B5=... C5=...:	Definice makra na konci bloku
G40 X... Y... Z...:	Deaktivování korekce rádiusu nástroje **Poznámka:** Deaktivování se musí nacházet v lineárním bloku (G0/G1) s pohybem geometrických os.	

Poznámka

Příkazy G-funkcí pro aktivování 3D korekce rádiusu nástroje jsou vyhodnocovány v najížděcím bloku, tzn. typicky v bloku, který obsahuje příkaz G41 nebo G42.

Příkaz G41, příp. G42, může být naprogramován také v blocích, ve kterých geometrické osy, jichž se korekce týká, neuskutečňují žádné posuvy. V tomto případě je najížděcím blokem první blok s posuvy, který za tímto blokem následuje.

Změna varianty 3D korekce rádiusu nástroje, když je korekce rádiusu nástroje aktivní, je ignorována, přičemž se neaktivuje žádný alarm.

Příklad:

Programový kód	**Komentář**
N10 $TC_DP1[1,1]=120	; Válcová fréza
N20 $TC_DP6[1,1]=10	
N30 $TC_DP15[1,1]=-3	
...	
; Opracování pomocí válcové frézy a funkce CUT3DCCD	
N110 TRAORI	; Aktivování transformace.
N120 A4=0 B4=0 C4=1	; Definice normálového vektoru plochy pro omezovací rovinu na začátku bloku
N130 X0 Y0 Z0 A0 C0 T1 D1 F20000	
N140 X10 Y0 Z0 G41 **CUT3DCCD** CDOF2 G64	; Aktivování 3D obvodového frézování při zohledňování omezovací plochy + deaktivování monitorování kolize.
N150 X20	
N160 X30 A45	; Tupý úhel ==> žádný přísuv
N170 X40 A-45	; Špičatý úhel ==> přísuv
N180 X55	
N190 Y10 Z10	; Pohyb ve směru nástroje.
N200 Y20	

Programový kód	Komentář
N210 C45	; Čistá změna orientace.
N220 Y30 C90	
N230 A5=-1 B5=0 C5=2 Y40	; Změna povrchu.
N240 Y50 G40	; Deaktivování korekce rádiusu nástroje.
...	

Další informace

Typ nástroje

Vyhodnocuje se typ nástroje (parametr nástroje $TC_DP1). Přípustné jsou pouze frézovací nástroje s válcovou stopkou (válcová nebo stopková fréza, Torus fréza a v mezním případě ještě i válcová zápustková fréza). To odpovídá typům nástrojů s čísly 1 - 399 s výjimkou čísel 111 a 155 až 157.

Normovaný nástroj se zaoblením rohů

Rohové zaoblení normovaného nástroje je popisováno pomocí parametru nástroje $TC_DP7. Z parametru nástroje $TC_DP16 vyplývá odchylka rohového zaoblení reálného nástroje oproti normovanému nástroji.

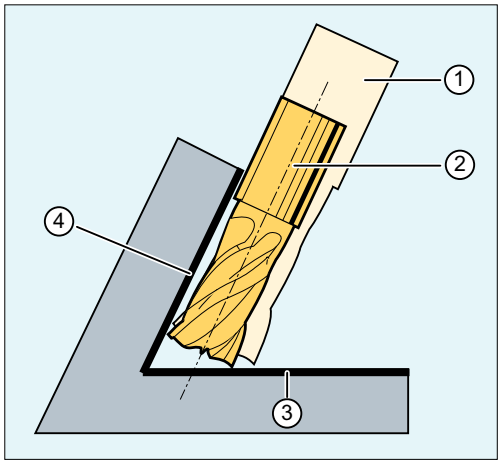
Příklad: Torus frézy se zmenšeným rádiusem oproti normovanému nástroji

Typ nástroje	Rádius stopky (R)	Rádius v rohu (r)
Normovaný nástroj se zaoblením rohů	R = $TC_DP6	r = $TC_DP7
Reálný nástroj se zaoblením rohů Typy nástrojů 121 a 131 Torus frézy (stopkové frézy se zaoblením rohů)	R' = $TC_DP6 + $TC_DP15 + OFFN	r' = $TC_DP7 + $TC_DP16

V tomto příkladu je záporná jak hodnota $TC_DP15 + OFFN, tak také hodnota $TC_DP16.

3D korekce rádiusu nástroje s funkcí CUT3DCCD: Dráha středu nástroje s přísuvem až po omezující plochu

Jestliže je použit nástroj, který má ve srovnání s vyhovujícím normovaným nástrojem menší rádius, potom se v podélném směru přisouvaná fréza pohybuje ještě dál o takovou hodnotu, dokud znovu nedojde k dotyku se dnem kapsy. Tímto způsobem jsou růžky, které jsou tvořeny mezi obráběnou a omezující plochou, obrobeny až do takové míry, kterou daný nástroj připouští. Přitom se jedná o smíšený způsob obránění skládající se z obvodového a čelního frézování. Analogicky k nástroji s menším rádiusem se v případě použití nástroje se zvětšeným rádiusem přísuv zmenšuje o odpovídající hodnotu v opačném směru.



① Normovaný nástroj
② Nástroj s menším rádiusem je přisouván až na omezující plochu
③ Omezující plocha
④ Obráběná plocha

Oproti všem ostatním korekcím nástroje ze skupiny G-funkcí č. 22 nemá pro příkaz CUT3DCCD zadávaný parametr nástroje $TC_DP6 žádný význam pro rádius nástroje a výslednou konturu proto neovlivňuje. Korekční offset vyplývá ze součtu hodnot opotřebení rádiusu nástroje (parametr nástroje $TC_DP15) a offsetu nástroje OFFN naprogramovaného pro výpočet kolmého offsetu k omezující ploše.

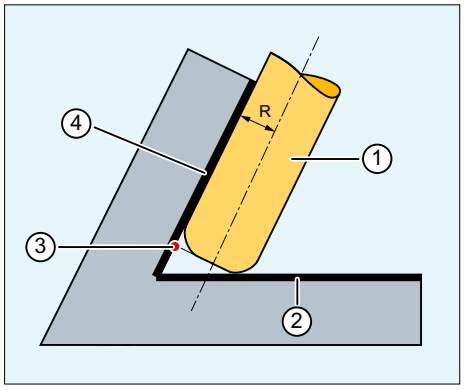
To, zda obráběná plocha leží vlevo nebo vpravo od dráhy, nemůže být ze sestaveného výrobního programu zjištěno. Vychází se proto z kladné hodnoty rádiusu a záporné hodnoty opotřebení původního nástroje. Záporná hodnota opotřebení vždy popisuje nástroj se zmenšeným průměrem.

Použití válcových nástrojů

Při použití válcových nástrojů je přísuv potřebný jedině tehdy, pokud obráběná plocha a omezující plocha mezi sebou svírají ostrý úhel (menší než 90 stupňů). Pokud se používají Torus frézy (stopková fréza s rohovým zaoblením), potom tato situace vyžaduje přísuv v podélném směru nástroje tak u ostrých, tak i u tupých úhlů.

3D korekce rádiusu nástroje s funkcí CUT3DCC: Kontura na obráběné ploše

Pokud je aktivní příkaz CUT3DCC a používá se Torus fréza, je naprogramovaná dráha vztažena na fiktivní válcovou frézu stejného průměru. Z toho vyplývající vztažný bod dráhy je v případě použití Torus frézy uveden na následujícím obrázku.



① Torus fréza
② Omezující plocha
③ Vztažný bod dráhy
④ Obráběná plocha
R Rádius stopky (rádius nástroje)

Je přípustné, aby se úhel mezi obráběnou a omezující plochou i v rámci jednoho bloku měnil z ostrého na tupý a obráceně.

Oproti normovanému nástroji smí být použitý reálný nástroj jak větší, tak také menší. Výsledný rohový rádius přitom nesmí být záporný a znaménko výsledného rádiusu nástroje musí zůstat zachováno.

U příkazu CUT3DCC se NC program vztahuje na konturu v rovině obrábění. Přitom se používá stejně jako v případě obvyklé korekce rádiusu nástroje celkový rádius, který se skládá z následujících složek:

- Rádius nástroje (parametr nástroje $TC_DP6)
- Hodnota opotřebení (parametr nástroje $TC_DP15)
- Offset nástroje OFFN naprogramovaný pro výpočet kolmého offsetu k omezující ploše

Poloha omezující plochy se určuje z následujícího rozdílu:

Rozměry normovaného nástroje - rádius nástroje (parametr nástroje $TC_DP6).

Advanced Surface / Top Surface

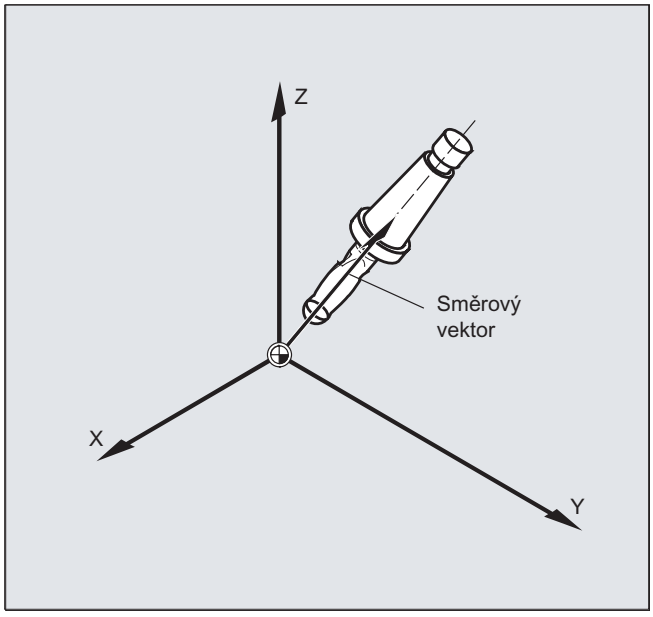
Poznámka

Při použití korekce rádiusu nástroje CUT3DCC / CUT3DCCD ve spojení s funkcí „Advanced Surface“ nebo „Top Surface“, pro které je nezbytná licence, je zapotřebí dodržet doporučení pro nastavení parametrů, která se týkají funkcí „Advanced Surface“ / „Top Surface“! Pro kontrolu nastavených dat je k dispozici speciální zkušební program, který si lze stáhnout přes portál SIOS.

→ Zkušební program pro funkce Advanced Surface / Top Surface (https://support.industry.siemens.com/cs/ww/en/view/109738423)

4.13.6 Orientace nástroje (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST)

Pod pojmem orientace nástroje se rozumí geometrické nasměrování nástroje v prostoru. V případě obráběcího stoje s 5 osami může být orientace nástroje nastavována pomocí programových příkazů.



Vyhlazovací pohyby orientace aktivované příkazy `OSD` a `OST` jsou v závislosti na druhu interpolace orientace nástroje realizovány různými způsoby.

Když je aktivní vektorová interpolace, interpoluje se vyhlazený průběh orientace také pomocí vektorové interpolace. Oproti tomu se při aktivní interpolaci kruhových os orientace vyhlazuje přímo pomocí pohybů kruhových os.

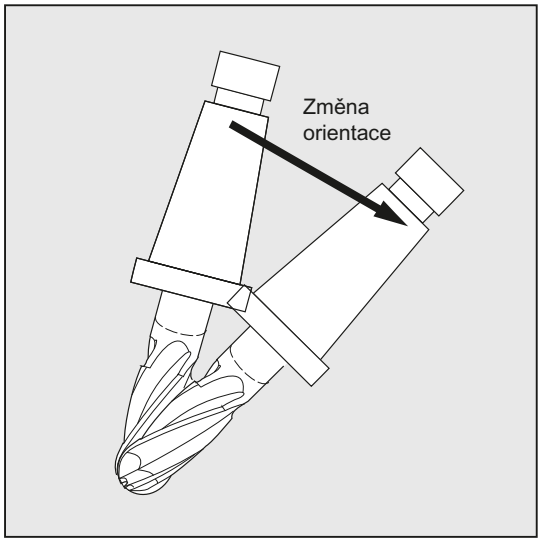
Programování

Programování změny orientace:

Změna orientace nástroje může být naprogramována pomocí těchto prostředků:

- Přímé programování kruhových os `A`, `B`, `C` (interpolace kruhových os)
- Eulerův úhel nebo úhel RPY.
- Směrový vektor (vektorová interpolace pomocí zadání příkazů `A3` nebo `B3` nebo `C3`)
- `LEAD/TILT` (frézování na čelní ploše)

Vztažným souřadným systémem je buď souřadný systém stroje (`ORIMKS`) nebo aktuální souřadný systém obrobku (`ORIWKS`).



Programování orientace nástroje:

ORIC:	Orientace a dráhový pohyb probíhají souběžně
ORID:	Orientace a dráhový pohyb probíhají jeden po druhém
OSOF:	Žádné vyhlazování orientace
OSC:	Konstantní orientace
OSS:	Vyhlazení orientace jen na začátku bloku
OSSE:	Vyhlazení orientace na počátku a konci bloku
ORIS:	Rychlost změn orientace při aktivovaném vyhlazování orientace ve stupních na mm (platí pro funkce OSS a OSSE)
OSD:	Vyhlazování orientace zadáním délky zaoblení pomocí nastavovaného parametru: SD42674 $SC_ORI_SMOOTH_DIST
OST:	Vyhlazování orientace v případě vektorové interpolace zadáním úhlové tolerance ve stupních pomocí nastavovaného parametru: SD42676 $SC_ORI_SMOOTH_TOL V případě interpolace kruhových os se zadaná tolerance používá jako maximální odchylka orientačních os.

Poznámka

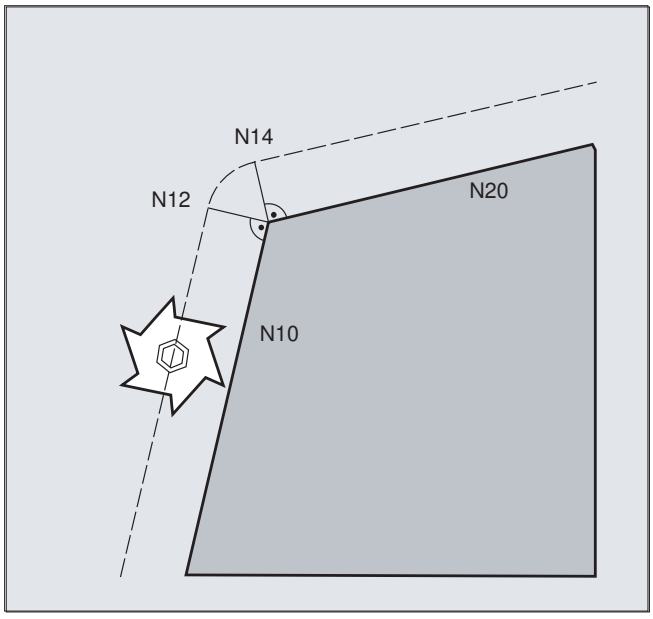
Všechny příkazy pro vyhlazování orientace nástroje (OSOF, OSC, OSS, OSSE, OSD a OST) jsou soustředěny ve skupině G-funkcí č. 34. Mají modální platnost, tzn. může být v platnosti vždy jen jeden z těchto příkazů.

Příklady

Příklad 1: ORIC

Pokud jsou mezi bloky posuvů N10 a N20 naprogramovány dva nebo více bloků se změnami orientace (např. A2=... B2=... C2=...) a pokud je aktivní funkce ORIC, potom se vkládaný

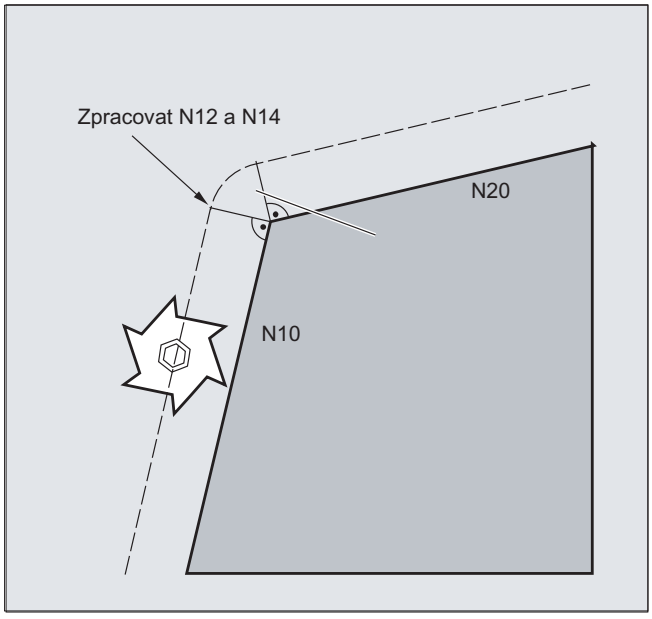
blok kruhového oblouku rozdělí v závislosti na počtu úhlových změn na tyto pomocné vložené bloky.



Programový kód	**Komentář**
ORIC	
N8 A2=… B2=… C2=…	
N10 X… Y… Z…	
N12 C2=… B2=…	; blok kruhového oblouku, který se vkládá na vnějších rozích, se rozdělí na N12 a N14 v souladu se změnou orientace Pohyb po kruhovém oblouku a změny orientace budou přitom prováděny souběžně.
N14 C2=… B2=…	
N20 X =…Y=… Z=… G1 F200	

Příklad 2: ORID

Jestliže je aktivní příkaz `ORID`, jsou všechny bloky mezi oběma posuvovými bloky uskutečňovány na konci prvního bloku posuvu. Blok kruhového oblouku s konstantní orientací bude uskutečněn bezprostředně před druhým blokem posuvu.



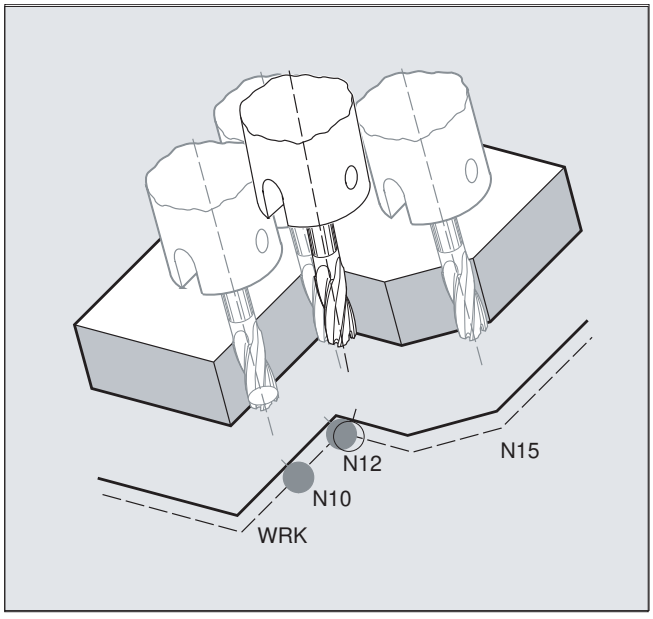
Programový kód	**Komentář**
ORID	
N8 A2=… B2=… C2=…	
N10 X… Y… Z…	
N12 A2=… B2=… C2=…	; bloky N12 a N14 se uskuteční na konci bloku N10 Potom bude následovat blok kruhového oblouku s momentálně nastavenou orientací.
N14 M20	; pomocné funkce atd.
N20 X… Y… Z…	

Poznámka

Pro způsob změny orientace na vnějším rohu je určující programový příkaz, který je aktivní v prvním bloku posuvu na vnějším rohu.

Beze změny orientace: Jestliže orientace na hranici bloku není změněna, potom se nástroj pohybuje po kruhové dráze, která se dotýká obou kontur.

Příklad 3: Změna orientace ve vnitřním rohu



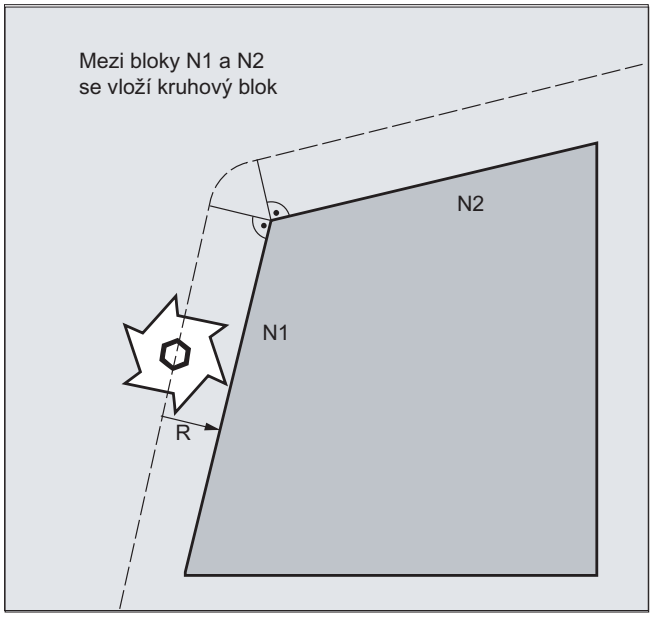
Programový kód
ORIC
N10 X …Y… Z… G1 F500
N12 X …Y… Z… A2=… B2=… C2=…
N15 X …Y… Z… A2=… B2=… C2=…

Další informace

Chování na vnějších rozích

Na vnějších rozích se vždy vkládá blok kruhového oblouku s rádiusem frézy.

Pomocí programových příkazů `ORIC`, příp. `ORID` může být definováno, zda se změny orientace, které byly naprogramovány mezi bloky `N1` a `N2`, uskuteční před zahájením zpracování vloženého bloku kruhového oblouku nebo současně s ním.



Jestliže je na vnějším rohu nezbytná změna orientace, potom si můžete vybrat, jestli se má uskutečnit souběžně s interpolací nebo odděleně od pohybu po dráze.

Je-li aktivní funkce `ORID`, uskuteční se napřed vložené bloky bez pohybu po dráze. Blok kruhového oblouku se vloží bezprostředně před druhý z obou bloků posuvu, kterými je roh tvořen.

Jestliže je na vnějším rohu vložen větší počet bloků změny orientace a pokud je aktivní funkce `ORIC`, potom se pohyb po kruhovém oblouku rozdělí na jednotlivé vkládané bloky v závislosti na počtu změn orientace.

Vyhlazení průběhu orientace pomocí příkazů OSD, příp. OST

V případě přechodových zaoblení aktivovaných funkcí `G642` se nemůže maximální odchylka pro konturové osy a orientační osy příliš lišit. Menší z obou tolerancí určuje formu pohybu na přechodových zaobleních, příp. toleranci úhlu, aby se průběh orientace relativně silně vyhladil, aniž by však přitom muselo docházet k velkým odchylkám od kontury.

Aktivováním příkazů `OSD`, příp. `OST` je možné zajistit, aby se s předem zadanou délkou přechodového zaoblení, příp. úhlovou tolerancí, probíhalo vyhlazení s jen velmi malými odchylkami průběhu orientace a bez velkých odchylek od kontury.

Poznámka

Narozdíl od vyhlazování kontury (a průběhu orientace) pomocí příkazu `G642` se při vyhlazování orientace pomocí příkazů `OSD`, příp. `OST` nevytváří žádný vlastní blok, nýbrž místo toho se zaoblovací pohyb vkládá přímo do naprogramovaných původních bloků.

Pomocí příkazů `OSD`, příp. `OST` nemohou být zaoblovány žádné přechody mezi bloky, ve který se uskutečňuje změna druhu interpolace pro orientaci nástroje (vektor --> kruhové osy, kruhové osy --> vektor). Tyto přechody mezi bloky mohou být v případě potřeby zaobleny pomocí běžných funkcí pro přechodová zaoblení `G641`, `G642`, příp. `G643`

4.13.7 Volné zadávání D-čísel, číslo břitu

4.13.7.1 Volné zadávání D-čísel, číslo břitu (adresa CE)

D-číslo

D-čísla mohou být stejně jako čísla korekcí zřetězena. Kromě toho je možné pomocí adresy CE určovat číslo břitu. Pomocí systémové proměnné $TC_DPCE může být číslo břitu popsáno.

Předdefinované nastavení: Číslo korekčních parametrů == číslo břitu

Prostřednictvím strojních parametrů jsou definovány maximální počet D-čísel (čísel břitů) a maximální počet břitů na jeden nástroj (--> výrobce stroje). Následující příkazy mají smysl jedině tehdy, pokud byl definován maximální počet břitů (MD18105) větší než počet břitů na jeden nástroj (MD18106). Věnujte prosím pozornost informacím od výrobce stroje.

Další informace

Příručka k funkcím, Nástroje

4.13.7.2 Volné zadávání D-čísel: Kontrola D-čísla (CHKDNO)

Pomocí příkazu `CHKDNO` zkontrolujete, zda příslušné D-číslo již bylo jednoznačně přiděleno. D-čísla všech v rámci jedné jednotky TO definovaných nástrojů se smí vyskytnout jen jedenkrát. Náhradní nástroje se přitom nezohledňují.

Syntaxe

```
state=CHKDNO(Tno1,Tno2,Dno)
```

Význam

`state`:	= TRUE:	D-číslo bylo pro kontrolovaný rozsah jednoznačně přiděleno.
	= FALSE:	Vyskytla se kolize D-čísel nebo nastavení parametrů je neplatné. Pomocí parametrů Tno1, Tno2 a Dno jsou předávány parametry, které měly za následek kolizi.. Tato data mohou být ve výrobním programu vyhodnocována.
`CHKDNO(Tno1,Tno2)`:	Budou zkontrolována všechna D-čísla uvedených nástrojů.	
`CHKDNO(Tno1)`:	Budou zkontrolována všechna D-čísla nástroje Tno1 proti všem ostatním nástrojům.	
`CHKDNO`:	Budou zkontrolována všechna D-čísla všech nástrojů proti všem ostatním nástrojům.	

4.13.7.3 Volné zadávání D-čísel: Přejmenovávání D-čísel (GETDNO, SETDNO)

D-čísla musí být přiřazována jednoznačně. Dva různé břity nástroje nemohou mít stejné D-číslo.

GETDNO

Tento příkaz zjišťuje D-číslo určitého břitu (ce) nástroje s T-číslem t. Jestliže žádné D-číslo k zadanému parametru neexistuje, bude nastaveno d=0. Pokud je D-číslo neplatné, bude výsledkem hodnota větší než 32000.

SETDNO

Pomocí tohoto příkazu přiřadíte hodnotu d D-číslu břitu ce nástroje t. Pomocí stavové proměnné je oznamován výsledek tohoto příkazu (TRUE nebo FALSE). Jestliže žádný datový blok k uvedeným parametrům neexistuje, bude výsledkem hodnota FALSE. Syntaktická chyba bude mít za následek alarm. D-číslo není možné explicitně nastavit na 0.

Syntaxe

```
d = GETDNO (t,ce)
state = SETDNO (t,ce,d)
```

Význam

d:	D-číslo břitu nástroje
t:	T-číslo nástroje
ce:	Číslo břitu (CE-číslo) nástroje
state:	Udává, zda bylo možné uskutečnit příkaz bezchybně (TRUE nebo FALSE).

Příklad - Přejmenování D-čísla

Programování	Komentář
`$TC_DP2[1,2] = 120`	
`$TC_DP3[1,2] = 5.5`	
`$TC_DPCE[1,2] = 3`	`; číslo břitu CE`
`...`	
`N10 def int DNrAlt, DNrNeu = 17`	
`N20 DNrAlt = GETDNO(1,3)`	
`N30 SETDNO(1,3,DNrNeu)`	

Tímto je břitu CE=3 přiřazena nová hodnota D-čísla 17. Nyní je přístup k datům tohoto břitu zajištěn přes D-číslo 17, a to jak prostřednictvím systémových proměnných, tak také při programování pomocí NC adresy.

4.13.7.4 Volné zadávání D-čísel: Zjišťování T-čísla k zadanému D-číslu (GETACTTD)

Předem definovaná funkce GETACTTD zjišťuje odpovídající T-číslo k absolutnímu D-číslu. Neuskutečňuje se žádná kontrola jednoznačnosti. Jestliže v rámci jedné jednotky TO existuje větší počet stejných D-čísel, bude výsledkem T-číslo prvního nalezeného nástroje.

Syntaxe

```
<status>=GETACTTD(<TNr>,<DNr>)
```

Význam

GETACTTD():	Volání funkce		
<DNr>:	D-číslo, pro které má být vyhledáno T-číslo.		
	Datový typ:	INT	
<TNr>:	Nalezené T-číslo		
	Datový typ:	VAR INT	
<status>:	Výsledek		
	Datový typ:	INT	
	Hodnota:	0	T-číslo bylo nalezeno. Parametr <TNr> obsahuje hodnotu T-čísla.
		-1	K zadanému D-číslu neexistuje žádné T-číslo, <TNr>=0.
		-2	D-číslo není absolutní. <TNr> obsahuje hodnotu prvního nalezeného nástroje, který má D-číslo, jehož hodnota odpovídá parametru <DNr>.
		-5	Funkce nemohla být z nějakého jiného důvodu uskutečněna.

4.13.7.5 Volné zadávání D-čísel: Nastavení D-čísla jako neplatného (DZERO)

4.13.8 Kinematika držáku nástroje

Předpoklady

Držák nástroje je schopen nastavit nástroj do všech možných prostorových orientací jen tehdy, pokud:

- Jsou k dispozici dvě kruhové osy V_1 a V_2
- Tyto kruhové osy jsou vůči sobě kolmé
- Podélná osa nástroje se nachází kolmo na druhou kruhovou osu V_2.

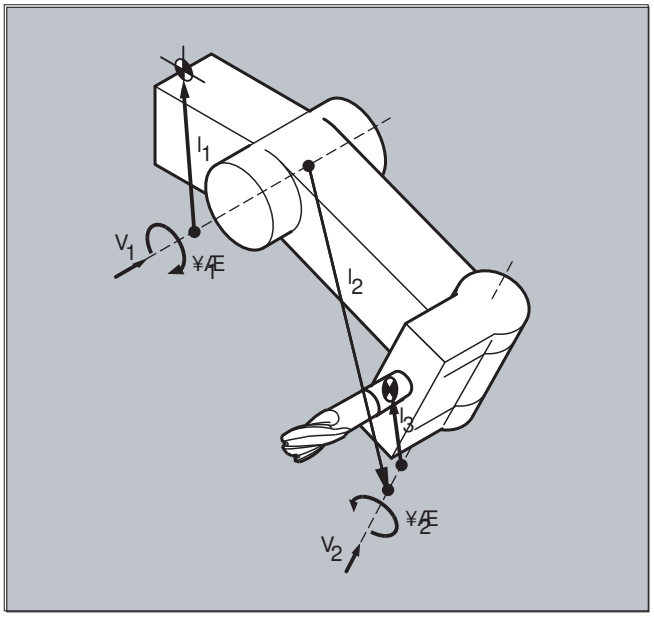
Kromě toho platí u strojů, u nichž musí být nastavitelné všechny možné orientace, následující předpoklad:

- Orientace nástroje nástroje musí být kolmo na první kruhovou osu V_1.

Funkce

Kinematika držáku nástroje s maximálně dvěma otočnými osami V_1 nebo V_2 je popsána pomocí 17 systémových proměnných $TC_CARR1[m] až $TC_CARR17[m]. Popis držáku nástroje se skládá z těchto částí:

- Vektorová vzdálenost od první kruhové osy po vztažný bod držáku nástroje I_1, vektorová vzdálenost od první ke druhé kruhové ose I_2 a vektorová vzdálenost od druhé kruhové osy ke vztažnému bodu nástroje I_3.
- Směrové vektory obou kruhových os V_1 a V_2
- úhlů otočení α1, α2 okolo obou os Úhly otočení se určují při pohledu ve směru vektorů kruhových os a ve směru hodinových ručiček jsou považovány za kladné.



Pro stroje s **kinematikou s více stupni volnosti** (jak nástroj, tak i obrobek se mohou otáčet) byly systémové proměnné rozšířeny o položky $TC_CARR18[m] až $TC_CARR23[m].

Parametr

Funkce systémových proměnných pro orientovatelný držák nástroje			
Popis	**složka x**	**složka y**	**složka z**
l_1 vektor offsetu	$TC_CARR1[m]	$TC_CARR2[m]	$TC_CARR3[m]
l_2 vektor offsetu	$TC_CARR4[m]	$TC_CARR5[m]	$TC_CARR6[m]
v_1 kruhová osa	$TC_CARR7[m]	$TC_CARR8[m]	$TC_CARR9[m]
v_2 kruhová osa	$TC_CARR10[m]	$TC_CARR11[m]	$TC_CARR12[m]
α_1 úhel otočení α_2 úhel otočení	$TC_CARR13[m] $TC_CARR14[m]		
l_3 vektor offsetu	$TC_CARR15[m]	$TC_CARR16[m]	$TC_CARR17[m]

Rozšíření systémových proměnných pro orientovatelný držák nástroje			
Popis	**složka x**	**složka y**	**složka z**
l_4 vektor offsetu	$TC_CARR18[m]	$TC_CARR19[m]	$TC_CARR20[m]
Identifikátor kruhové osy V_1 kruhové osy V_2	Identifikátor kruhových os V_1 a V_2 (předdefinované nastavení je nula) $TC_CARR21[m] $TC_CARR22[m]		
Typ kinematiky **T**ool **P**art **M**ixed mode	$TC_CARR23[m]		
	Typ kinematiky T ->	Typ kinematiky P ->	Typ kinematiky M ->
	Otočný je jen nástroj (předdefinované nastavení)	Otočný je jen obrobek	Obrobek i nástroj jsou otočné
Offset kruhové osy V_1 kruhové osy V_2	Úhel ve stupních kruhové osy V_1 a V_2 při zaujmutí základní polohy $TC_CARR24[m] $TC_CARR25[m]		
Úhlový offset kruhové osy V_1 kruhové osy V_2	Offset Hirthova ozubení kruhových os V_1 a V_2 ve stupních $TC_CARR26[m] $TC_CARR27[m]		
Úhlový inkrement v_1 kruhová osa v_2 kruhová osa	Inkrement Hirthova ozubení kruhových os V_1 a V_2 ve stupních $TC_CARR28[m] $TC_CARR29[m]		
Min. poloha kruhové osy V_1 kruhové osy V_2	Softwarové koncové spínače pro minimální pozici kruhových os V_1 a V_2 $TC_CARR30[m] $TC_CARR31[m]		
Max. poloha kruhové osy V_1 kruhové osy V_2	Softwarové koncové spínače pro maximální pozici kruhových os V_1 a V_2 $TC_CARR32[m] $TC_CARR33[m]		
Název držáku nástroje	Namísto čísla může být držáku nástroje přiřazen název. $TC_CARR34[m]		
Uživatel: Název osy 1 Název osy 2 Identifikace **Pozice**	Zamýšlené použití uživatelem v rámci měřicího systému. $TC_CARR35[m] $TC_CARR36[m] $TC_CARR37[m]		
	$TC_CARR38[m]	$TC_CARR39[m]	$TC_CARR40[m]
Jemné posunutí	Parametr, který se může **přičítat k hodnotám v základních parametrech**.		
l_1 vektor offsetu	$TC_CARR41[m]	$TC_CARR42[m]	$TC_CARR43[m]
l_2 vektor offsetu	$TC_CARR44[m]	$TC_CARR45[m]	$TC_CARR46[m]
l_3 vektor offsetu	$TC_CARR55[m]	$TC_CARR56[m]	$TC_CARR57[m]
l_4 vektor offsetu	$TC_CARR58[m]	$TC_CARR59[m]	$TC_CARR60[m]
v_1 kruhová osa	$TC_CARR64[m]		
v_2 kruhová osa	$TC_CARR65[m]		

Poznámka

Vysvětlivky k parametrům

Pomocí "m" se vždy zadává číslo popisovaného držáku nástroje.

Parametry $TC_CARR47 až $TC_CARR54, jakož i $TC_CARR61 až $TC_CARR63 nejsou definovány a v případě pokusu o přístup kvůli jejich čtení nebo zápisu se aktivuje alarm.

Počáteční, příp. koncové body vektorů vzdálenosti na osách mohou být vybrány libovolně. Pro úhly otočení α_1, α_2 okolo obou os je v základním postavení držáku nástroje definována poloha 0°. Kinematika držáku nástroje takto může být popsána libovolným množstvím možností.

Držák nástroje s jen jednou nebo s žádnou kruhovou osou může být popsán dosazením nul směrovým vektorům jedné nebo obou kruhových os.
U držáku nástrojů bez kruhové osy se vektory vzdálenosti uplatňují jako doplňkové korekce nástroje, jejichž komponenty nejsou při přepnutí roviny obrábění (G17 až G19) ovlivňovány.

Rozšíření parametrů

Parametry kruhových os

Systémové proměnné byly rozšířeny o položky $TC_CARR24[m] až $TC_CARR33[m] a jsou popsány následujícím způsobem:

Offset kruhové osy V_1, V_2	Změna polohy kruhové osy V_1 nebo V_2 v základní poloze orientovatelného držáku nástroje.
Úhlový offset / úhlový inkrement kruhové osy V_1, V_2	Offset nebo úhlový inkrement Hirthova ozubení kruhových os V_1 a V_2. Naprogramovaný nebo vypočítaný úhel se zaokrouhluje na nejbližší hodnotu, která je dána výrazem phi = s + n * d, kde n je celé číslo.
Minimální a maximální poloha kruhové osy V_1, V_2	Minimální pozice/maximální pozice kruhové osy, úhel otočení (softwarový koncový spínač) kruhové osy V1 a V2.

Parametry pro uživatele

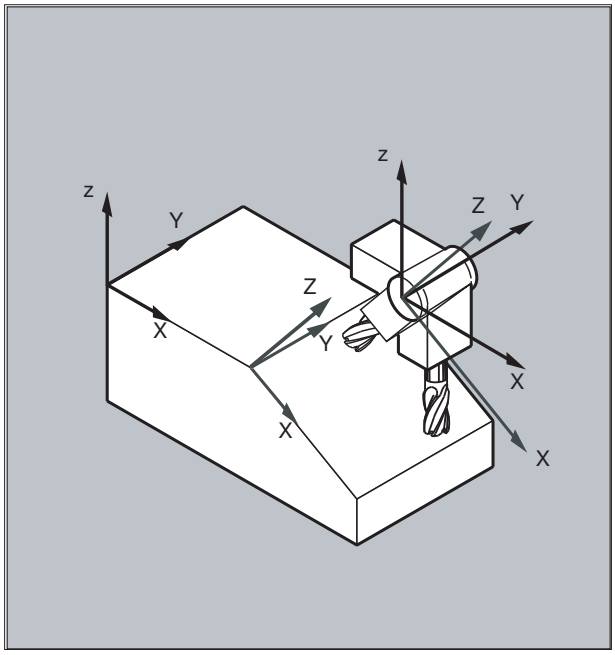
Proměnné $TC_CARR34 až $TC_CARR40 obsahují parametry, které jsou uživateli volně k dispozici a až do verze SW 6.4 standardně nejsou v rámci NCK dále vyhodnocovány a ani nemají žádný význam.

Parametry jemného posunutí

$TC_CARR41 až $TC_CARR65 obsahují parametry jemného posunutí, které se mohou přičítat k hodnotám základních parametrů. K hodnotě jemného posunutí přiřazené k určitému základnímu parametru se dostanete tak, že se k číslu parametru přičte hodnota 40.

Příklad:

Držák nástroje používaný v následujícím příkladu se dá úplně popsat pomocí otočení okolo osy Y.



Programový kód	**Komentář**
N10 $TC_CARR8[1]=1	; definice složky Y první kruhové osy držáku nástroje 1
N20 $TC_DP1[1,1]=120	; definice stopkové frézy
N30 $TC_DP3[1,1]=20	; definice stopkové frézy s délkou 20 mm
N40 $TC_DP6[1,1]=5	; definice stopkové frézy s rádiusem 5 mm
N50 ROT Y37	; definice framu s otočením o 37° okolo osy Y
N60 X0 Y0 Z0 F10000	; najíždění na výchozí pozici
N70 G42 CUT2DF TCOFR TCARR=1 T1 D1 X10	; nastavení korekce rádiusu, korekce délky nástroje v otočeném framu, aktivování držáku nástroje 1, nástroje 1
N80 X40	; uskutečnění obrábění s otočením 37°
N90 Y40	
N100 X0	
N110 Y0	
N120 M30	

Další informace

Kinematika s více stupni volnosti

Pro stroje s kinematikou s více stupni volnosti (jak nástroj, tak i obrobek se mohou otáčet) byly systémové proměnné rozšířeny o položky `$TC_CARR18[m]` až `$TC_CARR23[m]`, které jsou popsány následujícím způsobem:

Otočný nástrojový stůl se skládá z těchto částí:

- Vektorová vzdálenost mezi otočnou osou V_2 a vztažným bodem otočného nástrojového stolu I_4 třetí kruhové osy.

Kruhové osy se skládají z těchto součástí:

- Dva kanálové identifikátory pro popis vztahu mezi kruhovými osami V_1 a V_2, jejichž pozic systém v případě potřeby využívá při určování orientace orientovatelného držáku nástroje.

Typ kinematiky s hodnotou T, P nebo M:

- Typ kinematiky T: Otočný je jen nástroj.
- Typ kinematiky P: Otočný je jen obrobek.
- Typ kinematiky M: Obrobek i nástroj jsou otočné.

Vymazání dat držáku nástroje

Pomocí nastavení `$TC_CARR1[0]=0` mohou být data všech datových bloků držáku nástroje vymazána.

Pro typ kinematiky `$TC_CARR23[T]=T` musí být dosazeno jedno ze tří přípustných velkých nebo malých písmen (T, P, M) a z tohoto důvodu by proměnná neměla být vymazána.

Editace dat držáku nástroje

Kterákoli z popisovaných hodnot může být změněna přiřazením nové hodnoty ve výrobním programu. Jakýkoli jiný znak než T, P nebo M bude mít při pokusu o aktivování orientovatelného držáku nástroje za následek aktivování alarmu.

Načítání dat držáku nástroje

Kterákoli z popisovaných hodnot může být čtena přiřazením proměnné ve výrobním programu.

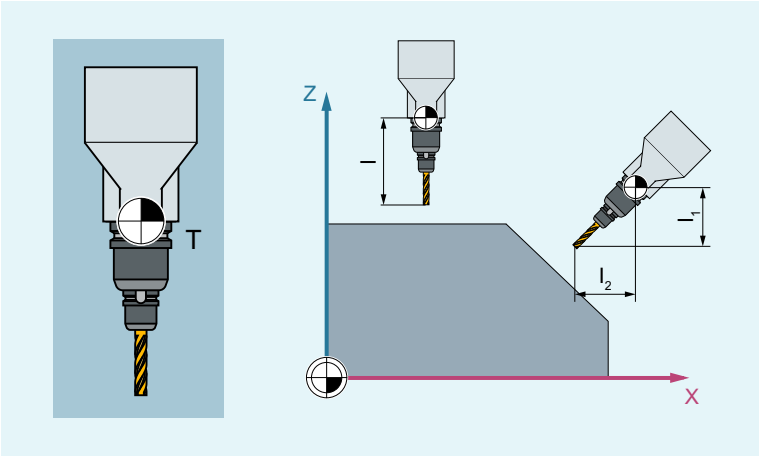
Jemná posunutí

Nepřípustná hodnota jemného posunutí bude rozpoznána až tehdy, když je aktivován orientovatelný držák nástroje, který takovou hodnotu obsahuje, a když je současně nastaven nastavovaný parametr SD42974 $SC_TOCARR_FINE_CORRECTION = TRUE.

Hodnota nepřípustného jemného posunutí je pomocí strojního parametru omezena na maximální přípustnou hodnotu.

4.13.9 Délková korekce nástroje pro orientovatelný držák nástroje (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ)

Při změně prostorové orientace držáku nástroje a v důsledku toho také orientace nástroje se mění také jeho složky délky.



T Vztažný bod držáku nástroje

l_1, l_2 Složky délky nástroje

Po přestavení, např. manuálním nastavením nebo změnou držáku nástroje s pevným prostorovým nasměrováním, je proto nutné znovu zjistit jednotlivé složky délky nástroje. K tomu slouží příkazy ze skupiny G-funkcí 42 „Držák nástroje“.

Syntaxe

```
TCARR=<m>
TCOABS
TCOFR
TCOFRZ/TCOFRY/TCOFRX
```

Význam

Prvek	Typ	Význam	
TCARR=<m>	Adresa	Vyžádání držáku nástroje	
		<m>	Číslo držáku nástroje
TCOABS	G-příkaz	Výpočet složek délky nástroje z aktuální orientace držáku nástroje	
TCOFR	G-příkaz	Stanovení složek délky nástroje z orientace aktivního framu	

Prvek	Typ	Význam	
TCOFRZ/TCOFRY/ TCOFRX	G-příkaz	Příkaz TCOFRX/TCOFRY/TCOFRZ vychází z nástroje orientovaného v odpovídajícím směru (X/Y/Z) a vypočítává úhel nastavení orientovatelného držáku nástroje tak, aby byl nástroj v aktivním framu orientován ve stejném směru.	
		TCOFRZ	Nástroj orientovaný ve směru osy Z se nasměruje tak, aby byl v aktivním framu rovněž orientován ve směru osy Z.
		TCOFRY	Nástroj orientovaný ve směru osy Y se nasměruje tak, aby byl v aktivním framu rovněž orientován ve směru osy Y.
		TCOFRX	Nástroj orientovaný ve směru osy X se nasměruje tak, aby byl v aktivním framu rovněž orientován ve směru osy X.

Další informace

Volba držáku nástroje (TCARR)

S držákem nástroje je spojen datový blok držáku nástroje, který popisuje jeho geometrii. Pokud je v okamžiku volby držáku nástroje aktivní nějaký nástroj, bude přiřazen nově vybranému držáku nástroje a geometrické parametry tohoto držáku nástroje budou okamžitě v platnosti. Výměna nebo nové naprogramování aktivního nástroje nejsou zapotřebí. Jestliže je nástroj znovu aktivován, potom se s ním zachází tak, jako kdyby byl namontován v aktivním držáku nástroje.

Pomocí příkazu `TCARR=0` se aktuální držák nástroje deaktivuje.

Poznámka

Aktuální geometrické údaje o držáku nástroje mohou být definovány i v NC programu prostřednictvím odpovídajících systémových proměnných (viz kapitola „Přístup k datovým blokům držáku nástroje").

Poznámka

Z pohledu řídícího systému jsou čísla držáků nástroje <m> a čísla nástrojů <n> volně kombinovatelná. V reálných aplikacích mohou být některé kombinace z důvodů obrábění a z mechanických důvodů vyloučeny. To, zda je daná kombinace smysluplná, řídící systém nekontroluje.

Přístup k datovým blokům držáku nástroje

Z NC programu jsou možnosti přístupu k aktuálním geometrickým parametrům držáku nástroje následující:

- Zápis:
  `$TC_CARR<n>[<m>]=<hodnota>`
- Čtení:
  `<hodnota>=$TC_CARR<n>[<m>]`
  (<hodnota> musí být proměnná datového typu REAL.)

Jestliže držák nástroje, na který se příkaz odkazuje, není definován, aktivuje se alarm.

Vynulování datových bloků držáku nástroje

Z NC programu mohou být prostřednictvím jednoho příkazu všechny hodnoty ve všech datových blocích držáku nástroje vymazány:

```
$TC_CARR1[0] = 0
```

Selektivně mohou být jednotlivé datové bloky držáku nástroje mazány pomocí předem definované procedury DELTC.

Funkce volby držáku nástroje

Držák nástroje je v platnosti jen tehdy, pokud byly aktivovány jak držák nástroje, tak také nástroj. Aktivování samotného držáku nástroje žádný efekt nemá.

Funkce volby držáku nástroje závisí na příkazu ze skupiny G-funkcí 42 (držák nástroje), který byl naprogramován spolu s příkazem TCARR.

Když je aktivní držák nástroje, způsobí změna příkazu G-funkce z této skupiny nový výpočet složek délky nástroje.

Výpočet korekcí délky nástroje z orientace držáku nástroje (TCOABS)

Příkaz TCOABS vypočítává korekce délky nástroje z aktuálního úhlu orientace držáku nástroje a ukládá je do systémových proměnných $TC_CARR13 a $TC_CARR14. Započítávaná orientace je nezávislá na orientaci aktivního framu.

Budete-li potřebovat nastudovat definici kinematiky držáku nástroje pomocí systémových proměnných, viz kapitola „Kinematika držáku nástroje (Strana 780)" v Programovací příručce, Programování NC systémů.

Stanovení složek délky nástroje z orientace aktivního framu (TCOFR)

V případě příkazu TCOFR se úhel orientace držáku nástroje zjišťuje z orientace aktivního framu. Hodnoty uložené v parametrech držáku nástroje zůstávají ale nezměněné. Tyto hodnoty se kromě toho používají k tomu, aby bylo možno vyloučit nejednoznačnost, která může vyplynout při výpočtu úhlu otočení z framu (viz následující odstavec).

Nejednoznačnosti

Určité orientace nástroje, které byly zadány prostřednictvím framu, mohou být pomocí dvou os obecně nastaveny pomocí **dvou** různých dvojic úhlů otočení. Řídící systém vybírá z těchto dvou možných nastavení to, u kterého leží úhly otočení co možno nejblíže naprogramovanému úhlu otočení.

Poznámka

V případech, ve kterých se může nejednoznačnost vyskytnout, je zpravidla zapotřebí do parametrů držáku nástroje uložit úhel, který je z framu očekáván.

Orientace nástroje v aktivním framu (TCOFRZ/TCOFRY/TCOFRX)

Příkazy TCOFRX/TCOFRY/TCOFRZ pro výpočet úhlového nastavení držáku nástroje vycházejí z orientace ve směrech X/Y/Z, a to nezávisle na aktivní orientaci. Aktivování roviny obrábění (G17/G18/G19) nemá žádný vliv na orientaci v aktivním framu. Má ale vliv na započítání korekcí délky nástroje.

Změna framu

Uživatel může po aktivování nástroje frame změnit. To nemá žádný vliv na složky korekcí délky nástroje.

Úhly otočení uložené v parametrech držáku nástroje nejsou prostřednictvím úhlů otočení uložených ve framu nijak ovlivněny. Při přepnutí z TCOFR na TCOABS se znovu stává platným původní (naprogramovaný) úhel otočení v parametrech držáku nástroje.

Nový výpočet korekce délky nástroje

Aby se uskutečnil nový výpočet korekce délky nástroje po změně framu, musí být nástroj ještě jednou aktivován. Při výpočtu délkové korekce nástroje se v jednom z mezikroků vypočítává také úhel natočení držáku nástroje. Protože u držáků nástrojů se dvěma kruhovými osami obecně existují také dva páry úhlů natočení, pomocí kterých je možné přizpůsobit orientaci nástroje právě aktivnímu framu, musí hodnoty úhlu natočení uložené v parametrech držáku nástroje alespoň přibližně odpovídat mechanicky nastaveným úhlům natočení.

Poznámka

Téměř ve všech případech, ve kterých se může nejednoznačnost vyskytnout, je zapotřebí do parametrů držáku nástroje uložit úhel, který je z framu očekáván.

Poznámka

Orientace nástroje musí být manuálně přizpůsobena aktivnímu framu.

Poznámka

Může se stát, že řídící systém nebude schopen kontrolovat, jestli je možné úhly otočení vypočítané na základě orientace framu nastavit na stroji. Pokud je stroj konstrukčně uspořádán tak, že kruhové osy držáku nástroje nebudou schopny dosáhnout orientaci nástroje vypočítanou na základě orientace framu, bude generován alarm.

Poznámka

Kombinace jemné korekce nástroje a funkcí pro korekci délky nástroje nejsou u pohyblivých držáků nástrojů přípustné. Při pokusu vyvolat obě funkce současně se vypíše chybové hlášení.

Poznámka

Pomocí příkazu TOFRAME (Strana 335) je možné definovat frame na základě směru orientace zvoleného držáku nástroje.

Poznámka

Když je aktivní transformace orientace (transformace ve 3, 4, 5 osách), je možné zvolit držák nástroje s orientací odlišující se od nulové polohy, aniž by byl přitom generován alarm.

Poznámka

V případě aktivního framu bez otočení vede nastavení orientace pomocí příkazů TCOFRX/ TCOFRY/TCOFRZ k triviálnímu řešení, protože orientace nástroje je již nastavena ve směru X, Y nebo Z.

Nastavení orientace ve směru osy Z (TOFRAME)

Pomocí příkazu TOFRAME ze skupiny G-funkcí 53 (otáčení framu vztahující se na nástroj) je možné definovat frame, jehož osa Z je nasměrována rovnoběžně se směrem orientace zvoleného držáku nástroje.

Pokud není aktivní žádný držák nástroje nebo pokud je aktivní držák nástroje beze změny orientace, potom je směr Z v novém framu následující:

- V případě roviny G17 se rovná starému směru osy Z.
- V případě roviny G18 se rovná starému směru osy Y.
- V případě roviny G19 se rovná starému směru osy X.

Předávané parametry pro standardní a měřicí cykly

Pro předávané parametry standardních a měřicích cyklů platí definované rozsahy hodnot.

Pokud jde o hodnoty úhlu, jsou definovány následující intervaly hodnot:

Otočení okolo ...	Rozsah hodnot
1. geometrická osa	-180° ≤ úhel ≤ +180°
2. geometrická osa	-90° ≤ úhel ≤ +90°
3. geometrická osa	-180° ≤ úhel ≤ +180°

Poznámka

Při předávání hodnot úhlů standardním a měřicím cyklům je potřeba mít na paměti:

Hodnoty menší než je přesnost výpočtu NC systému jsou zaokrouhleny na nulu!

Přesnost výpočtu úhlových pozic v NC systému ve definována ve strojním parametru:

MD10210 $MN_INT_INCR_PER_DEG

4.13.10 Modifikace orientovatelného držáku nástroje po měření stroje (CORRTC)

Pomocí funkce `CORRTC` mohou být změřené prvky kinematického řetězce náležející do držáku Toolcarrier zapsány do speciálních korekčních prvků.

Poznámka

Hodnoty korekcí, které byly zapsány pomocí funkce CORRTC, nejsou ve funkci ToolCarrier v platnosti okamžitě. Začnou se uplatňovat až po deaktivování funkce ToolCarrier, vyvolání příkazu NEWCONF a po opětovném aktivování funkce ToolCarrier.

Syntaxe

```
<_Corr_Status> = CORRTC(<_Corr_Vect>, <_Corr_Index>, <_Corr_Mode>,
[ <_No_Alarm>])
```

Význam

CORRTC:	Volání funkce		
<_Corr_Status>:	Výsledná hodnota funkce		
	Datový typ:	INT	
	Hodnoty:	0	Funkce byla zpracována bezchybně.
		1	Není aktivní žádný držák nástroje ToolCarrier.
		2	Aktivní držák nástroje ToolCarrier nebyl definován prostřednictvím kinematického řetězce.
		10	Parametr <_Corr_Index> sloužící při volání je záporný.
		11	Parametr <_Corr_Mode> sloužící při volání je záporný.
		12	Neplatný odkaz na dílčí úsek v řetězci znaků (_CORR_INDEX).
		13	V úseku, na který odkazuje parametr _CORR_INDEX, není definován žádný prvek korekce ($TC_CARR_CORR_ELEM).
		20	Místo stovek parametru _CORR_MODE je neplatné. Přípustné jsou pouze hodnoty 0 a 1.
		21	Místo tisíců parametru _CORR_MODE je neplatné. Přípustné jsou pouze hodnoty 0 a 1.
		30	U korekce offsetového vektoru je odchylka oproti aktuální hodnotě u minimálně jedné souřadnice větší, než je předem stanovená maximální hodnota definovaná v nastavovaném parametru SD41612 $SN_CORR_TOCARR_LIN_MAX.
		31	Pokus o zápis do systémové proměnné byl kvůli chybějícímu oprávnění k zápisu zablokován.
<_Corr_Vect>:	Vektor korekce Obsah vektoru korekce je definován prostřednictvím následujících parametrů <Corr_Index> a <_Corr_Mode>. Pokud je <_Corr_Status> = 30, obsah vektoru se přepisuje (viz výše.).		
	Datový typ:	REAL	
<_Corr_Index>:	Označení úseku, pro který se má uskutečnit korekce směrového vektoru korekčního prvku.		
	Datový typ:	INT	

<_Corr_Mode>:	Režim korekce		
	Datový typ:	INT	
	U parametru <Corr_Mode> se používá desítkové kódování (místo jednotek až tisíců):		
	Místo jednotek:	rezervováno	
	Místo desítek:	Určuje, jak má být prvek korekce, na který odkazuje obsah parametru <_Corr_Index>, modifikován.	
		xx**0**x	Vektor korekce se okamžitě zapisuje do prvku korekce. Tato varianta se může použít, aby se prvek korekce zapisoval okamžitě, aniž by musel být znám index <n> příslušného systémového parametru ($NK_OFF_DIR[<n>, ...]).
		xx**1**x	Stejně jako u hodnoty 0, ale s tím rozdílem, že předávaná hodnota korekce je interpretována v globálních souřadnicích. Rozdíl mezi variantou 0 a 1 může vzniknout vždy, když kinematický řetězec v základním nastavení (pozice všech kruhových os se rovnají 0) obsahuje další otočení.
		xx**2**x	Stejné jako 1, ale s tím rozdílem, že hodnota korekce se vztahuje na celý úsek, tzn. že v prvku korekce je uložena taková hodnota, že celý úsek dosahuje délky definované hodnotou korekce.
	Místo stovek:	Určuje, jak má být obsah parametru <_Corr_Vect> interpretován.	
		x**0**xx	Předávaný vektor korekce <_Corr_Vect> obsahuje celkovou novou délku prvku korekce, příp. úseku, na který odkazuje parametr <_Corr_Index> ve spojení s místem desítek parametru <_Corr_Mode> (absolutní korekce).
		x**1**xx	Předávaný vektor korekce <_Corr_Vect> obsahuje jen rozdíl oproti aktuální délce prvku korekce, příp. úseku, na který odkazuje parametr <_Corr_Index> ve spojení s místem desítek parametru <_Corr_Mode> (inkrementální korekce).
	Místo tisíců:	Určuje, zda má nebo nemá být maximální přípustná hodnota korekce omezena nastavovaným parametrem $SN_ CORR_TOCARR_LIN_MAX.	
		0xxx	Monitorování mezní hodnoty je aktivní.
		1xxx	Monitorování mezní hodnoty je potlačeno.
<_No_Alarm>:	Chování v případě chyby (vracená hodnota > 0) (**volitelné**)		
	Datový typ:	BOOL	
	Hodnota:	FALSE (předdefinované)	V případě chyby se zpracovávání programu pozastaví a aktivuje se alarm.
		TRUE	V případě chyby se zpracovávání programu nepozastaví a neaktivuje se žádný alarm. Případ použití: Specifická uživatelská reakce v závislosti na výstupní hodnotě.

Další informace k funkci CORRTC

Kinematická struktura držáku nástroje ToolCarrier je popsána prostřednictvím jednoho (typ T a typ P) nebo dvou (typ M) kinematických řetězců (dílčích řetězců), které vycházejí z odpovídajícího vztažného bodu (vztažný bod stroje nebo vztažný bod držáku nástroje). Jeden z těchto dvou řetězců, řetězec nástroje, končí ve vztažném bodě nástroje, zatímco ten druhý, řetězec obrobku, končí v počátku základního souřadného systému.

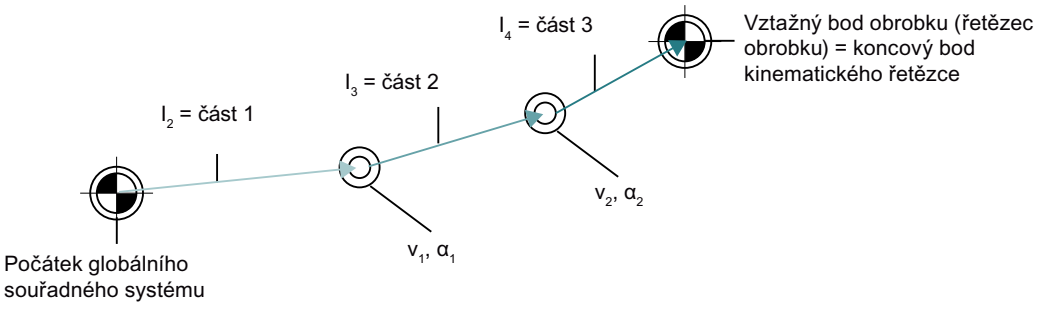
Funkce CORRTC zapisuje u strojů s funkcí ToolCarrier do speciálních korekčních prvků směry os. Kinematický řetězec je mimo jiné popsán prvky typu OFFSE, které jsou definovány prostřednictvím proměnné $NK_TYPE.

Funkce CORRTC pracuje s úseky

Oba dílčí řetězce mohou být rozděleny na maximálně čtyři úseky:

- Úsek 1 začíná v počátečním bodě řetězce a končí v první kruhové ose.
- Úsek 2 je vzdálenost mezi kruhovou osou 1 a kruhovou osou 2.
- Úsek 3 je vzdálenost mezi kruhovou osou 2 a koncem řetězce.

Následující obrázek znázorňuje orientovatelný držák nástroje se 2 kruhovými osami.



Obrázek 4-12 Příklad funkce CORRTC

Úseky jsou definovány jednoznačně: Když budete procházet dílčím kinematickým řetězcem od jeho počátečního bodu do koncového bodu, má první úsek index 1, následující úsek index 2 atd.

Prvky korekce

Na kterýkoli konstantní prvek kinematického řetězce (prvek řetězce typu $NK_TYPE[<n>] = "OFFSET") v každém tomto úseku je možno se odkázat pomocí systémových proměnných $TC_CARR_CORR_ELEM [, 0 ... 3]. Do takto označených prvků je možno pomocí funkce CORRTC zapsat hodnoty korekcí, které byly zjištěny při měření stroje.

Posloupnost odkazů v proměnné $TC_CARR_CORR_ELEM[m, 0..3] musí korespondovat s výše popsanými úseky, tzn. v proměnné $TC_CARR_CORR_ELEM[m, 0] se může nacházet jen prvek řetězce, který patří k offsetovému vektoru **l**1 atd.

Vztažná hodnota je přitom vždy odpovídající hodnota, která je v platnosti při volání aktivního držáku ToolCarrier s funkcí CORRTC. Obsah dat kinematiky v datové paměti, který byl změněn po aktivování držáku ToolCarrier, nemá žádný vliv na způsob, jakým se funkce CORRTC chová.

4.13.11 On-line korekce délky nástroje (TOFFON, TOFFOF)

Prostřednictvím systémové proměnné $AA_TOFF[<n>] mohou být trojrozměrně a v reálném čase korigovány efektivní délky nástroje ve všech třech jeho rozměrech.

Jako index <n> se používají tři identifikátory geometrických os. Tímto způsobem je definován počet aktivních směrů korekce prostřednictvím geometrických os, které jsou v daném okamžiku aktivní.

Všechny korekce mohou být aktivní současně.

Funkce On-line korekce délky nástroje se může používat v těchto případech:

- Transformace orientace TRAORI
- Orientovatelný držák nástroje TCARR

Poznámka

On-line korekce délky nástroje je **volitelný doplněk**, který musí být napřed instalován. Tato funkce má smysl jedině tehdy, když je aktivní transformace orientace nebo když je aktivní orientovatelný držák nástroje.

Syntaxe

```
TRAORI
TOFFON(<směr korekce>[,<hodnota offsetu>])
WHEN TRUE DO $AA_TOFF[<směr korekce>]                    ; v synchronních akcích
...
TOFFOF(<směr korekce>)
```

Význam

TOFFON:	**Aktivování** on-line korekce délky nástroje	
	<směr korekce>:	Směr nástroje (X, Y, Z), ve kterém se má on-line korekce délky nástroje uplatňovat.
	<hodnota offsetu>:	Při aktivování může být pro odpovídající směr korekce zadána hodnota offsetu, přičemž na tento offset se bude ihned najíždět.
TOFFOF:	**Vynulování** on-line korekce délky nástroje Hodnoty korekce v zadaném směru budou vynulovány a aktivuje se přitom zastavení předběžného zpracování.	

Příklady

Příklad 1: Aktivování korekce délky nástroje

Programový kód	Komentář
MD21190 $MC_TOFF_MODE =1	; najíždí se na absolutní hodnoty
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)	; aktivování transformace
N20 TOFFON(Z)	; aktivování On-line korekce délky nástroje pro směr Z
N30 WHEN TRUE DO $AA_TOFF[Z]=10 G4 F5	; pro rozměr nástroje ve směru osy Z se interpoluje korekce délky nástroje 10
...	
N100 XOFFSET=$AA_TOFF_VAL[X]	; přiřazení aktuální korekce ve směru osy X
N120 TOFFON(X,-XOFFSET) G4 F5	; pro rozměr nástroje ve směru osy X se najede zpátky na hodnotu korekce délky nástroje 0

Příklad 2: Deaktivování korekce délky nástroje

Programový kód	Komentář
N10 TRAORI(1)	; aktivování transformace
N20 TOFFON(X)	; aktivování On-line korekce délky nástroje pro směr X
N30 WHEN TRUE DO $AA_TOFF[X]=10 G4 F5	; pro rozměr nástroje ve směru osy X se interpoluje korekce délky nástroje 10
...	
N80 TOFFOF(X)	; offset polohy ve směru X nástroje se vymaže
	...$AA_TOFF[X]=0
	Žádná osa se nebude pohybovat.
	K aktuální poloze ve WCS se připočítá offset polohy v souladu s momentální orientací.

Další informace

Příprava bloku

Při přípravě bloku v předběžném zpracování jsou v rámci hlavního zpracování započítávány také momentálně platné offsety délky nástroje. Aby bylo možné v maximální možné míře využívat nejvyšší možnou rychlost pohybu os, je zapotřebí v době, kdy se započítávají offsety nástroje, přerušit přípravu bloku pomocí zastavení předběžného zpracování příkazem STOPRE.

Offset nástroje je v okamžiku předběžného zpracování znám také vždy tehdy, pokud se korekce délky nástroje po spuštění programu už nemění nebo pokud bylo po změně korekce délky nástroje zpracováno více bloků, než kolik jich může být uloženo do vyrovnávací paměti IPO mezi předběžným a hlavním zpracováním.

Proměnná $AA_TOFF_PREP_DIFF

Hodnota rozdílu mezi korekcí, která je momentálně v platnosti v interpolátoru, a korekcí, která byla v platnosti v okamžiku přípravy bloku, může být zjištěna v proměnné $AA_TOFF_PREP_DIFF[<n>].

Nastavení strojních parametrů a nastavovaných parametrů

Pro On-line korekci délky nástroje máte k dispozici následující systémové proměnné:

- MD20610 $MC_ADD_MOVE_ACCEL_RESERVE (rezerva zrychlení pro superponovaný pohyb)
- MD21190 $MC_TOFF_MODE
  Na obsah systémové proměnné $AA_TOFF[<n>] se buď najede jako na absolutní hodnotu nebo se tato hodnota integruje do výpočtů.
- MD21194 $MC_TOFF_VELO (rychlost on-line korekce délky nástroje)
- MD21196 $MC_TOFF_ACCEL (zrychlení on-line korekce délky nástroje)
- Nastavovaný parametr pro zadání mezních hodnot:
  SD42970 $SC_TOFF_LIMIT (horní mezní hodnota korekce délky nástroje)

4.13.12 Modifikace korekčních parametrů u točivých nástrojů

4.13.12.1 Výpočet orientace (ORISOLH)

Předdefinovaná funkce ORISOLH pomáhá uživateli nastavovat polohy kruhových os stroje tak, aby byl soustružnický nástroj přestaven do definované polohy vzhledem k obrobku, a to nezávisle na kinematice stroje. Předpokladem je, že je aktivována 6-osá transformace, jejíž parametry jsou nastaveny prostřednictvím kinematického řetězce.

K dispozici jsou dvě základní funkce:

- Nastavení orientace nástroje
  Jsou zadávány oba úhly β a γ. Funkce vypočítává pro daný účel potřebné úhly tří orientačních os.
- Přímé nastavování orientace nástroje
  Je zadán úhel druhé a třetí orientační osy. Funkce na základě těchto údajů vypočítává odpovídající úhly β a γ, jakož i úhel dosud chybějící první orientační osy.

Poznámka

Posloupnost orientačních os

Když se prochází kinematickým řetězcem, který popisuje konstrukci stroje od obrobku až po nástroj, platí pro posloupnost tří orientačních os 6-osé transformace následující pravidla:

- Orientační osa, která leží nejblíže **obrobku**, je **první** orientační osou.
- Orientační osa, která leží nejblíže **nástroji**, je **třetí** orientační osou.

První orientační osou je zpravidla vřeteno a odpovídající otočení se proto v těchto případech realizuje prostřednictvím otočného framu.

Syntaxe

```
<RetVal> = ORISOLH(<Cntrl>,<W1>,<W2>)
```

Význam

ORISOLH:	Volání funkce		
<RetVal>:	Výsledná hodnota funkce		
	Datový typ:	INT	
	Rozsah hodnot:	0, -2, -3, ..., -17	
	Hodnoty:	0	Funkce byla ukončena bezchybně.
		-2	Není aktivní žádná platná transformace (6-osá transformace orientace).
		-3	První parametr (<Cntrl>) je záporný.
		-4	Místo jednotek prvního parametru (<Cntrl>) je neplatné. Přípustné jsou pouze hodnoty 0 a 1.
		-5	Místo desítek prvního parametru (<Cntrl>) je neplatné. Přípustné jsou pouze hodnoty 0 až 3.
		-6	Místo stovek prvního parametru (<Cntrl>) je neplatné. Přípustné jsou pouze hodnoty 0 a 1.
		-7	Místo tisíců prvního parametru (<Cntrl>) je neplatné. Přípustné jsou pouze hodnoty 0 až 3.
		-8	U funkce „Přímé nastavování orientace nástroje" je úhel γ příliš velký.
		-9	U funkce „Přímé nastavování orientace nástroje" minimálně jedna ze zadaných pozic osy narušuje mezní hodnotu osy.
		-10	Není aktivní žádný nástroj.
		-11	Požadovaná orientace není nastavitelná.
		-12	Přizpůsobení volného úhlu osy u Hirthova ozubení není pro první nebo jediné řešení možné.
		-13	Přizpůsobení volného úhlu osy u Hirthova ozubení není pro druhé řešení možné.
		-14	Přizpůsobení volného úhlu osy u Hirthova ozubení není pro žádné z obou řešení možné.
		-15	V parametrech je nastaveno, že první orientační osa je Hirthova osa.
		-16	V parametrech je nastaveno, že jak druhá, tak i třetí kruhová osa je Hirthovou osou. Hirthovou osou smí být maximálně jedna z obou těchto os.
		-17	U funkce „Přímé naklápění" je minimálně jedna ze zadaných poloh osy nekompatibilní s odpovídajícím Hirthovým ozubením.

<Cntrl>:	Ovládá chování funkce		
	Datový typ:	INT	
	U parametru <Cntrl> se používá desítkové kódování (místo jednotek až tisíců):		
	Místo jednotek:	Místo jednotek ovládá chování v případě chyby.	
		xxx**0**	V případě chyby (výsledná hodnota < 0) je zpracovávání programu přerušeno a aktivuje se alarm 14106. **Poznámka:** Tento alarm se aktivuje nezávisle na hodnotě na místě jednotek také tehdy, když je hodnota parametru <Cntrl> záporná.
		xxx**1**	V případě chyby (výsledná hodnota < 0) se neaktivuje **žádný** alarm. Uživatel má v programu možnost sám vhodným způsobem reagovat.
	Místo desítek:	Ovládání chování v případě, že je k dispozici jedna orientační osa s Hirthovým ozubením. **Poznámka:** Tento parametr je vyhodnocován pouze v případě funkce „Nastavení orientace nástroje" (tzn. jestliže je na místě stovek hodnota „0").	
		xx**0**x	Poloha osy je zaokrouhlována na následující pozici.
		xx**1**x	Poloha osy je zaokrouhlována tak, že odchylka úhlu β je od jeho naprogramované hodnoty minimální.
		xx**2**x	Poloha osy je zaokrouhlována tak, aby se úhle β rovnal největší možné hodnotě, která je menší než naprogramovaná hodnota (úhel β se zaokrouhluje dolů).
		xx**3**x	Poloha osy je zaokrouhlována tak, aby se úhle β rovnal nejmenší možné hodnotě, která je větší než naprogramovaná hodnota (úhel β se zaokrouhluje nahoru).
	Místo stovek:	Udává, jaká funkce se má uskutečňovat, příp. jaký význam mají dva následující parametry <W1> a <W2>.	
		x**0**xx	Funkce „Nastavení orientace nástroje" Parametry <W1> a <W2> mají následující význam: • <W1> = β • <W2> = γ Vypočítávají se odpovídající úhly orientačních os.
		x**1**xx	Funkce „Přímé nastavování orientace nástroje" <W1> je údaj polohy pro druhou orientační osu, <W2> je údaj polohy pro třetí orientační osu v 6-osé transformaci. Určí se poloha první orientační osy a úhly β a γ, které jsou kompatibilní s oběma předem zadanými údaji poloh. Jestliže se nevyskytne žádná chyba, v systémové proměnné $P_ORI_POS[<n>, <m>] jsou vždy poskytnuta dvě řešení. První index <n> (0 nebo 1) přitom odkazuje na řešení a druhý index <m> (0 ... 2) odkazuje na orientační osu: • $P_ORI_POS[0/1, **0**]: Poloha první orientační osy • $P_ORI_POS[0/1, **1**]: Úhel β • $P_ORI_POS[0/1, **2**]: Úhel γ Kontroluje se, zda údaje polohy <W1> a <W2> jsou kompatibilní s eventuálně aktivním Hirthovým ozubením nebo s aktivními softwarovými mezními hodnotami. Jestliže tomu tak

			není, je výsledkem odpovídající číslo chyby (viz parametr <RetVal>). Jestliže jsou oba úhly <W1> a <W2> zvoleny libovolně, břit nástroje se obecně nenachází v rovině obrábění. Úhel γ, o který je břit z roviny obrábění otočený, nesmí být větší než určitá mezní hodnota, která je stanovena prostřednictvím nastavovaného parametru SD42999 $SC_ORISOLH_INCLINE_TOL.
	Místo tisíců:	Udává, jak jsou pozice vyplývající z řešení v případě potřeby upravovány, jestliže je na místě stovek hodnota „0“, tzn. když je aktivována funkce „Nastavení orientace nástroje“.	
		0xxx	Vypočítaná pozice osy by se měla nacházet co možno nejblíže momentální polohy strojní osy.
		1xxx	Vypočítané pozice os by měly být v případě os typu modulo co možno nejblíže středu oblasti modulo, u jiných os co možno nejblíže 0. V případě os, které nejsou typu modulo, to znamená, že polohy os jsou omezeny na rozmezí -180° ... 180°.
		2xxx	Vypočítané polohy os mají být nezávisle na jejich typu omezeny na rozmezí -180° ... 180°.
<W1>:	První úhel Význam vyplývá z hodnoty na místě stovek parametru <Cntrl>.		
	Datový typ:	REAL	
<W1>:	Druhý úhel Význam vyplývá z hodnoty na místě stovek parametru <Cntrl>.		
	Datový typ:	REAL	

Poznámka

Nenaprogramovaným parametrům je dosazována standardní hodnota "0".

Další informace

Počet nalezených řešení spolu s dalšími stavovými informace je možno při zpracování funkce ORISOLH načítat prostřednictvím následujících systémových proměnných:

Systémové proměnné	Význam		
$P_ORI_POS [<n>, <m>]	Tato proměnná obsahuje úhly os pro nastavení orientace, které vyplývají z naprogramované orientace.		
	<n>:	Index řešení	
		Rozsah hodnot:	0, 1
	<m>:	Index orientační osy	
		Rozsah hodnot:	0 ... 2
			Posloupnost orientačních os (1 ... 3) se vztahuje na definici os v parametru $NT_ROT_AX_NAME.
	Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje" obsahují proměnné $P_ORI_POS[0/1, 1] a P_ORI_POS[0/1, 2] hodnoty obou úhlů β a γ, které patří k oběma řešením. První řešení uložené v proměnné $P_ORI_POS[<n>, <m>], tzn. s indexem <n> = 0, je vždy to, které je vybráno řídícím systémem, když se přímo najíždí na požadovanou orientaci. Druhý index <m> se přitom vztahuje na orientační osu, tzn. na $NT_ROT_AX_NAME. Polohy os, které jsou uloženy v proměnné $P_ORI_POS[<n>, <m>] zohledňují offsety, které jsou uloženy v proměnných $NK_OFF a $NK_OFF_FINE, tzn. že tyto úhly os se mohou v následujících blocích bez dalších úprav používat, aby byla nastavena požadovaná orientace. Pokud je kruhová osa Hirthovou osou, jsou pozice vyplývající z daného řešení zaokrouhlovány na nejbližší následující stabilní polohu Hirthova ozubení. V systémových proměnných $P_ORI_DIFF mohou být pro kruhové osy s Hirthovým ozubením načítány rozdíly poloh os, aby bylo možné pracovat s přesnými řešeními, a řešení, která jsou přizpůsobena stabilním polohám Hirthova ozubení.		
$P_ORI_DIFF [<n>, <m>]	Tato proměnná poskytuje rozdíly mezi přesnými polohami orientačních os a polohami těchto os, které jsou k dispozici v proměnné $P_ORI_POS a které vyplývají z naprogramované orientace.		
	<n>:	Index řešení	
		Rozsah hodnot:	0, 1
	<m>:	Index orientační osy	
		Rozsah hodnot:	0 ... 2
			Posloupnost orientačních os (1 ... 3) se vztahuje na definici os v parametru $NT_ROT_AX_NAME.
	Obsah může být jiný než nulový jen tehdy, pokud jsou polohy indexovány (Hirthovo ozubení), tzn. jestliže se systémový parametr $NT_HIRTH_INCR příslušné osy nerovná nule a pokud se v případě této osy jedná o manuálně nastavovanou kruhovou osu.		

Systémové proměnné	Význam		
$P_ORI_SOL	Jestliže se v případě transformace orientace s více než jednou orientační osou vypočítávají úhly os, které mají vést k nastavení zadané orientace, obecně existuje více než jedno řešení. V systémové proměnné $P_ORI_SOL je obsažen počet platných řešení spolu s doplňkovými stavovými informacemi. Obsah proměnné $P_ORI_SOL je kódován následujícím způsobem:		
	Hodnoty < 0	Všeobecné chybové stavy	
		-1	Pro aktivní transformaci nebylo dosud vypočítáno žádné řešení (chybějící volání funkce ORISOLH).
		-2	Není aktivní žádná transformace nebo aktivní transformace není transformací orientace (6-osá transformace), která by mohla poskytnout pozice odpovídající požadované naprogramované orientaci.
		-4	Požadovaná orientace není prostřednictvím zadané kinematiky nastavitelná.
		-5	Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ nebylo nalezeno žádné řešení.
		-6	Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ je úhel γ příliš velký.
		-7	Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ byl zadán úhel, který prostřednictvím Hirthova ozubení není nastavitelný.
		-8	V parametrech je nastaveno, že první orientační osa (osa rámu) je Hirthova osa.
		-9	V parametrech je nastaveno, že jak druhá, tak i třetí kruhová osa je Hirthovou osou. Hirthovou osou smí být maximálně jedna z obou těchto os.
		-10	Nebylo nalezeno žádné přizpůsobení řešení Hirthovu ozubení.
	Hodnoty > 0 Místo jednotek	Počet matematicky možných řešení bez zohlednění mezních poloh os a případných chybových podmínek.	
		0	Neexistuje žádné řešení, tzn. požadovaná orientace není nastavitelná. Tento případ může mít tři různé příčiny: • Požadovaná orientace je i při libovolném rozsahu pohybů orientačních os principiálně nedosažitelná, a to v důsledku dané kinematiky stroje (orientační osy nejsou vůči sobě v pravých úhlech). V tomto případě jsou na místě desítek a stovek proměnné $P_ORI_SOL nuly, ve stavové proměnné $P_ORI_STAT, která je přiřazena orientačním osám, je hodnota „-4“. • Vypočítaná řešení nejsou dosažitelná, protože by znamenala narušení mezních hodnot os. Polohy orientačních os, které by vyplynuly, kdyby mezní hodnoty os neplatily, je možno načíst v proměnné $P_ORI_POS. • Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ byly zadány také polohy os, že buď vektor orientace nebo normálový vektor orientace nástroje jsou nasměrovány rovnoběžně s první orientační osou, jejíž

Systémové proměnné	Význam		
			poloha má být vypočítána. V těchto případech poloha osy není definována.
		1	Existuje jedno řešení. Tento případ může mít tři různé příčiny: • Na základě zadané orientace a kinematiky stroje, a také bez zohlednění mezních poloh os, existuje jen jedno řešení (matematicky viděno se jedná o dvě vzájemně se překrývající řešení). Tento případ se vyskytuje u nepravoúhlých kinematik na okrajích oblastí, v nichž lze provádět nastavování orientace. Proměnná $P_ORI_POS obsahuje obě identická řešení. • Existuje jen jedno řešení, protože druhé řešení je v důsledku narušení mezních hodnot polohy osy neplatné. Platné řešení je potom vždy prvním řešením v proměnné $P_ORI_POS. Druhé řešení, které by vyplynulo, pokud by nebyly zohledňovány mezní hodnoty polohy osy, je v případě potřeby možné načíst z proměnné $P_ORI_POS. • Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje" se jedná o normální případ. K předem zadaným polohám druhé orientační osy obecně existuje jen jedna platná poloha orientační osy, která ještě nebyla vypočítána.
		2	Existují dvě řešení.
		8	Existuje nekonečně mnoho řešení, tzn. poloha orientační osy (osa procházející pólem) je libovolná. Z obou možných poloh ostatních os (osy) je však jedna kvůli narušení mezních hodnot osy vyloučena.
		9	Existuje nekonečně mnoho řešení, tzn. poloha orientační osy (osa procházející pólem) není stanovena. Osu, jejíž poloha není stanovena, je možno zjistit pomocí místa stovek nebo ze systémové proměnné $P_ORI_STAT.
	Hodnoty > 0 Místo desítek	Bitově kódovaný údaj narušených mezních hodnot polohy osy. Přesnou příčinu chyby je možno zjistit ze systémové proměnné $P_ORI_STAT.	
		Bit 0 (hodnota 10):	Pro minimálně jedno řešení došlo k narušení minimálně jedné mezní hodnoty polohy 1. orientační osy.
		Bit 1 (hodnota 20):	Pro minimálně jedno řešení došlo k narušení minimálně jedné mezní hodnoty polohy 2. orientační osy.
		Bit 2 (hodnota 40):	Pro minimálně jedno řešení došlo k narušení minimálně jedné mezní hodnoty polohy 3. orientační osy.
	Hodnoty > 0 Místo stovek	Bitově kódovaný údaj pro nedefinované polohy os (může se vyskytnout tehdy, když existuje nekonečně mnoho řešení, tzn. když je na místě jednotek hodnota „9").	
		Bit 0 (hodnota 100):	Poloha 1. orientační osy není definována.

Systémové proměnné	Význam		
		Bit 1 (hodnota 200):	Poloha 2. orientační osy není definována.
		Bit 2 (hodnota 400):	Poloha 3. orientační osy není definována.
	Označení 1., 2. a 3. orientační osy se vztahuje na definici os v parametru $NT_ROT_AX_NAME.		

Systémové proměnné	Význam			
$P_ORI_STAT [<n>]	Po zpracování příkazu ORISOLH obsahuje stavové informace pro každou z maximálně 3 orientačních os.			
	<n>:	Index orientační osy (odpovídá indexu příslušné orientační osy v parametru $NT_ROT_AX_NAME)		
		Rozsah hodnot:	0 ... 2	
			Posloupnost orientačních os (1 ... 3) se vztahuje na definici os v parametru $NT_ROT_AX_NAME.	
	Obsah proměnné $P_ORI_STAT je kódován následujícím způsobem:			
	Hodnoty < 0	Všeobecné chybové stavy		
		-1	Stavové informace nejsou definovány (chybějící volání funkce ORISOLH).	
		-2	Není aktivní žádná transformace nebo aktivní transformace není transformací orientace (6-osá transformace), která by mohla poskytnout pozice odpovídající požadované naprogramované orientaci.	
		-3	Daná osa není v aktivní transformaci obsažena.	
		-4	Polohu osy není možné vypočítat, protože s uvedenou kinematikou je požadovaná orientace i v případě libovolně definovaného rozsahu pohybů osy nedosažitelná.	
		-5	Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ byly zadány také polohy os, že buď vektor orientace nebo normálový vektor orientace nástroje jsou nasměrovány rovnoběžně s první orientační osou, jejíž poloha má být vypočítána. V těchto případech poloha osy není definována.	
		-6	Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ je úhel γ příliš velký.	
		-7	Při volání funkce ORISOLH v režimu „Přímé nastavování orientace nástroje“ byl zadán úhel, který prostřednictvím Hirthova ozubení není nastavitelný.	
		-8	V parametrech je nastaveno, že první orientační osa (osa rámu) je Hirthova osa.	
		-9	V parametrech je nastaveno, že jak druhá, tak i třetí kruhová osa je Hirthovou osou. Hirthovou osou smí být maximálně jedna z obou těchto os.	
		-10	Nebylo nalezeno žádné přizpůsobení řešení Hirthovu ozubení.	
	Hodnoty > 0 Místo jednotek	Bitově kódovaný údaj narušených mezních hodnot polohy osy pro první řešení.		
		Bit 0 (hodnota 1):	První řešení narušuje dolní mezní hodnotu polohy osy.	
		Bit 1 (hodnota 2):	První řešení narušuje horní mezní hodnotu polohy osy.	
	Hodnoty > 0 Místo desítek	Bitově kódovaný údaj narušených mezních hodnot polohy osy pro druhé řešení.		
		Bit 0 (hodnota 10):	Druhé řešení narušuje dolní mezní hodnotu polohy osy.	
		Bit 1 (hodnota 20):	Druhé řešení narušuje horní mezní hodnotu polohy osy.	

Systémové proměnné	Význam		
	Hodnoty > 0	Údaj nedefinované polohy osy.	
	Místo stovek	Bit 0 (hodnota 100):	Poloha orientační osy její definována, tzn. požadovaná orientace je dosažena s jakýmkoli libovolným nastavením kruhové osy (osa prochází pólem). Tato informace je obsažena také v systémové proměnné $P_ORI_SOL.
	Pokud jde o čísla chyb, které signalizují narušení mezních hodnot polohy osy, může se jich vyskytnout i více najednou. Protože se při narušení mezní hodnoty polohy osy systém pokouší přičítáním nebo odečítáním násobků 360° dosáhnout polohy v rámci přípustných mezních polohy osy, není v případě, že je to možné, jednoznačně definováno, zda došlo k narušení dolní nebo horní mezní hodnoty polohy osy. Jestliže pro požadovanou orientaci neexistuje žádné Řešení ($P_ORI_SOL = 0), je stavu orientačních os podílejících se na transformaci přiřazena hodnota "0".		

Poznámka

$NT_ROT_AX_NAME

Pomocí této systémové proměnné je možno se odkazovat na maximálně 3 osy, které slouží pro nastavování orientace. Obsahuje názvy prvků řetězce ($NK_NAME), které definují ty strojní osy (kruhové osy), jež musí uskutečňovat pohyby potřebné pro nastavení orientace, které vyplývají z dané kinematické transformace. Posloupnost, v jaké jsou maximálně tři kruhové osy v této systémové proměnné obsaženy, nemá pro kinematiku stroje žádný význam, protože tato posloupnost je odvozena ze struktury kinematického řetězce. Protože jste ale definovali posloupnost, kterou pro přístup ke kruhovým osám používají jiné proměnné, musí se pořadí orientačních os v proměnné $NT_ROT_AX_NAME shodovat s popisem kinematiky.

Poznámka

Stavové informace

Stavové informace, které např. ukazují, že orientace je nedosažitelná nebo že je dosažitelná jen v případě narušení určitých mezních hodnot polohy osy, nezpůsobují alarm NC systému. Je odpovědností uživatele, aby na uváděné podmínky vhodným způsobem reagoval.

4.13.12.2 Aktivování modifikací korekčních parametrů u točivých nástrojů (CUTMOD, CUTMODK)

Modifikace korekčních parametrů u točivých nástrojů jsou v NC programu aktivovány prostřednictvím příkazů jazyka CUTMOD (ve spojení s orientovatelnými držáky nástrojů), příp. CUTMODK (pro transformace orientace, které byly definovány prostřednictvím kinematického řetězce).

Poznámka

Protože orientovatelný držák nástroje a transformace orientace, které byly definovány prostřednictvím kinematického řetězce, nemohou být aktivní současně, nedochází ke konfliktům mezi oběma variantami.

Syntaxe

```
CUTMOD = <hodnota>
```

nebo

```
CUTMODK = <příkaz>
```

Význam

CUTMOD:	Volání funkce ve spojení s orientovatelnými držáky nástrojů		
<hodnota>:	Přiřazovaná hodnota		
	Datový typ:	INT	
	Hodnota:	0	Funkce je deaktivována. Hodnoty poskytované systémovými proměnnými $P_AD... se rovnají odpovídajícím parametrům nástroje.
		> 0	Funkce se aktivuje, jestliže je orientovatelný držák nástroje s uvedeným číslem aktivní, tzn. aktivování je vázáno na určitý orientovatelný držák nástroje. Hodnoty poskytované systémovými proměnnými $P_AD... jsou oproti odpovídajícím parametrům nástroje v případě potřeby modifikovány v závislosti na aktivním otočení. Deaktivováním uvedeného orientovatelného držáku nástroje se funkce dočasně deaktivuje, aktivováním jiného orientovatelného držáku nástroje se deaktivuje trvale. V prvním případě se proto funkce při opětovném vyvolání stejného orientovatelného držáku nástroje znovu aktivuje, ve druhém případě je nové vyvolání nezbytné, a to i tehdy, když se orientovatelný držák nástroje s příslušným číslem znovu aktivuje později. Reset systému nemá na funkci žádný vliv.
		-1	Funkce je aktivována vždy, když je aktivní orientovatelný držák nástroje. Při výměně držáku nástroje nebo při jeho deaktivování a pozdějším opětovném aktivování nemusí být funkce CUTMOD znovu nastavována.
		-2	Funkce je aktivována vždy, když je aktivován orientovatelný držák nástroje, jehož číslo se rovná momentálně aktivnímu orientovatelnému držáku nástroje. Jestliže žádný orientovatelný držák není aktivní, má tato situace stejný význam, jako kdyby bylo nastaveno CUTMOD=0. Jestliže je aktivní orientovatelný držák, má tato situace stejný význam, jako kdyby bylo bezprostředně zadáno číslo aktuálního držáku nástroje.
		< -2	Hodnoty menší než 2 jsou ignorovány, tzn. v tomto případě je situace stejná, jako kdyby příkaz CUTMOD nebyl vůbec naprogramován. **Poznámka:** Tento interval hodnot by se vůbec neměl používat, protože je rezervován pro eventuální pozdější rozšíření.
CUTMODK:	Volání funkce ve spojení s transformacemi orientace, které byly definovány prostřednictvím kinematického řetězce.		

<příkaz>:	Přiřazený příkaz		
	Datový typ:	STRING	
	Hodnota:	"NEW"	Do paměti se ukládají stavy aktivní transformace definované prostřednictvím kinematického řetězce, které jsou důležité pro funkci „Modifikace korekčních parametrů“, název transformace a frame aktuální kontury. **Poznámka:** Tento příkaz je přípustný jen tehdy, pokud je aktivní vhodná transformace (TRAORI_DYN, TRAORI_STAT nebo TRAANG_K).
		"OFF"	Deaktivovaní aktivní funkce „Modifikace korekčních parametrů“. Data, která byla předtím uložena pomocí příkazu „NEW“, zůstávají zachována. **Poznámka:** Tento příkaz je přípustný jen tehdy, pokud není aktivní příkaz CUTMODK. Je-li tomu tak, nemá žádnou funkci. Případně existující datový blok pro funkci „Modifikace korekčních parametrů“ zůstává zachován.
		"ON"	Pomocí tohoto příkazu se znovu aktivuje funkce „Modifikace korekčních parametrů“ s datovým blokem, který byl předtím uložen pomocí příkazu „NEW“. Jestliže je při zpracovávání tohoto příkazu aktivní transformace s názvem uloženého datového bloku, funkce „Modifikace korekčních parametrů“ je okamžitě v platnosti. Jinak se aktivování zpozdí, dokud není aktivní transformace aktivována.
		"CLEAR"	Stejně jako příkaz „OFF“ deaktivuje funkci „Modifikace korekčních parametrů“ a kromě toho ještě vymaže uložený datový blok. **Poznámka:** Tento příkaz je přípustný jen tehdy, pokud není aktivní příkaz CUTMODK.

Poznámka

SD42984 $SC_CUTDIRMOD

Funkce, která může být aktivována příkazem CUTMOD, příp. CUTMODK, nahrazuje funkci, která může být aktivována pomocí nastavovaného parametru SD42984 $SC_CUTDIRMOD. Tato funkce však zůstává i nadále k dispozici nezměněna. Protože ale není rozumné využívat souběžně obě funkce, může být tato funkce aktivována jen tehdy, pokud je CUTMOD roven nule a CUTMODK se rovná nulovému řetězci.

Další informace

Načítání modifikovaných korekčních parametrů

Modifikované korekční parametry jsou k dispozici v následujících systémových proměnných a proměnných BTSS:

Význam	Systémová proměnná	Proměnná BTSS
Poloha břitu	$P_AD[2]	cuttEdgeParam2
Úhel držáku	$P_AD[10]	cuttEdgeParam10

Význam	Systémová proměnná	Proměnná BTSS
Směr břitu	$P_AD[11]	cuttEdgeParam11
Úhel volného řezání	$P_AD[24]	cuttEdgeParam24

Tato data jsou oproti odpovídajícím parametrům nástroje ($TC_DP2[..., ...], ...) upravována vždy, když byla pomocí příkazu `CUTMOD`, příp. `CUTMODK` aktivována funkce „Modifikace korekčních parametrů u točivých nástrojů" a když byl nástroj otočen prostřednictvím orientovatelného držáku nástroje, příp. pomocí vhodné transformace orientace.

Další systémové proměnné důležité pro tuto funkci

Systémová proměnná	Význam
$P_CUTMOD_ANG / $AC_CUTMOD_ANG	Obsahuje údaj úhlu, o který byl nástroj otočen v aktivní rovině obrábění a který u funkcí CUTMOD, příp. CUTMODK, představuje základ pro stanovení upravených parametrů břitu.
$P_CUTMOD / $AC_CUTMOD	Načítá momentální platnou hodnotu, která byla naposled naprogramována pomocí příkazu CUTMOD (číslo držáku nástroje, pro který má být aktivována modifikace korekčních parametrů). Jestliže byla naposled naprogramovaná hodnota pro funkci `CUTMOD=-2` (aktivování s momentálně aktivním orientovatelným držákem nástroje), v systémových proměnných není k dispozici hodnota „-2", ale číslo orientovatelného držáku nástroje, který byl v okamžiku naprogramování aktivní.
$P_CUTMODK / $AC_CUTMODK	Načítá název transformace, pod kterou byl založen momentálně platný datový blok pro „modifikaci korekčních parametrů".
$P_CUT_INV / $AC_CUT_INV	Pokud je nástroj otočený tak, že směr otáčení vřetena musí být invertován, je výslednou hodnotou TRUE. Za tím účelem musí být v bloku, na který se příslušná operace čtení vztahuje, splněny následující podmínky: 1. Je aktivní soustružnický nebo brusný nástroj (typy nástrojů 400 až 599 a / nebo SD42950 $SC_TOOL_LENGTH_TYPE = 2). 2. Modifikace korekčních parametrů byla aktivována příkazem `CUTMOD`, příp. `CUTMODK`. 3. Je aktivní orientovatelný držák nástroje, příp. transformace orientace definovaná pomocí kinematického řetězce, které byly zvoleny prostřednictvím příkazu CUTMOD, příp. CUTMODK. 4. Nástroj je s využitím orientovatelného držáku nástroje, příp. kinematické transformace orientace, otočen tak, aby výsledná normála břitu nástroje byla oproti výchozí poloze otočena o více než 90° (za obvyklých okolností 180°). Pokud alespoň jedna z výše uvedených čtyř podmínek není splněna, obsahuje proměnná hodnotu FALSE. Pro nástroje, jejichž poloha břitu není definována, obsahuje tato proměnná vždy hodnotu FALSE.

Systémová proměnná	Význam	
$P_CUTMOD_ERR	Chybový stav po posledním volání funkce CUTMOD. Funkce CUTMOD může být vyvolávána také implicitně při výměně nástroje. Proměnná se při resetu nastavuje zpět na nulu. Při každé výměně nástroje se napřed resetuje a potom se jí v případě potřeby nastavuje nová hodnota. Tato proměnná je bitově kódována. Bity mají následující význam:	
	Bit 0:	Pro aktivní nástroj není definován žádný platný směr břitu.
	Bit 1:	Oba úhly břitu (úhel volného řezání a úhel držáku) aktivního nástroje jsou nulové.
	Bit 2:	Úhel volného řezání aktivního nástroje má nepřípustnou hodnotu (< 0°nebo > 180°).
	Bit 3:	Úhel držáku aktivního nástroje má nepřípustnou hodnotu (< 0°nebo > 90°).
	Bit 4:	Úhel destičky aktivního nástroje má nepřípustnou hodnotu (< 0°nebo > 90°).
	Bit 5:	Kombinace polohy břitu - úhel držáku aktivního nástroje je nepřípustná (jestliže je poloha břitu 1 až 4, musí být úhel držáku ≤ 90°, jestliže je poloha břitu 5 až 8, musí být úhel ≥ 90°).
	Bit 6:	Nepřípustné otočení aktivního nástroje. Nástroj byl otočen z aktivní roviny obrábění o ± 90° (s tolerancí přibližně 1°). V důsledku toho poloha břitu v rovině obrábění již není definována.
	Bit 7:	Řezná destička se již nenachází v rovině obrábění a úhel mezi řeznou destičkou a rovinou obrábění překračuje horní mezní hodnotu definovanou pomocí nastavovaného parametru SD42998 $SC_CUTMOD_PLANE_TOL.
	Bit 8:	Řezná destička se nenachází v rovině obrábění. Absolutní hodnota úhlu α je větší než 1°. Úhel α je úhel otočení okolo souřadné osy, která je kolmá jak na osu otáčení, k níž je vztažen úhel β, tak také na osu otáčení, k níž je vztažen úhel γ (v případě G18 je to osa X).

$P_...: Proměnné předběžného zpracování

$AC_...: Proměnné hlavního zpracování

Všechny proměnné hlavního zpracování mohou být čteny v synchronních akcích. Přístup kvůli čtení z předběžného zpracování má za následek zastavení předběžného zpracování.

Změna roviny

Pro stanovení změněné polohy břitu, směru řezání a úhlu držáku, příp. úhlu volného řezání je určující sledování břitu v právě aktivní rovině (G17 - G19).

Jestliže však nastavovaný parametr SD42940 $SC_TOOL_LENGTH_CONST (změna složek délky nástroje v případě změny roviny) obsahuje platnou hodnotu nerovnající se nule (plus nebo mínus 17, 18 nebo 19), potom určuje její obsah rovinu, ve které jsou příslušné veličiny započítány.

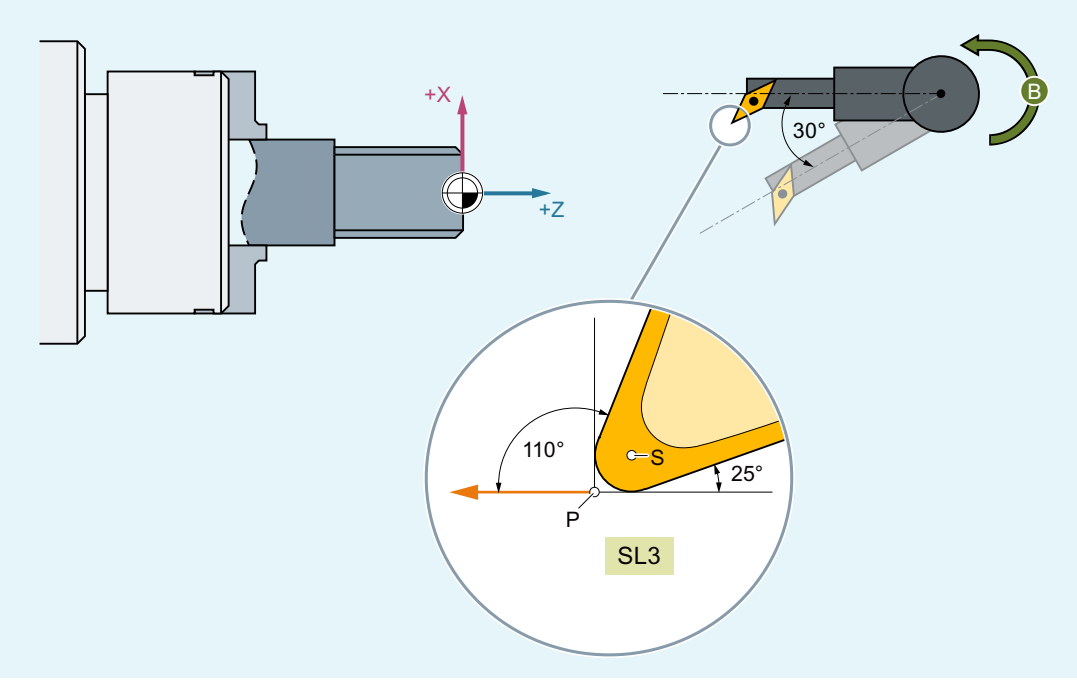
Toto pravidlo dané nastavovaným parametrem mající přednost před G-kódem může být zneplatněno nastavením bitu 18 strojního parametru $MC_TOOL_PARAMETER_DEF_MASK. To znamená, že pokud je tento bit nastaven, platí vždy rovina definovaná příkazem G-funkce ze skupiny 6.

Platnost změněných parametrů břitu

Změněná poloha břitu a změněný vztažný bod břitu jsou při programování okamžitě v platnosti i pro již aktivní nástroj. Nové aktivování nástroje není proto nutné.

Příklad

V případě nástroje s polohou břitu 3 a orientovatelného držáku nástroje, který může nástroj otáčet okolo osy B, má být pomocí příkazu CUTMOD poloha břitu po otočení nástroje modifikována.



S: Střed břitu

P: Vztažný bod břitu

SL: Poloha břitu

Programový kód	Komentář
N10 $TC_DP1[1,1]=500	
N20 $TC_DP2[1,1]=3	; Poloha břitu
N30 $TC_DP3[1,1]=12	
N40 $TC_DP4[1,1]=1	
N50 $TC_DP6[1,1]=6	
N60 $TC_DP10[1,1]=110	; úhel držáku
N70 $TC_DP11[1,1]=3	; směr břitu
N80 $TC_DP24[1,1]=25	; úhel volného řezání
N90 $TC_CARR7[2]=0 $TC_CARR8[2]=1 $TC_CARR9[2]=0	; osa B
N100 $TC_CARR10[2]=0 $TC_CARR11[2]=0 $TC_CARR12[2]=1	; osa C

Programový kód	**Komentář**
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	

Numerické hodnoty v komentářích uvádějí vždy pozici na konci bloku v souřadném systému stroje (MCS) v posloupnosti X → Y → Z.

Vysvětlení

V bloku N180 je napřed aktivován nástroj s nastavením CUTMOD=0 a neotočeným orientovatelným držákem nástroje. Protože všechny offsetové vektory orientovatelného držáku nástroje jsou 0, najede se na pozici, která odpovídá délkám nástroje uloženým v parametrech $TC_DP3[1,1] a $TC_DP4[1,1].

V bloku N200 se aktivuje orientovatelný držák nástroje s otočením o 30° okolo osy B. Protože poloha břitu kvůli nastavení CUTMOD=0 není změněna, je stejná jako před starým vztažným bodem břitu. Z tohoto důvodu se v bloku N210 najíždí na pozici, která v počátku (nule) obsahuje starý vztažný bod břitu (tzn. vektor (1, 12) je v rovině Z/X otočen o 30°).

V bloku N260 je na rozdíl od bloku N200 aktivní příkaz CUTMOD=2. V důsledku otočení orientovatelného držáku nástroje se stává platnou změněná poloha břitu 8. Z toho vyplývají také odlišné polohy os.

V blocích N220, příp. N270 je aktivována korekce rádiusu nástroje. Odlišné polohy břitu v obou úsecích programu nemají na koncovou pozici v blocích, v nichž je korekce rádiusu nástroje aktivována, žádný vliv, odpovídající polohy jsou proto identické. Teprve po blocích s deaktivováním N260, příp. N300 se odlišné polohy břitu opět projeví.

4.13.13 Práce s nástrojovým prostředím

Přehled funkcí

- Uložení nástrojového prostředí (TOOLENV) (Strana 813)
- Vymazání nástrojového prostředí (DELTOOLENV) (Strana 816)
- Načtení T-čísla, D-čísla a DL-čísla (GETTENV) (Strana 817)
- Načtení délek nástroje, příp. složek délky nástroje (GETTCOR) (Strana 818)
- Změna komponent nástroje (SETTCOR) (Strana 824)

Přehled systémových proměnných

- Načítání informací týkajících se uložených nástrojových prostředí ($P_TOOLENVN, ($P_TOOLENV) (Strana 818)

4.13.13.1 Uložení nástrojového prostředí (TOOLENV)

Funkce TOOLENV slouží pro uložení do paměti všech aktuálních stavů, které jsou důležité pro vyhodnocování parametrů nástroje uchovávaných v paměti.

Jedná se o následující data:

- Aktivní příkaz G-funkce ze skupiny:
  - 6 (G17, G18, G19)
  - 56 (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)
- Aktivní příčná osa
- Strojní parametry:
  - MD18112 $MN_MM_KIND_OF_SUMCORR (vlastnosti součtové korekce v oblasti TO)
  - MD20360 $MC_TOOL_PARAMETER_DEF_MASK (definice parametru nástroje)

- Nastavované parametry:
  - SD42900 $SC_MIRROR_TOOL_LENGTH (změna znaménka délky nástroj při zrcadlovém převrácení)
  - SD42910 $SC_MIRROR_TOOL_WEAR (změna znaménka opotřebení nástroje při zrcadlovém převrácení)
  - SD42920 $SC_WEAR_SIGN_CUTPOS (znaménko opotřebení u nástrojů se zařízeními pro nastavení břitu)
  - SD42930 $SC_WEAR_SIGN (znaménko opotřebení)
  - SD42935 $SC_WEAR_TRANSFORM (transformace pro komponenty nástroje)
  - SD42940 $SC_LENGTH_CONST (změna délkové složky rozměrů nástroje při změně roviny)
  - SD42942 $SC_TOOL_LENGTH_CONST_T (změna délkové složky rozměrů nástroje při změně roviny pro soustružnické nástroje)
  - SD42950 $SC_TOOL_LENGTH_TYPE (přiřazení složek délky nástroje nezávisle na typu nástroje)
  - SD42954 $SC_TOOL_ORI_CONST_M (výměna složek orientace pro frézovací nástroje pro změně roviny)
  - SD42956 $SC_TOOL_ORI_CONST_T (výměna složek orientace pro soustružnické nástroje pro změně roviny)
- Součást aktuálního celkového framu popisující orientaci (otočení a zrcadlové převrácení, žádná posunutí počátku nebo změny měřítka)
- Součást popisující orientaci a výsledná délka aktivního orientovatelného držáku nástroje
- Součást popisující orientaci a výsledná délka aktivní transformace

Kromě výše uvedených dat, která popisují nástrojové prostředí, jsou ukládány také T-číslo, D-číslo a DL-číslo aktivního nástroje, aby se později bylo možné dostat k tomuto nástroji ve stejném prostředí, jako při vyvolání příkazu TOOLENV, aniž by se nástroj musel znovu popisovat.

Syntaxe

```
<Status> = TOOLENV(<název>)
```

Význam

`TOOLENV(...)`:	Předem definovaná funkce pro uložení nástrojového prostředí	
	Samostatný blok:	ano

<Status>:	Výsledná hodnota funkce. Záporná hodnota signalizuje chybový stav.		
	Datový typ:		INT
	Hodnota:	0	Funkce je v pořádku
		-1	Pro nástrojové prostředí není v paměti rezervováno žádné místo: MD18116 $MN_MM_NUM_TOOL_ENV = 0 Tzn. funkce „Nástrojová prostředí“ není k dispozici.
		-2	Pro nástrojová prostředí už nejsou v paměti k dispozici žádná volná místa.
		-3	Nulový řetězec je jako název nástrojového prostředí nepřípustný.
		-4	Není zadán žádný parametr (<název>).
Parametry			
1 <název>:	Název, pod kterým má být aktuální datový blok uložen. Jestliže datový blok se stejným názvem už existuje, bude přepsán. Status je v tomto případě „0“.		
	Datový typ:		STRING

Další informace

Základní rozměr/rozměr adaptéru - korekce délky nástroje

To, zda do výpočtu délky nástroje vstupuje délka adaptéru nebo základní rozměr (parametr konkrétního břitu $TC_DP21, $TC_DP22 a $TC_DP23), rozhoduje v případě aktivní správy zásobníku nástrojů (k dispozici pouze spolu s volitelným doplňkem „Správa nástrojů) hodnota strojního parametru:

MD18104 $MN_MM_NUM_TOOL_ADAPTER (adaptér nástroje v oblasti TO)

Protože změna tohoto strojního parametru vstupuje v platnost pouze při spouštění řídícího systému, do nástrojového prostředí se neukládá.

Výsledná délka orientovatelných držáků nástroje a transformace

Poznámka

Jak v případě orientovatelných držáků nástroje, tak také u transformací existují systémové proměnné, příp. strojní parametry, které se chovají jako doplňkové složky délky nástroje a které jsou zcela nebo částečně podřízeny otočením, která se mají uskutečnit. Doplňkové složky délky nástroje, které takto vznikají, musí být při volání příkazu TOOLENV ukládány také, protože tvoří součást prostředí, ve kterém se nástroj používá.

Transformace adaptéru

Transformace adaptéru je vlastností adaptéru nástroje a tedy také celého nástroje. Není proto součástí nástrojového prostředí, které může být použito i na nějaký jiný nástroj.

Prostřednictvím uložení veškerých dat, která jsou zapotřebí pro stanovení celkové délky nástroje, je možné později vypočítat efektivní délku nástroje, a to i když v tomto okamžiku nástroj už není aktivní, příp. když se změnily okolní podmínky (např. příkazy G-funkcí nebo nastavované parametry). Stejně tak je možné vypočítat efektivní délku jiného nástroje za předpokladu, že by byl použit za stejných podmínek jako nástroj, pro který byly stavové parametry uloženy.

Maximální počet datových bloků pro nástrojová prostředí

Prostřednictvím strojního parametru MD18116 $MN_MM_NUM_TOOL_ENV se určuje, kolik datových bloků pro popis nástrojových prostředí může být maximálně uloženo. Tato data jsou uložena v oblasti TOA. Zůstávají zachována i po vypnutí řídícího systému.

Ukládání dat není možné. To znamená, že tato data není možné přenášet mezi různými řídícími systémy.

4.13.13.2 Vymazání nástrojového prostředí (DELTOOLENV)

Pomocí funkce DELTOOLENV můžete vymazat datové bloky pro popis nástrojového prostředí. Vymazání znamená, že k datovému bloku, který byl uložen pod určitým názvem, se již není možné dostat (pokus o přístup má za následek aktivování alarmu).

Poznámka

Datové bloky je možné vymazat pouze pomocí funkce DELTOOLENV, načtením souboru INITIAL.INI nebo prostřednictvím studeného startu (spuštění NC systému se standardními strojními parametry). Neexistují žádné další operace automatického mazání.

Syntaxe

```
<Status> = DELTOOLENV(<název>)
<Status> = DELTOOLENV()
```

Význam

DELTOOLENV(...):	Předem definovaná funkce pro vymazání nástrojového prostředí		
	Samostatný blok:		ano
<Status>:	Výsledná hodnota funkce. Záporná hodnota signalizuje chybový stav.		
	Datový typ:		INT
	Hodnota:	0	Funkce je v pořádku
		-1	Pro nástrojové prostředí není v paměti rezervováno žádné místo: MD18116 $MN_MM_NUM_TOOL_ENV = 0 Tzn. funkce „Nástrojová prostředí" není k dispozici.
		-2	Nástrojové prostředí se zadaným názvem už neexistuje.
Parametry			
1 <název>:	Název datového bloku, který má být vymazán		
	Datový typ:		STRING
DELTOOLENV():	Příkaz DELTOOLENV() bez udání názvu vymaže všechny datové bloky pro popis nástrojového prostředí		

4.13.13.3 Načtení T-čísla, D-čísla a DL-čísla (GETTENV)

Funkce GETTENV slouží k tomu, aby bylo možno načíst T-, D- a DL-čísla, která byla uložena v nástrojovém prostředí.

Syntaxe

```
<Status> = GETTENV(<název>, <TDDL>)
```

Význam

`GETTENV(...)`:	Předem definovaná funkce pro načítání T-čísel, D.-čísel a DL-čísel v datovém bloku pro popis nástrojového prostředí		
	Samostatný blok:	ano	
`<Status>`:	Výsledná hodnota funkce. Záporná hodnota signalizuje chybový stav.		
	Datový typ:	INT	
	Hodnota:	0	Funkce je v pořádku
		-1	Pro nástrojové prostředí není v paměti rezervováno žádné místo: MD18116 $MN_MM_NUM_TOOL_ENV = 0 Tzn. funkce „Nástrojová prostředí" není k dispozici.
		-2	Nástrojové prostředí se zadaným názvem už neexistuje.
Parametry			
1 `<název>`:	Název datového bloku, ze kterého se má T-číslo, D-číslo a DL-číslo načíst.		
	Datový typ:	STRING	
2 `<TDDL>`:	Pole tohoto výsledného parametru obsahují T-číslo, D-číslo a DL-číslo nástroje, jehož nástrojové prostředí je v uvedeném datovém bloku uloženo: • <TDDL> [0]: T-číslo • <TDDL> [1]: D-číslo • <TDDL> [2]: DL-číslo		
	Datový typ:	INT[3]	
`GETTENV(,<TDDL>)`, `GETTENV("",<TDDL>)`:	Při vyvolávání funkce GETTENV je přípustné první parametr vypustit nebo jako první parametr předávat nulový řetězec. V obou těchto zvláštních případech funkce v proměnné <TDDL> poskytuje T-číslo, D-číslo a DL-číslo **aktivního** nástroje.		

4.13.13.4 Načítání informací týkajících se uložených nástrojových prostředí ($P_TOOLENVN, ($P_TOOLENV)

Informace týkající se uložených nástrojových prostředí je možno načítat pomocí následujících systémových proměnných:

`$P_TOOLENVN`:	Poskytuje počet datových bloků pro popis nástrojových prostředí definovaných (a ještě nevymazaných) pomocí příkazu TOOLENV.			
	Syntaxe:	`<n> = $P_TOOLENVN`		
	Význam:	`<n>`:	Počet definovaných datových bloků	
			Datový typ:	INT
			Rozsah hodnot:	0 ... MD18116 $MN_MM_NUM_TOOL_ENV
	Přístup k této systémové proměnné je přípustný také tehdy, když žádná nástrojová prostředí nejsou možná (MD18116 = 0). V tomto případě funkce vrací hodnotu „0“.			
`$P_TOOLENV`:	Poskytuje název <i>-tého datového bloku pro popis nástrojového prostředí			
	Syntaxe:	`<název> = $P_TOOLENV[<i>]`		
	Význam:	`<název>`:	Název datového bloku s číslem <i>	
			Datový typ:	STRING
		`<i>`:	Číslo datového bloku	
			Datový typ:	INT
			Rozsah hodnot:	1 ... $P_TOOLENVN
	Přiřazování čísel datovým blokům není pevné, může se v důsledku mazání a zakládání nových datových bloků měnit. Číslování datových bloků se uskutečňuje interně. Pokud index <i> odkazuje na datový blok, který není definován, funkce vrací nulový řetězec. Jestliže je index <i> neplatný, tzn. <i> je menší než 1 nebo větší, než je maximální počet datových bloků pro nástrojová prostředí (MD18116 $MN_MM_NUM_TOOLENV), aktivuje se následující alarm: Alarm 17020 „Nepřípustný index1 pole“			

4.13.13.5 Načtení délek nástroje, příp. složek délky nástroje (GETTCOR)

Funkce GETTCOR slouží pro načítání délek, příp. složek délky nástroje.

Přitom může být prostřednictvím nastavení parametrů zadáno, které komponenty mají být zohledňovány, a při jakých podmínkách použití má být nástroj monitorován.

Syntaxe

```
<Status> = GETTCOR(<Len>[, <Comp>, <Stat>, <T>, <D>, <DL>])
```

Význam

`GETTCOR(...)`:	Předem definovaná funkce pro načítání délek nástroje, příp. složek délky nástroje	
	Samostatný blok:	ano

<Status>:	Výsledná hodnota funkce. Záporná hodnota signalizuje chybový stav.		
	Datový typ:		INT
	Hodnota:	0	Funkce je v pořádku
		-1	Pro nástrojové prostředí není v paměti rezervováno žádné místo: MD18116 $MN_MM_NUM_TOOL_ENV = 0 Tzn. funkce „Nástrojová prostředí" není k dispozici.
		-2	Nástrojové prostředí s názvem zadaným v parametru <Stat> neexistuje.
		-3	Neplatný řetězec v parametru <Comp>. Příčinou této chyby může být neplatný nebo dvojitě naprogramovaný znak.
		-4	Neplatné T-číslo
		-5	Neplatné D-číslo
		-6	Neplatné DL-číslo
		-7	Pokus o přístup do paměťového modulu, který není k dispozici.
		-8	Pokus o přístup k volitelnému doplňku, který není k dispozici (např. programovatelná orientace nástroje, správa nástrojů).
		-9	Řetězec <Comp> obsahuje dvojtečku (identifikace pro specifikace souřadného systému), nenásleduje ale žádné platné znaky pro popis požadovaného souřadného systému.
Parametry			
1 <Len>:	Výsledný vektor:		
	Datový typ:		REAL[11]
	Vektorové složky jsou uspořádány v následující posloupnosti: • <Len> [0]: Typ nástroje • <Len> [1]: Poloha břitu • <Len> [2]: Abscisa • <Len> [3]: Ordináta • <Len> [4]: Aplikáta • <Len> [5]: Rádius nástroje Jako vztažný souřadný systém pro komponenty délky platí souřadný systém definovaný v parametrech <Comp> a <Stat>. Pokud v parametru <Comp> žádný souřadný systém definován není, budou se délky nástroje zobrazovat v souřadném systému stroje. Přiřazení abscisy, ordináty a aplikáty ke geometrickým osám závisí na rovině, která je v použitém nástrojovém prostředí aktivní. Tzn. v případě G17 je abscisa rovnoběžná s osou X, v případě G18 s osou Z atd. Složky <Len>[6] až <Len>[10] obsahují doplňkové parametry, které mohou být zadávány za účelem popisu geometrie nástroje (např. $TC_DP7 až $TC_DP11 pro geometrii, příp. odpovídající komponenty pro opotřebení, příp. součtové a seřizovací korekce). Těchto 5 doplňkových prvků a rádius nástroje jsou definovány pouze pro složky E, G, S a W. Jejich vyhodnocování na parametru <Stat> nezávisí. Odpovídající hodnoty v parametrech <Len>[6] až <Len>[10] mohou být proto nenulové jen tehdy, pokud se na výpočtu délky nástroje podílí minimálně jedna z těchto čtyř jmenovaných složek. Zbývající složky nemají na výsledek žádný vliv. Údaje rozměrů jsou vztaženy na základní systém řídícího systému (palce nebo metrické jednotky).		

2	<Comp>:	Složky délky nástroje (volitelné)			
		Datový typ:	STRING		
		Řetězec znaků skládající se ze dvou dílčích řetězců, které jsou od sebe odděleny dvojtečkou. Obecný formát: "<SubStr_1> [: <SubStr_2]"			
		<SubStr_1>:	**První dílčí řetězec** označuje složky délky nástroje, které mají být zohledňovány při výpočtu délky nástroje. Posloupnost znaků v dílčím řetězci, stejně jako způsob zápisu (malá nebo velká písmena), jsou libovolné. Mezi znaky může být vložen libovolný počet mezer nebo znaky tabulátoru (netisknutelné znaky). **Upozornění:** Znaky v dílčím řetězci **nesmí být naprogramovány dvakrát**!		
			Znaky:	–	Znak mínus (přípustný jen jako první znak) Bude vypočítána celková délka nástroje mínus složky, které jsou specifikovány v následujícím řetězci.
				C	Rozměr adaptéru nebo základní rozměr (použije se ta z obou alternativně existujících složek, která je pro používaný nástroj aktivní)
				E	Korekční parametry pro seřizování
				G	Geometrie
				K	Kinematická transformace (vyhodnocuje se pouze v případě generické transformace se 3, 4 a 5 osami!)
				S	Součtové korekce
				T	Orientovatelný držák nástroje
				W	Opotřebení
			Pokud je první dílčí řetězec prázdný (s výjimkou mezer), znamená to, že má být vypočítána celková délka nástroje při zohlednění všech složek. Totéž platí i tehdy, když parametr <Comp> není vůbec uveden.		
		<Substr_2>:	Nepovinný **druhý dílčí řetězec** označuje souřadný systém, ve kterém má být délka nástroje udána. Druhý dílčí řetězec se skládá z jednoho jediného relevantního znaku.		
			Znaky:	A	Nastavitelný souřadný systém (ACS)
				B	Základní souřadný systém (BCS)
				K	Souřadný systém nástroje kinematické transformace (KCS)
				M	Souřadný systém stroje (MCS)
				T	Souřadný systém nástroje (TCS)
				W	Souřadný systém obrobku (WCS)
			Jestliže není uveden žádný souřadný systém, uskutečňuje se vyhodnocování v souřadném systému stroje (MKS). Otočení, která je v případě potřeby nutno započítat, jsou stanovena prostřednictvím nástrojového prostředí definovaného v parametru <Stat>.		

3	<_Stat>:	Název datového bloku pro popis nástrojového prostředí (volitelné)	
		Datový typ:	STRING
		Jestliže se v tomto parametru nachází nulový řetězec („“) nebo pokud není udán, použije se aktuální stav. Pokud není specifikován žádný nástroj, použije se ten aktuální.	
4	<T>:	Interní T-číslo nástroje (nepovinné)	
		Datový typ:	INT
		Jestliže tento parametr není uveden nebo pokud je jeho hodnota „0“, použije se nástroj určený v parametru <Stat>. Je-li hodnota tohoto parametru „-1“, použije se T-číslo aktivního nástroje. Je rovněž přípustné číslo aktivního nástroje explicitně zadat. **Upozornění:** Pokud parametr <Stat> není uveden, jako nástrojové prostředí se použije momentální stav. Protože zadání <T> = 0 znamená odkaz na T-číslo uložené v nástrojovém prostředí, použije se v tomto případě aktivní nástroj, což znamená, že zadání <T> = 0 a <T> = -1 mají za těchto speciálních okolností stejný význam.	
5	<D>:	Číslo břitu nástroje (nepovinné)	
		Datový typ:	INT
		Jestliže tento parametr není uveden nebo pokud je jeho hodnota „0“, řídí se použité D-číslo podle zdroje T-čísla. Pokud se používá T-číslo z nástrojového prostředí, načte se z tohoto nástrojového prostředí také D-číslo, jinak se použije momentálně aktivní nástroj.	
6	<DL>:	Číslo korekce závislé na místě (nepovinné)	
		Datový typ:	INT
		Jestliže tento parametr není uveden, řídí se použité DL-číslo podle zdroje T-čísla. Pokud se používá T-číslo z nástrojového prostředí, načte se z tohoto nástrojového prostředí také D-číslo, jinak se použije momentálně aktivní nástroj.	

Příklady

`GETTCOR(_LEN)`	Vypočítá se délka momentálně aktivního nástroje v souřadném systému stroje, přičemž budou zohledněny všechny složky.
`GETTCOR(_LEN,"CGW:W")`	Vypočítá se délka nástroje skládající se z rozměru adaptéru, příp. základního rozměru, geometrie a opotřebení pro aktivní nástroj. Další složky, jako např. orientovatelný držák nástroje nebo kinematická transformace, nejsou zohledňovány. Výsledek je uveden v souřadném systému obrobku.
`GETTCOR (_LEN,"-K:B")`	Vypočítává se celková délka aktivního nástroje bez zohlednění délkových složek eventuálně aktivní kinematické transformace. Výstup je v základním souřadném systému.
`GETTCOR (_LEN,":M","Testenv1",,3)`	Vypočítává se celková délka nástroje pro nástroj uložený v nástrojovém prostředí s názvem „Testenv1“ v souřadném systému stroje. Výpočet se ale uskutečňuje nezávisle na uloženém čísle břitu pro břit D3.

Další informace

Transformace adaptéru / orientovatelný držák nástroje / kinematická transformace

Otočení a záměny složek, které v případě potřeby vyplývají z transformace adaptéru, orientovatelného držáku nástroje a kinematické transformace, jsou součástí nástrojového prostředí. Z tohoto důvodu jsou vždy započítávány, a to i tehdy, když odpovídající složky délky nemají být brány v úvahu. Pokud to není požadováno, musí být definována nástrojová prostředí, v nichž odpovídající transformace nejsou aktivní. V mnoha případech (jmenovitě vždy, když se na stroji nepoužívají žádné transformace, příp. žádný orientovatelný držák nástroje) splňují ukládané datové bloky pro nástrojová prostředí tyto podmínky automaticky, takže uživatel na ně nemusí brát žádný zvláštní zřetel.

Soustružnické a brusné nástroje: Výpočet délky nástroje v závislosti na parametru MD20360 $MC_TOOL_PARAMETER_DEF_MASK

To, jak je u soustružnických a brusných nástrojů zapotřebí vyhodnocovat opotřebení, příp. délku nástroje ve směru eventuálně existující osy, ve které je průměr nástroje, je definováno ve strojním parametru:

MD20360 $MC_TOOL_PARAMETER_DEF_MASK (definice parametru nástroje)

Bit	Hodnota	
0	U soustružnických a brusných nástrojů se **parametr opotřebení** v příčné ose započítává jako hodnota průměru.	
	= 0 (předdefinovaná hodnota)	ne
	= 1	ano
1	U soustružnických a brusných nástrojů se **délková složka rozměrů nástroje** v příčné ose započítává jako hodnota průměru.	
	= 0 (předdefinovaná hodnota)	ne
	= 1	ano

Jestliže jsou příslušné bity nastaveny, vyhodnocuje se odpovídající údaj s faktorem 0,5. Toto vyhodnocení se projevuje také v délce nástroje poskytované funkcí GETTCOR.

Příklad:

MD20360 $MC_TOOL_PARAMETER_DEF_MASK = 3

MD20100 $MC_DIAMETER_AX_DEF (geometrická osa s funkcí příčné osy) = "X"

X je osa, ve které je průměr (standardní konfigurace soustruhu)

```
Programový kód                          Komentář
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                  ; opotřebení L1
N90 $TC_DP13[1,1]=13.0
```

```
Programový kód                          Komentář
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
```

Složky délky v rámci kinematické transformace a orientovatelného držáku nástroje

Pokud je v případě výpočtu délky nástroje zohledňován **orientovatelný držák nástroje**, vstupují do výpočtu délky nástroje následující vektory:

Typ	Vektory
M	l1 a l2
T	l1, l2 a l3
P	Délka nástroje není prostřednictvím orientovatelného držáku nástroje ovlivňována.

V případě generické **5-osé transformace** vstupují u typů transformace 24 a 56 do výpočtu délky nástroje následující strojní parametry:

Typ transformace	Strojní parametry
24	MD24550/24650 $MC_TRAFO5_BASE_TOOL_1/2 MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2 MD24558/24658 $MC_TRAFO5_PART_OFFSET_1/2
56	MD24550/24650 $MC_TRAFO5_BASE_TOOL_1/2 MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2

Typ transformace 56 (pohyblivý nástroj a pohyblivý obrobek) odpovídá typu M v případě orientovatelného držáku nástroje.

U této transformace s 5 osami odpovídá v dosavadních verzích programového vybavení vektor MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2 (vektor kinematického offsetu 1./2. transformaci s 5 osami v kanálu) součtu obou vektorů l_1 a l_3 u orientovatelného držáku nástroje typu M.

Pro transformaci je v obou případech důležitý pouze součet. Způsob, jakým se skládá z obou jednotlivých komponent, nemá žádný význam. Pří výpočtu délky nástroje je ale důležité, která složka je přiřazena nástroji a která nástrojovému stolu. Z tohoto důvodu byl zaveden strojní parametr MD24558/24658 $MC_TRAFO5_JOINT_OFFSET_PART_1/2 (vektor kinematického offsetu ve stole). Odpovídá vektoru l3. Strojí parametr MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2 nyní již neodpovídá součtu vektorů l1 a l3, nýbrž pouze vektoru l1. Jestliže je strojní parametr MD24558/24658 $MC_TRAFO5_JOINT_OFFSET_PART_1/2 roven nule, je toto chování zcela shodné s dřívějším stavem.

Kompatibilita

Funkce GETTCOR slouží spolu s funkcemi TOOLENV a SETTCOR mimo jiné k tomu, aby nahradily část funkcí, které byly až dosud realizovány externě v rámci měřicích cyklů.

V měřicích cyklech byla vyhodnocována pouze část parametrů, které nakonec určovaly efektivní délku nástroje. Pro uváděné funkce mohou být parametry nastaveny tak, že chování měřicích cyklů, pokud jde o výpočet délky nástroje, může být reprodukováno.

4.13.13.6 Změna komponent nástroje (SETTCOR)

Funkce SETTCOR slouží k tomu, aby bylo možné směnit složky nástroje při zohledňování všech okrajových podmínek, které se na vyhodnocování jednotlivých složek podílejí.

Poznámka

Poznámka k terminologii: Jestliže je v následujícím textu v souvislosti s délkou nástroje řeč o složkách nástroje, jsou míněny složky vektorové povahy, ze kterých se celková délka nástroje skládá (např. geometrie nebo opotřebení). Taková složka se proto skládá ze tří jednotlivých hodnot (L1, L2, L3), které jsou v následujícím textu označovány jako hodnoty souřadnic.

Složka nástroje označovaná jako „geometrie" se tak například skládá ze tří hodnot souřadnic $TC_DP3 až $TC_DP5.

Syntaxe

```
<Status> = SETTCOR(<CorVal>, <Comp>, [<CorComp>, <CorMode>, <GeoAx>,
<Stat>, <T>, <D>, <DL>])
```

Význam

`SETTCOR(...)`:	Předem definovaná funkce pro změnu složek nástroje	
	Samostatný blok:	ano

<Status>:	Výsledná hodnota funkce. Záporná hodnota signalizuje chybový stav.		
	Datový typ:		INT
	Hodnota:	0	Funkce je v pořádku
		-1	Pro nástrojové prostředí není v paměti rezervováno žádné místo: MD18116 $MN_MM_NUM_TOOL_ENV = 0 Tzn. funkce „Nástrojová prostředí" není k dispozici.
		-2	Nástrojové prostředí s názvem zadaným v parametru <Stat> neexistuje.
		-3	Neplatný řetězec v parametru <Comp>. Příčinou této chyby může být neplatný nebo dvojitě naprogramovaný znak.
		-4	Neplatné T-číslo
		-5	Neplatné D-číslo
		-6	Neplatné DL-číslo
		-7	Pokus o přístup do paměťového modulu, který není k dispozici.
		-8	Pokus o přístup k volitelnému doplňku, který není k dispozici (např. programovatelná orientace nástroje, správa nástrojů).
		-9	Nepřípustná číselná hodnota pro parametr <CorComp**>**.
		-10	Nepřípustná číselná hodnota pro parametr <CorMode>.
		-11	Údaje obsažené v parametrech <Comp> a <CorComp> si vzájemně odporují.
		-12	Údaje obsažené v parametrech <Comp> a <CorMode> si vzájemně odporují.
		-13	Obsah parametru <GeoAx> neoznačuje žádnou geometrickou osu.
		-14	Pokus o zápis do seřizovacího korekčního parametru, který není k dispozici.
Parametry			
1	<CorVal>:	Vektor korekce V souřadném systému obrobku (WCS) definovaném prostřednictvím parametru <Stat> platí následující přiřazení: • <CorVal> [0]: Abscisa • <CorVal> [1]: Ordináta • <CorVal> [2]: Aplikáta Jestliže má být korekce uskutečněna jen v jedné složce nástroje (tzn. žádná vektorová korekce, viz parametr <CorMode>), nachází se hodnota korekce vždy v <CorVal>[0] bez ohledu na to, na kterou osu je uplatňována. Obsah obou zbývajících složek se potom nevyhodnocuje. Pokud se vztahuje hodnota v parametru <CorVal>, příp. jedna ze složek parametru <CorVal>, na příčnou osu, je tento údaj vyhodnocován jako **rozměr rádiusu**. To znamená, že pokud je nástroj např. o uvedený rozměr „delší", způsobí to dvakrát tak velkou změnu průměru obrobku. Údaje rozměrů jsou vztaženy na **základní systém** řídícího systému (palce nebo metrické jednotky).	
		Datový typ:	REAL[3]

2	<Comp>:	Složky nástroje			
		Datový typ:	STRING		
		Řetězec znaků skládající se ze dvou dílčích řetězců, které jsou od sebe odděleny dvojtečkou. Obecný formát: "<SubStr_1> [: <SubStr_2]"			
		<SubStr_1>:	**První dílčí řetězec** musí být vždy k dispozici a může se skládat z jednoho nebo ze dvou znaků. První nebo jediný znak přitom reprezentuje 1. složku (Val_1) a druhý znak 2. složku (Val_2), které jsou zpracovávány v závislosti na následujících parametrech <CorComp> a <CorMode>.		
			Znaky:	C	Rozměr adaptéru nebo základní rozměr (použije se ta z obou alternativně existujících složek, která je pro používaný nástroj aktivní)
				E	Korekční parametry pro seřizování
				G	Geometrie
				S	Součtové korekce
				W	Opotřebení
		<Substr_2>:	**Druhý dílčí řetězec** je nepovinný. Může se skládat z (jediného) písmena „W“ nebo „T“.		
			Znaky:	W	Pokud je druhý dílčí řetězec prázdný nebo pokud obsahuje písmeno „W“, jsou hodnoty korekcí započítávány tak, jako kdyby byly změřeny v souřadném systému **obrobku** (WKS).
				T	Pokud obsahuje druhý dílčí řetězec písmeno „T“, jsou hodnoty korekcí započítávány tak, jako kdyby byly změřeny v souřadném systému **nástroje** (**T**ool Coordinate System, TCS).
		Způsob zápisu (malá nebo velká písmena) je libovolný. Může být vloženo libovolně mnoho prázdných znaků nebo znaků tabulátoru (netisknutelné znaky).			

3	<CorComp>:	Specifikuje složky datového bloku nástroje, které mají být popsány (nepovinné).		
		Datový typ:		INT
		Hodnota:	0	Hodnota korekce <CorVal>[0] se vztahuje na geometrickou osu v souřadném systému obrobku nebo v souřadném systému nástroje předávanou v parametru <GeoAx> (bližší informace viz popis parametru <Comp>). To znamená, že hodnota korekce musí být ve stanovených složkách nástroje započítána tak, že jako výsledek vznikne změna celkové délky nástroje v definovaném směru osy o předdefinovanou hodnotu, přičemž jsou zohledněny všechny parametry, které mohou výpočet délky nástroje ovlivňovat. Tato změna má být dosažena prostřednictvím korekce složky zadané v parametru <Comp> a symbolického matematického předpisu zadaného v parametru <CorMode> (viz následující parametr). Výsledná korekce se proto může uplatňovat ve složkách všech tří os.
			1	Jako „0“, ale vektorově. Obsah vektoru <CorVal> se vztahuje na abscisu, ordinátu a aplikátu v souřadném systému obrobku nebo v souřadném systému nástroje předávanou v parametru <GeoAx> (bližší informace viz popis parametru <Comp>). Následující parametr <GeoAx> se nevyhodnocuje.
			2	Vektorovou korekci, tzn. L1, L2 a L3, je možno upravovat simultánně. Na rozdíl od variant „0“ a „1“ se ale hodnoty korekcí obsažené v parametru <CorVal> vztahují na souřadnice složky Val_1 (viz následující parametr <CorMode>) nástroje. Šikmá poloha nástroje vůči souřadnému systému obrobku, která se případně může vyskytnout, nemá na korekce žádný vliv.
			3 - 5	Korekce délky nástroje L1 až L3 ($TC_DP3 až $TC_DP5), příp. odpovídající hodnoty u opotřebení, seřizovacích korekcí nebo součtových korekcí. Hodnota korekce je obsažena v parametru <CorVal>[0]. Je změřena v souřadnicích složky Val_1 (viz následující parametr <CorMode>) nástroje. Šikmá poloha nástroje vůči souřadnému systému obrobku, která se případně může vyskytnout, nemá na korekce žádný vliv.
			6	Korekce rádiusu nástroje ($TC_DP6), příp. odpovídající hodnoty u opotřebení, seřizovacích korekcí nebo součtových korekcí. Bity 10 a 11 (údaje průměru, příp. opotřebení ve směru průměru, jsou vyhodnocovány volitelně jako údaje rádiusu nebo průměru) ve strojním parametru MD20360 $MC_TOOL_PARAMETER_DEF_MASK jsou zohledňovány.
			7 – 11	Korekce $TC_DP7 až $TC_DP11, příp. odpovídající hodnoty u opotřebení, seřizovacích korekcí nebo součtových korekcí. S těmito parametry se zachází stejně jako s rádiusem nástroje.
		Jestliže tento parametr není uveden, je jeho hodnota „0“.		

<table>
<tr><td>4</td><td><CorMode>:</td><td colspan="3">Specifikuje druh operace zápisu, kterou je potřeba uskutečnit (nepovinné)</td></tr>
<tr><td></td><td></td><td colspan="2">Datový typ:</td><td>INT</td></tr>
<tr><td></td><td></td><td rowspan="4">Hodnota:</td><td>0</td><td>$Val_{1nová}$ = <CorVal></td></tr>
<tr><td></td><td></td><td>1</td><td>$Val_{1nová}$ = $Val_{1stará}$ + <CorVal></td></tr>
<tr><td></td><td></td><td>2</td><td>$Val_{1nová}$ = <CorVal>
$Val_{2nová}$ = 0</td></tr>
<tr><td></td><td></td><td>3</td><td>$Val_{1nová}$ = $Val_{1stará}$ + $Val_{2stará}$ + <CorVal>
$Val_{2nová}$ = 0</td></tr>
<tr><td></td><td></td><td colspan="3">Způsobu zápisu $Val_{1stará}$ + $Val_{2stará}$ je potřeba rozumět symbolicky. Pokud jsou obě složky (kvůli stavu parametru <_Stat>) vyhodnocovány odlišně, tzn. mezi oběma složkami se uplatňuje otočení, je hodnota parametru $Val_{2stará}$ před přičtením transformována tak, aby výsledná délka nástroje po vymazání hodnoty $Val_{2nová}$ a po přičtení parametru <CorVal> zůstala nezměněna.
<CorVal> se vždy vztahuje na parametr Val_1. <CorVal> je hodnota, která se v závislosti na druhé části parametru <Comp> měří buď v souřadném systému obrobku (WCS), nebo v souřadném systému nástroje (TCS). Z tohoto důvodu je na rozdíl od složek nástroje, ve kterých se má započítávat, v případě potřeby již podrobena transformaci. Není tedy možné počítat bezprostředně s uloženými hodnotami, nýbrž před přičítáním k hodnotám Val_1, příp. Val_2 musí být provedena zpětná transformace. To může mít za následek, že se korekce uplatňuje v jiné ose, než jaká je definována prostřednictvím parametru <CorComp>, příp. že se korekce uplatňuje ve více osách.
V případě, že se <CorComp> = 0, tzn. jestliže parametr <CorVal> neobsahuje vektor, ale jednotlivou hodnotu, jsou popisované operace uskutečňovány v souřadnicích, ve kterých byla hodnota <CorVal> změřena (WCS/TCS). To platí zejména také pro dosazení nuly pro parametr $Val_{2nový}$ ve variantách 2 a 3. Tento výsledek se potom zpětně transformuje do souřadnic nástroje. To může mít za následek, že žádná z vynulovaných souřadnic (L1, L2, L3) nebude nulová, příp. že hodnoty souřadnic, které byly nulové, už nulové nejsou. Jestliže jsou ale odpovídající operace uskutečněny postupně pro všechny tři geometrické osy, je zaručeno, že také všechny tři hodnoty souřadnic mazaných složek jsou nulové. Jestliže nástroj není oproti souřadnému systému obrobku otočený, nebo pokud je otočený, ale tak, že všechny složky nástroje zůstaly rovnoběžně se souřadnými osami (záměna os), je zaručeno, že se také změní jen jedna souřadnice nástroje.
Postupné uskutečnění stejné operace (<CorMode>) s parametrem <CorComp> = 0 pro všechny tři osy souřadného systému v libovolné posloupnosti má identický výsledek jako jednorázové provedení stejné operace s nastavením <CorComp>=1.
Pro hodnoty parametru „0“ a „1“ musí parametr <Comp> obsahovat jeden a pro hodnoty parametru „2“ a „3“ dva znaky.
Příklad:
<Comp> obsahuje řetězec „ES“, <CorMode> hodnotu „2“.
⇒ Seřizovací korekce$_{nová}$ = <CorVal>, součtová korekce$_{nová}$ = 0
Jestliže parametr <CorMode> není uveden, je jeho hodnota „0“.</td></tr>
<tr><td>5</td><td><GeoAx>:</td><td colspan="3">Udává index geometrické osy, ve které byla změřena hodnota korekce <CorVal>[0] (nepovinné).</td></tr>
<tr><td></td><td></td><td colspan="2">Datový typ:</td><td>INT</td></tr>
<tr><td></td><td></td><td colspan="2">Rozsah hodnot:</td><td>0 ... 2</td></tr>
<tr><td></td><td></td><td colspan="3">Indexy 0 až 2 se vztahují na abscisu, ordinátu a aplikátu v platné rovině (G17/G18/G19) momentálního nástrojového prostředí.
Obsah tohoto parametru se vyhodnocuje jen tehdy, pokud má parametr <CorComp> hodnotu „0“.</td></tr>
</table>

6	<Stat>:	Název datového bloku pro popis nástrojového prostředí (volitelné)	
		Datový typ:	STRING
		Jestliže se v tomto parametru nachází nulový řetězec („“) nebo pokud není udán, použije se aktuální stav. Pokud není specifikován žádný nástroj, použije se ten aktuální.	
7	<T>:	Interní T-číslo nástroje (nepovinné)	
		Datový typ:	INT
		Jestliže tento parametr není uveden nebo pokud je jeho hodnota „0“, použije se nástroj určený v parametru <Stat>. Je-li hodnota tohoto parametru „-1“, použije se T-číslo aktivního nástroje. Je rovněž přípustné číslo aktivního nástroje explicitně zadat. **Upozornění:** Pokud parametr <Stat> není uveden, jako nástrojové prostředí se použije momentální stav. Protože zadání <T> = 0 znamená odkaz na T-číslo uložené v nástrojovém prostředí, použije se v tomto případě aktivní nástroj, což znamená, že zadání <T> = 0 a <T> = -1 mají za těchto speciálních okolností stejný význam.	
8	<D>:	Číslo břitu nástroje (nepovinné)	
		Datový typ:	INT
		Jestliže tento parametr není uveden nebo pokud je jeho hodnota „0“, řídí se použité D-číslo podle zdroje T-čísla. Pokud se používá T-číslo z nástrojového prostředí, načte se z tohoto nástrojového prostředí také D-číslo, jinak se použije momentálně aktivní nástroj.	
9	<TL>:	Číslo korekce závislé na místě (nepovinné)	
		Datový typ:	INT
		Jestliže tento parametr není uveden, řídí se použité DL-číslo podle zdroje T-čísla. Pokud se používá T-číslo z nástrojového prostředí, načte se z tohoto nástrojového prostředí také D-číslo, jinak se použije momentálně aktivní nástroj. Jestliže parametry T, D a DL specifikují nástroj bez korekcí závislých na místě, nesmí být v parametru <Comp> specifikovány žádné součtové nebo seřizovací korekce (chybový kód v parametru <Status>).	

Poznámka

Negace všechny možné kombinace trojice parametrů <Comp>, <CorComp> a <CorMode> mají smysl. Například matematický předpis 3 v parametru <CorComp> vyžaduje v parametru <Comp> zadání dvou znaků. Jestliže je zadána nepřípustná kombinace parametrů, v parametru <Status> se objeví příslušný chybový kód.

Příklady

Příklad 1

```
Programový kód                          Komentář
N10 DEF REAL _CORVAL[3]
N20 $TC_DP1[1,1]=120                    ; frézovací nástroj
N30 $TC_DP3[1,1]=10.0                   ; geometrie L1
N40 $TC_DP12[1,1]=1.0                   ; opotřebení L1
N50 _CORVAL[0]=0.333
```

Programový kód	**Komentář**
N60 T1 D1 G17 G0	
N70 R1=**SETTCOR(_CORVAL,"G",0,0,2)**	
N80 T1 D1 X0 Y0 Z0	; ==> poloha v MCS X0.000 Y0.000 **Z1.333**
N90 M30	

<CorComp> má hodnotu „0“, proto je hodnota souřadnice geometrické složky, která se uplatňuje ve směru Z, nahrazena hodnotou korekce 0,333.

Výsledná celková délka nástroje tedy je: L1 = 0,333 + 1,000 = 1,333

Příklad 2

Programový kód	**Komentář**
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	; frézovací nástroj
N30 $TC_DP3[1,1]=10.0	; geometrie L1
N40 $TC_DP12[1,1]=1.0	; opotřebení 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	; ==> poloha v MCS X0.000 Y0.000 Z11.333
N90 M30	

<CorComp> má hodnotu „1“, proto je hodnota korekce 0,333, která se uplatňuje ve směru Z, přičtena k hodnotě opotřebení 1,0.

Výsledná celková délka nástroje tedy je: L1 = 10,0 + 1,333 = 11,333

Příklad 3

Programový kód	**Komentář**
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	; frézovací nástroj
N30 $TC_DP3[1,1]=10.0	; geometrie L1
N40 $TC_DP12[1,1]=1.0	; opotřebení 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	; ==> poloha v MCS X0.000 Y0.000 Z0.333
N90 M30	

<CorComp> má hodnotu „2“, z tohoto důvodu se hodnota korekce, která se uplatňuje ve směru osy Z, přenáší do složky geometrie (stará hodnota se přepisuje) a hodnota opotřebení se vymaže.

Výsledná celková délka nástroje tedy je: L1 = 0,333 + 0,0 = 0,333

Příklad 4

Programový kód	**Komentář**
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	; frézovací nástroj

Programový kód	**Komentář**
N30 $TC_DP3[1,1]=10.0	; geometrie L1
N40 $TC_DP12[1,1]=1.0	; opotřebení 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	; ==> poloha v MCS X0.000 Y0.000 Z11.333
N90 M30	

<CorComp> má hodnotu „3“, proto se hodnota opotřebení a hodnota korekce přičítají ke složkám geometrie a složky opotřebení se vymažou.

Výsledná celková délka nástroje tedy je: L1 = 11,333 + 0,0 = 11,333.

Příklad 5

Programový kód	**Komentář**
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	; frézovací nástroj
N30 $TC_DP3[1,1]=10.0	; geometrie L1
N40 $TC_DP12[1,1]=1.0	; opotřebení 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	; ==> poloha v MCS X0.333 Y0.000 Z11.000
N90 M30	

<CorComp> má hodnotu „3“ jako v předcházejícím příkladu, korekce se ale nyní uplatňuje na geometrickou osu s indexem „0“ (osa X), které je kvůli G17 u frézovacího nástroje přiřazena složka nástroje L3. Vyvolání příkazu SETTCOR z tohoto důvodu parametry nástroje $TC_DP3 a $TC_DP12 neovlivňuje. Hodnota korekce se místo toho ukládá do parametru $TC_DP5.

Příklad 6

Programový kód	**Komentář**
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; soustružnický nástroj
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	; opotřebení L1
N60 $TC_DP13[1,1]=0.0	; opotřebení 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	; ==> poloha v MCS X24.330 Y0.000 Z17.500
N120 M30	

Nástrojem je soustružnický nůž. V bloku N80 je aktivováno otočení framu, takže základní souřadný systém (BCS) je vůči souřadnému systému obrobku (WCS) pootočený. Hodnota

korekce (N70) se ve WCS uplatňuje na geometrickou osu s indexem „1“, tzn., protože je aktivní G18, na osu X. Protože se parametr <CorMode> = 3, platí, že opotřebení nástroje ve směru osy X ve WCS musí být po zpracování řádku N100 rovno nule.

Obsah příslušných parametrů nástroje na konci programu proto je:

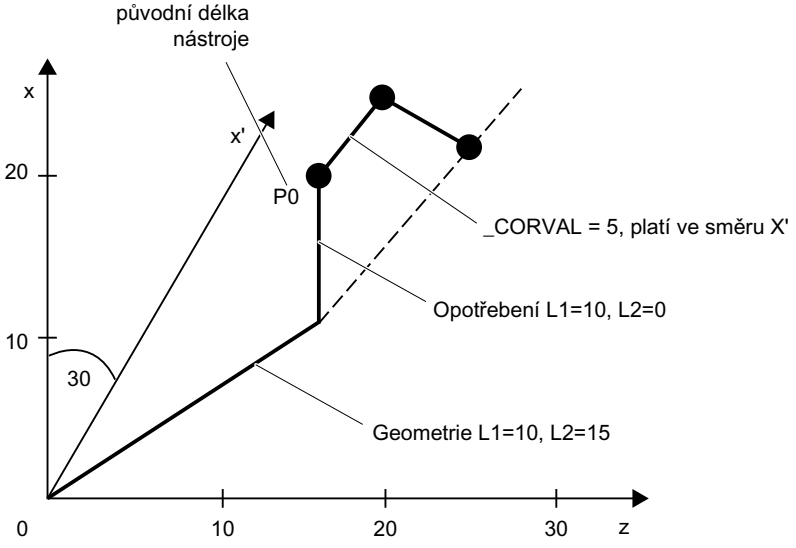
$TC_DP3[1,1] : 21.830 ; geometrie L1

$TC_DP4[1,1] : 21.830 ; geometrie L2

$TC_DP12[1,1] : 2.500 ; opotřebení L1

$TC_DP13[1,1] : -4.330 ; opotřebení L2

Tyto geometrické poměry zachycuje následující obrázek. Celkové opotřebení včetně hodnoty _CORVAL se promítá ve směru osy X' ve WCS. Z toho vyplývá bod P2. Souřadnice tohoto bodu (změřené v souřadnicích X-Y) jsou uloženy v geometrických složkách nástroje. V opotřebení zbývá rozdílový vektor P_2 - P_1. V důsledku toho již nemá opotřebení žádnou složku ve směru _CORVAL.



Pokud by program v příkladu pokračoval po bloku N110 následujícími příkazy, přeneslo by se zbývající opotřebení úplně do geometrie, protože korekce se nyní uplatňuje ve směru osy Z' (parametr <GeoAx> = 0):

```
N120 _CORVAL[0]=0.0
N130 R1=SETTCOR(_CORVAL,"GW",0,3,0)
N140 T1 D1 X0 Y0 Z0 ; ==> poloha v MCS X24.330 Y0.000 Z17.500
```

Protože nová hodnota korekce je „0“, nesmí se celková délka nástroje a tím pádem také pozice, na kterou se najíždí v bloku N140, změnit. Pokud by se hodnota _CORVAL v bloku N120 nerovnala „0“, vznikla by v bloku N140 nová celková délka nástroje a proto také změněná poloha, složka opotřebení délky nástroje by však byla v každém případě nulová, tzn. celková délka nástroje je potom v každém případě obsažena ve složce geometrie nástroje.

Stejný výsledek jako při dvojnásobném vyvolání funkce SETTCOR s nastavením parametru <CorComp> = 0 se dosáhne také prostřednictvím jediného vyvolání s nastavením <CorComp> = 1 (vektorová korekce):

Programový kód	**Komentář**
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; soustružnický nástroj
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	; opotřebení L1
N60 $TC_DP13[1,1]=0.0	; opotřebení 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	; ==> poloha v MCS X24.330 Y0.000 Z17.500
N120 M30	

V tomto případě jsou všechny složky opotřebení nástroje okamžitě po prvním vyvolání funkce SETTCOR v bloku N100 rovny nule.

Příklad 7

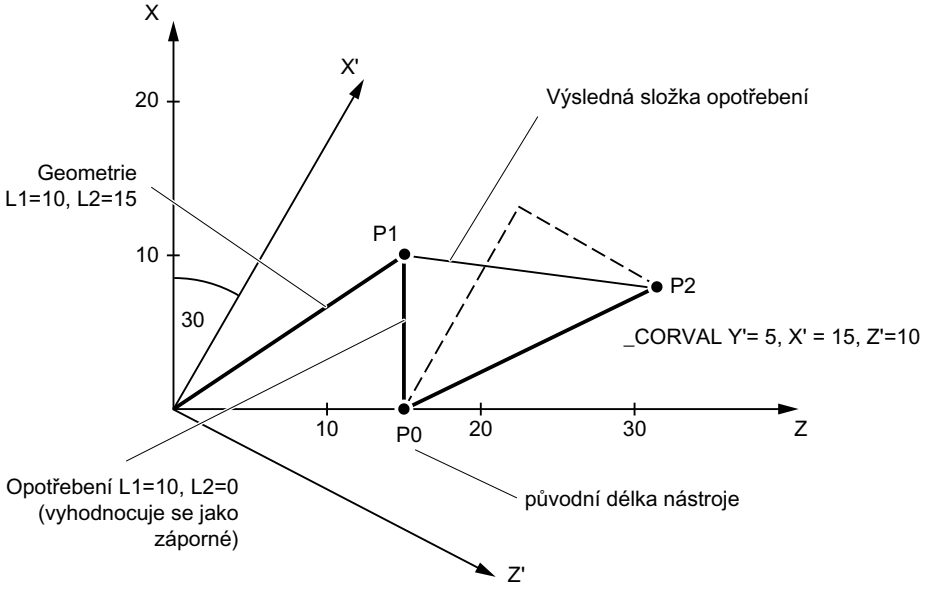
Programový kód	**Komentář**
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; soustružnický nástroj
N30 $TC_DP3[1,1]=10.0	; geometrie L1
N40 $TC_DP4[1,1]=15.0	; geometrie L2
N50 $TC_DP12[1,1]=10.0	; opotřebení L1
N60 $TC_DP13[1,1]=0.0	; opotřebení L2
N70 _CORVAL[0]=5.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=**SETTCOR(_CORVAL,"GW",3,3)**	
N110 T1 D1 X0 Y0 Z0	; ==> poloha v MCS X25.000 Y0.000 Z15.000
N120 M30	

Oproti příkladu 6 je v tomto případě parametr <CorComp> = 3, proto může údaj parametru <GeoAx> odpadnout. Hodnota obsažená v parametru _CORVAL[0] se nyní uplatňuje bezprostředně na složku nástroje L1, otočení v bloku N80 nemá na výsledek žádný vliv, složky opotřebení v $TC_DP12 se spolu s hodnotou v parametru _CORVAL[0] přenášejí do složky geometrie, takže kvůli $TC_DP13 se celková délka nástroje již po prvním vyvolání funkce SETTCOR v bloku N100 nachází ve složce geometrie nástroje.

Příklad 8

Programový kód	**Komentář**
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; soustružnický nástroj
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	; opotřebení L1
N70 $TC_DP13[1,1]=0.0	; opotřebení L2
N80 $TC_DP14[1,1]=0.0	; opotřebení 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	; ==> poloha v MCS X7.990 Y25.000 Z31.160
N170 M30	

V bloku N90 je přiřazována hodnota nastavovanému parametru SD42930 $SC_WEAR_SIGN, tzn. opotřebení musí být vyhodnocováno se záporným znaménkem. Korekce je vektorová (<CorComp> = 1) a vektor korekce musí být přičten k opotřebení (<CorMode> = 1). Tyto geometrické poměry v rovině Z-X ukazuje následující obrázek:



Kvůli nastavení <CorMode> = 1 zůstává složka geometrie nástroje nezměněna. Vektor korekce definovaný ve WCS (otočení okolo osy Y) musí být převzat do složky opotřebení tak, aby celková délka nástroje v obrázku 3 odpovídala bodu P_2. Výsledná složka opotřebení nástroje je proto dána vzdáleností bodů P_1 a P_2.

Protože je ale hodnota opotřebení kvůli nastavovanému parametru SD42930 $SC_WEAR_SIGN vyhodnocována jako záporná, musí být takto zjištěná korekce přenesena do paměti korekčních parametrů se záporným znaménkem. Obsah příslušných parametrů nástroje na konci programu proto je:

$TC_DP3[1,1] : 10.000 ; geometrie L1 (nezměněna)

$TC_DP4[1,1] : 15.000 ; geometrie L2 (nezměněna)

$TC_DP5[1,1] : 10.000 ; geometrie L3 (nezměněna)

$TC_DP12[1,1] : 2.010 ; opotřebení L1 (= 10 - 15 * cos(30) + 10 * sin(30))

$TC_DP13[1,1] : -16.160 ; opotřebení L2 (= -15 * sin(30) - 10 * cos(30))

$TC_DP14[1,1] : -5.000 ; opotřebení L3

Jako složka L3 ve směru osy Y se dá vliv nastavovaného parametru SD42930 $SC_WEAR_SIGN bez dalších komplikací rozpoznat prostřednictvím otočení framu.

Další informace

Soustružnické / brusné nástroje: Výpočet délky nástroje v závislosti na parametru MD20360 $MC_TOOL_PARAMETER_DEF_MASK

To, jak je u soustružnických / brusných nástrojů zapotřebí vyhodnocovat opotřebení, příp. délku nástroje ve směru eventuálně existující osy, ve které je průměr nástroje, je definováno ve strojním parametru:

MD20360 $MC_TOOL_PARAMETER_DEF_MASK.<Bit> = <hodnota>

<Bit>	<hodnota>	Význam
0	0	U soustružnických a/ brusných nástrojů se **parametr opotřebení** v příčné ose započítává jako **hodnota rádiusu**:
	1	U soustružnických / brusných nástrojů se **parametr opotřebení** v příčné ose započítává jako **hodnota průměru**:
1	0	U soustružnických / brusných nástrojů se **délková složka rozměrů nástroje** v příčné ose započítává jako **hodnota rádiusu**:
	1	U soustružnických / brusných nástrojů se **délková složka rozměrů nástroje** v příčné ose započítává jako **hodnota průměru**:

Jestliže jsou příslušné bity nastaveny, vyhodnocuje se odpovídající údaj s faktorem 0,5. Korekce je prostřednictvím funkce SETTCOR realizována tak, že se celková efektivní změna délky nástroje rovné přenášené hodnotě v proměnné <CorVal>. Pokud je délka na základě strojního parametru MD20360 $MC_TOOL_PARAMETER_DEF_MASK při výpočtu vyhodnocována s faktorem 0,5, musí se proto korekce této složky uskutečňovat s dvojnásobkem přenášené hodnoty.

Příklad

MD20360 $MC_TOOL_PARAMETER_DEF_MASK = 2 (délka nástroje musí být v ose, ve které se měří průměr, vyhodnocována s faktorem 0,5)

Osa X je osou průměru.

Programový kód	Komentář
N10 DEF REAL _LEN[11]	

Programový kód	**Komentář**
N20 DEF REAL _CORVAL[3]	
N30 $TC_DP1[1,1]=500	; Typ nástroje
N40 $TC_DP2[1,1]=2	; Poloha břitu
N50 $TC_DP3[1,1]=3.	; Geometrie – délka 1
N60 $TC_DP4[1,1]=4.	; Geometrie – délka 2
N70 $TC_DP5[1,1]=5.	; Geometrie – délka 3
N80 _CORVAL[0]=1.	
N90 _CORVAL[1]=1.	
N100 _CORVAL[2]=1.	
N110 T1 D1 G18 G0 X0 Y0 Z0	; ==> poloha v MCS X1.5 Y5 Z4
N120 R1=SETTCOR(_CORVAL,"G",1,1)	
N130 T1 D1 X0 Y0 Z0	; ==> poloha v MCS X2.5 Y6 Z5
N140 R3=$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	

Korekce délky nástroje by v každé ose měla být 1 mm (N80 až N100). V délkových parametrech L2 a L3 se proto k původní délce přičte vždy 1 mm. K původní délce nástroje v parametru L1 se oproti tomu přičítá dvojnásobek hodnoty korekce (2 mm), aby se celková délka nástroje prodloužila o 1 mm, jak je požadováno. Při porovnání pozic, na které se najíždí v blocích N110 a N130 je možno zjistit, že se všechny pozice os změnily o 1 mm.

4.13.14 Načítání přiřazení délek nástroje L1, L2, L3 osám souřadného systému (LENTOAX)

Funkce LENTOAX poskytuje informace o přiřazení délek nástroje L1, L2 a L3 **aktivního** nástroje abscise, ordinátě a aplikátě. Přiřazení abscisy, ordináty a aplikáty geometrickým osám je ovlivňováno framy a aktivní rovinou (G17 - G19).

Uvažovány jsou přitom pouze složky geometrie nástroje ($TC_DP3[<t>,<d>] až $TC_DP5[<t>,<d>]), tzn. přiřazení dalších složek osám, které se v případě potřeby od tohoto liší (např. opotřebení), nemá na výsledek žádný vliv.

Syntaxe

```
<Status> = LENTOAX(<AxInd>, <Matrix>[, <Coord>])
```

Princip

LENTOAX(...):	Předem definovaná funkce pro načítání přiřazení délek nástroje L1, L2 a L3 aktivního nástroje osám souřadného systému.	
	Samostatný blok:	ano

<Status>:	Výsledná hodnota funkce. Záporná hodnota signalizuje chybový stav.		
	Datový typ:		INT
	Hodnota:	0	Funkce je v pořádku Informace v parametru <AxInd> jsou pro popis postačující (všechny složky délky nástroje jsou rovnoběžné s geometrickými osami).
		1	Funkce je v pořádku, aby byl ale popis správný, musí být vyhodnocován obsah parametru <Matrix> (složky délky nástroje nejsou rovnoběžné s geometrickými osami).
		-1	Neplatný řetězec v parametru <Coord>.
		-2	Není aktivní žádný nástroj.
Parametry			
1 <AxInd>:	Pokud jsou složky délky nástroje rovnoběžné s geometrickými osami, jsou v poli <AxInd> funkcí poskytovány indexy os, které jsou přiřazeny délkovým složkám L1 až L3. • <AxInd> [0]: Abscisa • <AxInd> [1]: Ordináta • <AxInd> [2]: Aplikáta		
	Datový typ:		INT[3]
	Hodnota:	0	Žádné přiřazení není k dispozici (osa neexistuje)
		1 ... 3 nebo -1 ... -3	Číslo délky, které se uplatňuje v odpovídající souřadné ose. Pokud složka délky nástroje ukazuje v záporném směru příslušné souřadnice, je znaménko záporné.
	Jestliže ne všechny složky délky jsou rovnoběžné, příp. antiparalelní, s geometrickými osami, bude parametr <AxInd> poskytovat příslušný index osy, která obsahuje největší podíl složky délky nástroje. V tomto případě (jestliže funkce z nějakého jiného důvodu nehlásí chybu), je výsledná hodnota <Status> = 1. Zobrazení složek délky nástroje L1 až L3 do geometrických os 1 až 3 je potom úplně popsáno obsahem 2. parametru <Matrix>.		
2 <Matrix>:	Matice, která provádí zobrazení vektoru délek nástroje (L1=1, L2=1, L3=1) do vektoru souřadných os (abscisa, ordináta, aplikáta), tzn. sloupcům jsou přiřazeny složky délky nástroje v posloupnosti L1, L2, L3 a řádkům osy v posloupnosti abscisa, ordináta a aplikáta.		
	Datový typ:		REAL
	V parametru Matrix jsou vždy v platnosti všechny prvky, a to i tehdy, když geometrická osa patřící k určité ose souřadného systému vůbec neexistuje, tzn. když je odpovídající položka v parametru <AxInd> nulová.		

3	<Coord>:	Souřadný systém, pro který přiřazení platí (nepovinné)		
		Datový typ:		STRING
		Znaky:	MCS M	Zobrazení délky nástroje v souřadném systému stroje
			BCS B	Zobrazení délky nástroje v základním souřadném systému
			WCS W	Zobrazení délky nástroje v souřadném systému obrobku (standardní nastavení)
			KCS K	Zobrazení délky nástroje v souřadném systému nástroje kinematické transformace
			TCS T	Zobrazení délky nástroje v souřadném systému nástroje
		Způsob zápisu (malá nebo velká písmena) je libovolný. Jestliže parametr <Coord> není uveden, použije se WCS (předdefinované nastavení).		

Poznámka

V TCS jsou všechny složky délky nástroje vždy paralelní nebo antiparalelní s osami.

Antiparalelní mohou být složky jen tehdy, jestliže je aktivní zrcadlové převrácení a je-li aktivován následující nastavovaný parametr:

SD42900 $SC_MIRROR_TOOL_LENGTH (změna znaménka délky nástroj při zrcadlovém převrácení)

Příklad

Standardní případ frézovacího nástroje v případě G17.

L1 se uplatňuje v ose Z (aplikáta), L2 se uplatňuje v ose Y (ordináta), L3 se uplatňuje v ose X (abscisa).

Vyvolání funkce ve formě:
```
<Status>=LENTOAX(<AxInd>,<Matrix>,"WCS")
```

Výsledný parametr <AxInd> potom obsahuje hodnoty:

<AxInd>[0] = 3

<AxInd>[1] = 2

<AxInd>[2] = 1

logické zkráceně: (3, 2, 1)

Odpovídající matice v tomto případě je:

$$\text{<Matrix>} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Přepnutím z G17 do G18 nebo G19 se výsledek nezmění, protože přiřazení složek délky ke geometrickým osám se změní stejným způsobem jako přiřazení abscisy, ordináty a aplikáty.

Když je aktivní funkce G17, naprogramuje se nyní otočení framu okolo osy Z o 60°, např.:
`ROT Z60`

Směr aplikáty (směr osy Z) zůstává nezměněn, hlavní složka L2 nyní leží ve směru nové osy X, hlavní složka L1 se nachází v záporném směru osy Y. Výsledná hodnota (<Status>) je proto „1“, <AxInd> obsahuje hodnoty (2, -3, 1).

Odpovídající matice v tomto případě je:

$$\text{<Matrix>} = \begin{pmatrix} 0 & \sin 60° & \cos 60° \\ 0 & \cos 60° & -\sin 60° \\ 1 & 0 & 0 \end{pmatrix}$$

4.14 Chování při pohybu po dráze

4.14.1 Průběh charakteristiky posuvu (FNORM, FLIN, FCUB, FPO)

Pro účely pružnějšího zadávání průběhu posuvu bylo jeho programování podle normy DIN 66025 rozšířeno o lineární a kubické průběhy.

Kubické průběhy mohou být naprogramovány buď přímo nebo jako interpolační spliny. Tak je možné naprogramovat, v závislosti na zakřivení opracovávaného obrobku, spojité a hladké funkční průběhy rychlosti.

Tyto funkční průběhy rychlosti umožňují změny zrychlení bez trhavých pohybů a tím tedy i výrobu hladkých povrchů obrobku.

Syntaxe

```
F... FNORM
F... FLIN
F... FCUB
F=FPO(...,...,...)
```

Význam

`FNORM`	**Normální** posuv podle DIN 66025 (základní nastavení) Hodnota posuvu se zadává prostřednictvím dráhového úseku daného bloku a potom platí jako modální hodnota.
`FLIN`	Profil rychlosti pohybu po dráze je **lineární** Hodnota posuvu se z hodnoty aktuální na začátku bloku do hodnoty na konci bloku po celém úseku dráhy lineárně mění a potom platí jako modální hodnota. Průběh hodnoty posuvu FLIN se uplatňuje při příkazech G93, G94 a G95, **nikoli** u příkazů G96/G961 a G97/G971. V případě příkazu G95 a při změněných otáčkách vřetena prostřednictvím korekce (override) nebo synchronní akce je možné, že se synchronní akce trochu zpozdí.

`FCUB`	Profil rychlosti pohybu po dráze je **kubický** Blokově naprogramované hodnoty F, které jsou vztaženy na konec bloku, jsou spojovány spliny. Spline začíná a končí s tangenciálním napojením na předešlý, resp. následující údaj posuvu. Pokud v nějakém bloku F-adresa chybí, použije se pro tento účel naposled naprogramované F-slovo. Průběh hodnoty posuvu FCUB se uplatňuje při příkazech G93 a G94, **nikoli** u příkazů G95, G96/G961 a G97/G971.
`F=FPO(...,...,...)`	Profil rychlosti pohybu po dráze je zadán **pomocí polynomu** F-adresa popisuje průběh posuvu od aktuální hodnoty do hodnoty na konci bloku pomocí polynomu. Koncová hodnota pak platí jako modální.

Nezávisle na naprogramovaném průběhu posuvu platí funkce pro programování chování při pohybu po dráze.

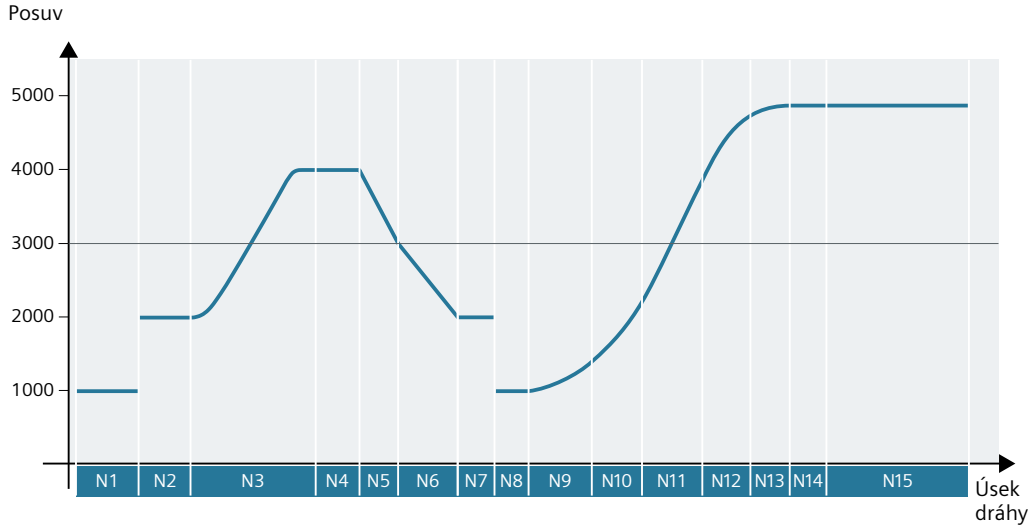
Naprogramovaný průběh posuvu má v zásadě absolutní platnost, nezávisle na příkazech G90 nebo G91.

Poznámka

Optimalizace posuvu v případě zakřivených úseků dráhy

Pro pohyby s posuvem daným polynomem F=FPO() a splinem FCUB by vždy měla být nastavena konstantní hodnota posuvu na kontuře CFC. Tímto způsobem je možné vytvořit profil požadované hodnoty posuvu se spojitým zrychlením.

Příklad: Různé profily hodnoty posuvu



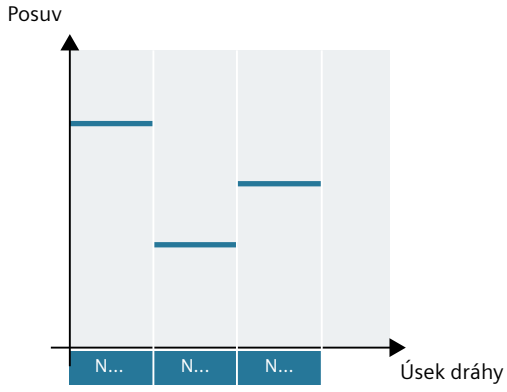
Programový kód	**Komentář**
`N1 F1000 FNORM G1 X8 G91 G64`	`; konstantní profil hodnoty posuvu, zadávání inkrementálních rozměrů`
`N2 F2000 X7`	`; skoková změna požadované hodnoty rychlosti`

Programový kód	Komentář
N3 F=FPO(4000, 6000, -4000)	; profil hodnoty posuvu zadaný pomocí polynomu s hodnotou posuvu 4000 na konci bloku
N4 X6	; posuv 4000 daný polynomem platí jako modální hodnota
N5 F3000 FLIN X5	; lineární profil posuvu
N6 F2000 X8	; lineární profil posuvu
N7 X5	; lineární posuv platí jako modální hodnota
N8 F1000 FNORM X5	; konstantní profil hodnoty posuvu se skokovou změnou zrychlení
N9 F1400 FCUB X8	; všechny následující blokově naprogramované hodnoty F budou pospojovány spliny
N10 F2200 X6	
N11 F3900 X7	
N12 F4600 X7	
N13 F4900 X5	; deaktivování splinového profilu
N14 FNORM X5	
N15 X20	

Další informace

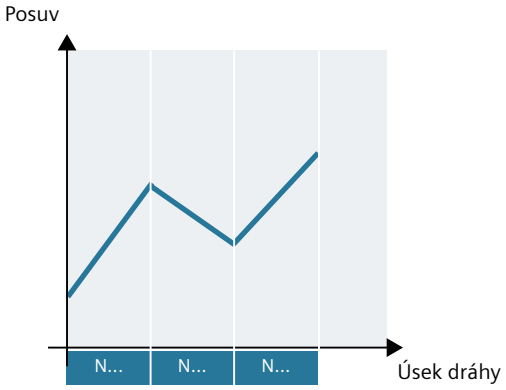
FNORM

Podle normy DIN 66025 označuje adresa posuvu F posuv po dráze jako konstantní hodnotu.



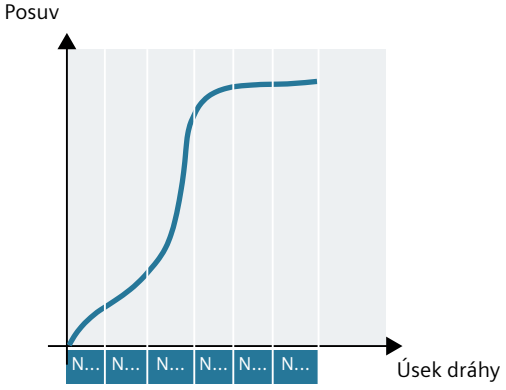
FLIN

Hodnota posuvu se bude lineárně měnit od momentální hodnoty posuvu až do naprogramovaného F-slova, přičemž této hodnoty bude dosaženo na konci bloku.



FCUB

Hodnota posuvu se bude měnit od momentální hodnoty posuvu až do naprogramovaného F-slova podle kubické funkce, přičemž této hodnoty bude dosaženo na konci bloku. Řídící systém pomocí splinových funkcí pospojuje všechny hodnoty posuvu naprogramované pomocí příkazu FCUB s blokovou platností. Hodnoty posuvu přitom slouží jako uzlové body pro výpočet splinové interpolace.



F=FPO(... , ... , ...)

Průběh hodnoty posuvu je přímo naprogramován pomocí polynomu. Zadávání koeficientů polynomu se uskutečňuje podobně jako při polynomické interpolaci.

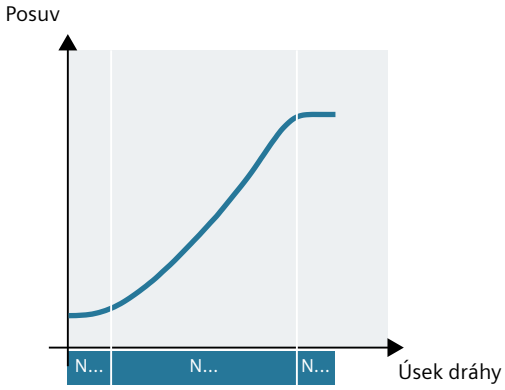
Příklad:

```
F=FPO(endfeed,quadf,cubf)
```

Parametry endfeed, quadf a cubf jsou dříve definované proměnné.

endfeed	Posuv na konci bloku
quadf	Kvadratický koeficient polynomu
cubf	Kubický koeficient polynomu

Když je aktivní příkaz FCUB, napojují se spliny na začátku bloku a na konci bloku tangenciálně na průběh zadaný pomocí funkce FPO.



Chování při aktivní funkci kompresoru

Když je aktivní funkce kompresoru (COMP...), platí při spojování většího počtu bloků do jednoho splinového úseku následující:

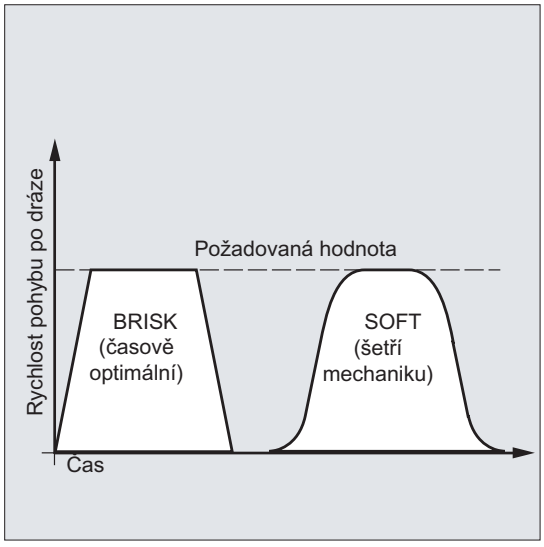
FNORM:	Pro splinový segment platí F-slovo z posledního bloku, který do tohoto segmentu patří.
FLIN:	Pro splinový segment platí F-slovo z posledního bloku, který do tohoto segmentu patří. Naprogramovaná hodnota F platí na konci segmentu a bude jí dosaženo pomocí lineární funkce.
FCUB:	Vytvořený spline posuvu se od naprogramované hodnoty v koncovém bodu odchyluje maximálně o hodnotu definovanou ve strojním parametru MD20172 $MC_COMPRESS_VELO_TOL.
F=FPO(..., ..., ...):	Tyto bloky nejsou nijak komprimovány.

4.14.2 Chování zrychlení:

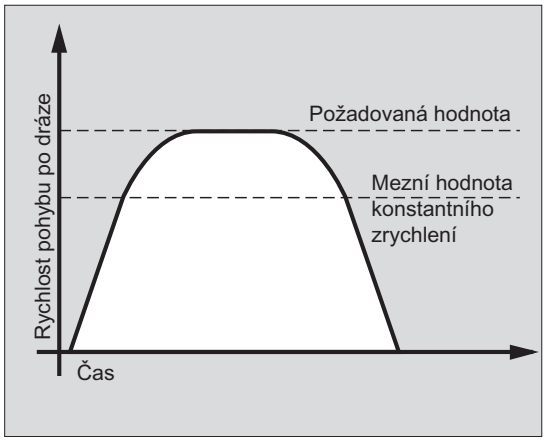
4.14.2.1 Způsoby chování při změnách zrychlení (BRISK, BRISKA, SOFT, SOFTA, DRIVE, DRIVEA)

Pro programování způsobu chování při změnách zrychlení jsou k dispozici následující příkazy výrobního programu:

- "BRISK, BRISKA"
  Jednotlivé osy, příp. dráhové osy se pohybují s maximálním zrychlením až do dosažení naprogramované rychlosti posuvu (**zrychlení bez omezení ryvu**).
- "SOFT, SOFTA"
  Jednotlivé osy, příp. dráhové osy se pohybují se spojitým zrychlením až do dosažení naprogramované rychlosti posuvu (**zrychlení s omezením ryvu**).
- "DRIVE, DRIVEA"
  Jednotlivé osy, příp. dráhové osy se pohybují s maximálním zrychlením až do dosažení mezní hodnoty rychlosti nastavené v konfiguraci (nastavení pomocí MD!). Potom dochází ke snížení zrychlení (nastavení pomocí MD!) až do dosažení naprogramované rychlosti posuvu.



Obrázek 4-13 Průběh rychlosti pohybu po dráze v případě příkazů `BRISK` a `SOFT`



Obrázek 4-14 Průběh rychlosti pohybu po dráze v případě příkazu `DRIVE`

Syntaxe

```
BRISK
BRISKA(<osa1>,<osa2>,…)
SOFT
SOFTA(<osa1>,<osa2>,…)
DRIVE
DRIVEA(<osa1>,<osa2>,…)
```

Význam

`BRISK`:	Příkaz pro aktivování "Zrychlení bez omezení ryvu" pro dráhové osy.
`BRISKA`:	Příkaz pro aktivování "Zrychlení bez omezení ryvu" pro pohyby jednotlivých os (JOG, JOG/INC, polohovací osy, kyvné osy atd.).

SOFT:	Příkaz pro aktivování "Zrychlení s omezením ryvu" pro dráhové osy.
SOFTA:	Příkaz pro aktivování "Zrychlení s omezením ryvu" pro pohyby jednotlivých os (JOG, JOG/INC, polohovací osy, kyvné osy atd.).
DRIVE:	Příkaz pro aktivování omezené hodnoty zrychlení nad mezní hodnotou rychlosti nastavenou v konfiguraci (MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT) pro dráhové osy.
DRIVEA:	Příkaz pro aktivování omezené hodnoty zrychlení nad mezní hodnotou rychlosti nastavenou v konfiguraci (MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT) pro pohyby jednotlivých os (JOG, JOG/INC, polohovací osy, kyvné osy atd.).
(<osa1>,<osa2>,...):	Jednotlivé osy, pro které má vyvolávaný způsob chování zrychlení platit.

Okrajové podmínky

Změna způsobu chování při změnách zrychlení v průběhu obrábění

Jestliže se ve výrobním programu v průběhu opracovávání mění způsob chování při změnách zrychlení (BRISK <--> SOFT), potom se i při řízení pohybu po dráze uskutečňuje na přechodech mezi bloky přesné najetí na konci bloku.

Příklady

Příklad 1: SOFT a BRISKA

Programový kód

```
N10 G1 X... Y... F900 SOFT
N20 BRISKA(AX5,AX6)
...
```

Příklad 2: DRIVE a DRIVEA

Programový kód

```
N05 DRIVE
N10 G1 X... Y... F1000
N20 DRIVEA (AX4, AX6)
...
```

4.14.2.2 Ovlivňování chování zrychlení u vlečných os (VELOLIMA, ACCLIMA, JERKLIMA)

V případě vazby mezi osami (Strana 905) (tangenciální vlečení, synchronizované osy, spojení s řídící hodnotou a elektronická převodovka) se pohyb vlečných os/vřeten uskutečňuje v závislosti na jedné nebo na několika řídících osách/vřetenech.

Omezení dynamiky pohybů vlečných os/vřeten je možno ovlivňovat buď z výrobního programu nebo ze synchronní akce pomocí funkcí VELOLIMA, ACCLIMA a JERKLIMA, a to i když je vazba mezi osami už aktivní.

Poznámka

Funkce JERLIMA není k dispozici pro všechny druhy vazeb mezi osami.

- Příkazy ACCLIMA a VELOLIMA se uplatňují v režimu vřetena na celkový pohyb vřetena, takže i v případě, když je vazba deaktivována. V proměnných $AC_SMAXACC_INFO, příp. $AC_SMAXVELO_INFO, je prostřednictvím identifikace 22 signalizováno omezení.
- V režimu vřetena je funkce VELOLIMA omezena na 100 procent.
- Pokud je vřeteno ve vazbě, uplatňuje se případná přípustná rychlost:
  Jestliže již byly pro pohyb v důsledku spojení vazbou vyčerpány maximální otáčky a pokud se v důsledku toho zastaví naprogramovaný základní pohyb, aktivuje se alarm 22015 „Vlečné vřeteno nemá žádnou dynamiku pro doplňkový pohyb".

Syntaxe

```
VELOLIMA(<osa>)=<hodnota>
ACCLIMA(<osa>)=<hodnota>
JERKLIMA(<osa>)=<hodnota>
```

Význam

VELOLIMA:	Příkaz pro korekci v parametrech nastavené maximální **rychlosti**.
ACCLIMA:	Příkaz pro korekci v parametrech nastaveného maximálního **zrychlení**.
JERKLIMA:	Příkaz pro korekci v parametrech nastaveného maximálního **ryvu**.
<osa>:	Vlečná osa, jejíž omezení dynamiky pohybů má být korigováno.
<hodnota>:	Procentuální hodnota korekce

Příklady

Příklad 1: Korekce omezení dynamiky pohybů pro jednu vlečnou osu (AX4)

Programový kód	Komentář
...	
VELOLIMA[AX4]=75	; omezovací korekce na 75% maximální rychlosti osy uložené ve strojním parametru
ACCLIMA[AX4]=50	; omezovací korekce na 50% maximálního zrychlení osy uložené ve strojním parametru
JERKLIMA[AX4]=50	; omezovací korekce při pohybu po dráze na 50% maximálního ryvu osy uloženého ve strojním parametru
...	

Příklad 2: Elektronická převodovka

Osa 4 je prostřednictvím vazby "Elektronická převodovka" spojena s osou X. Chování zrychlení vlečné osy je omezeno na 70% maximálního zrychlení. Maximální přípustná rychlost bude omezena na 50% maximální rychlosti. Po úspěšném aktivování vazby bude maximální rychlost znovu nastavena na 100%.

```
Programový kód                          Komentář
...
N120 ACCLIMA[AX4]=70                    ; snížené maximální zrychlení
N130 VELOLIMA[AX4]=50                   ; snížená maximální rychlost
...
N150 EGON(AX4,"FINE",X,1,2)             ; aktivování vazby typu "Elektronická převo-
                                        dovka"
...
N200 VELOLIMA[AX4]=100                  ; plná maximální rychlost
...
```

Příklad 3: Spojení pomocí řídící hodnoty s ovlivňováním pomocí statické synchronní akce

Osa 4 je spojena pomocí vazby s řídící hodnotou s osou X. Chování zrychlení je omezeno prostřednictvím statické synchronní akce 2 od pozice 100 na 80%.

```
Programový kód                                  Komentář
...
N120 IDS=2 WHENEVER $AA_IM[AX4] >               ; synchronní akce
100 DO ACCLIMA[AX4]=80
N130 LEADON(AX4, X, 2)                          ; aktivování vazby řídící hodno-
                                                tou
...
```

4.14.2.3 Aktivování specifických technologických hodnot dynamiky (DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN, DYNFINISH), DYNPREC

Pomocí příkazů z G-skupiny 59 „Režim dynamiky pro dráhovou interpolaci" je možné pro různé technologické kroky postupu opracování aktivovat k nim vyhovující dynamiku.

Hodnoty dynamiky a příkazy G-funkcí jsou nastavovány v konfiguraci a proto jsou závislé na nastaveních strojních parametrů.

Další informace: Příručka k funkcím, Základní funkce

Syntaxe

Aktivování hodnot dynamiky:
DYNNORM/DYNPOS/DYNROUGH/DYNSEMIFIN/DYNFINISH/DYNPREC

Poznámka

Hodnoty dynamiky jsou v platnosti už v bloku, ve kterém je příslušný příkaz G-funkce naprogramován. Nedochází k žádnému zastavení obrábění.

Čtení nebo zápis určitého prvku pole:
`R<m>=$MA...[n,X]`
`$MA...[n,X]=<hodnota>`

Význam

`DYNNORM`:	Aktivování **normální dynamiky**	
`DYNPOS`:	Aktivování dynamiky pro **režim polohování, vrtání závitu**	
`DYNROUGH`:	Aktivování dynamiky pro **obrábění nahrubo**	
`DYNSEMIFIN`:	Aktivování dynamiky pro **předběžné obrábění načisto**	
`DYNFINISH`:	Aktivování dynamiky pro **obrábění načisto**	
`DYNPREC`:	Aktivování dynamiky pro **jemné obrábění načisto**	
`R<m>`:	Početní parametr s číslem `<m>`	
`$MA...[n,X]`:	Strojní parametr s prvkem pole určujícím dynamiku	
`<n>`:	Index pole	
	Rozsah hodnot:	0 ... 5
	0	Normální dynamika (DYNNORM)
	1	Dynamika pro režim polohování (DYNPOS)
	2	Dynamika pro obrábění nahrubo (DYNROUGH)
	3	Dynamika pro předběžné obrábění načisto (DYNSEMIFIN)
	4	Dynamika pro obrábění načisto (DYNFINISH)
	5	Dynamika pro jemné obrábění načisto (DYNPREC)
`<X>`:	Adresa osy	
`<hodnota>`:	Hodnota dynamiky	

Příklady

Příklad 1: Aktivování hodnot dynamiky

Programový kód	**Komentář**
`DYNNORM G1 X10`	`; základní nastavení`
`DYNPOS G1 X10 Y20 Z30 F…`	`; režim polohování, vrtání závitu`
`DYNROUGH G1 X10 Y20 Z30 F10000`	`; obrábění nahrubo`
`DYNSEMIFIN G1 X10 Y20 Z30 F2000`	`; Předběžné obrábění načisto`
`DYNFINISH G1 X10 Y20 Z30 F1000`	`; obrábění načisto`
`DYNPREC G1 X10 Y20 Z30 F600`	`; jemné obrábění načisto`

Příklad 2: Čtení nebo zápis určitého prvku pole

Maximální zrychlení pro obrábění nahrubo, osa X.

```
Programový kód                          Komentář
R1=$MA_MAX_AX_ACCEL[2,X]                ; čtení
$MA_MAX_AX_ACCEL[2,X]=5                 ; zápis
```

4.14.3 Najíždění s dopřednou regulací (FFWON, FFWOF)

Prostřednictvím dopředné regulace se při pohybu po dráze snižuje na rychlosti závislá velikost doběhového úseku na nulu. Pohyby s dopřednou regulací umožňují vyšší přesnost pohybu po dráze a tím i lepší výsledky obrábění.

Syntaxe

```
FFWON
FFWOF
```

Význam

FFWON:	Příkaz pro **zapnutí** dopředné regulace
FFWOF:	Příkaz pro **vypnutí** dopředné regulace

Poznámka

Druh dopředné regulace je definován pomocí strojních parametrů, stejně jako je takto určeno, u kterých dráhových os má být pohyb pomocí dopředné regulace řízen.

Standard: Dopředná regulace závisející na rychlosti

Volba: Dopředná regulace závisející na zrychlení

Příklad:

```
Programový kód
N10 FFWON
N20 G1 X… Y… F900 SOFT
```

4.14.4 Programování tolerance kontury/orientace (CTOL, OTOL, ATOL)

Prostřednictvím adres CTOL, OTOL a ATOL je možné přizpůsobovat tolerance opracování definované ve strojních a nastavovaných parametrech pro funkce kompresoru, zaoblení a vyhlazování orientace ve výrobním programu.

Naprogramované hodnoty tolerancí platí, dokud nejsou znovu naprogramovány nebo dokud nejsou přiřazením záporné hodnoty vymazány. Kromě toho jsou vymazány na konci programu, při resetu. Po vymazání jsou opět v platnosti hodnoty tolerancí nastavené v parametrech.

Syntaxe

```
CTOL=<Value>
OTOL=<Value>
ATOL[<Axis>]=<Value>
```

Význam

CTOL:	Adresa pro programování **tolerance kontury**			
	Oblast použití:	• všechny funkce kompresoru • všechny pohyby na přechodových zaobleních, kromě příkazů G641 a G644		
	Zastavení předběžného zpracování:	ne		
	Platnost:	modální		
	<Value>:	Hodnota pro toleranci kontury je délkový údaj.		
		Typ:	REAL	
		Jednotka:	Palce/mm (v závislosti na momentálním nastavení měřicích jednotek)	
		Rozsah hodnot:	≥ 0:	Hodnota tolerance
			< 0:	Vymazání naprogramované hodnoty tolerance ⇒ Znovu platí hodnoty tolerancí definované ve strojních, příp. nastavovaných parametrech.
OTOL:	Adresa pro programování **tolerance orientace**			
	Oblast použití:	• všechny funkce kompresoru • Vyhlazování orientace pomocí příkazu ORISON • všechny pohyby na přechodových zaobleních, kromě příkazů G641, G644 a OSD		
	Zastavení předběžného zpracování:	ne		
	Platnost:	modální		
	<Value>:	Hodnota pro toleranci orientace je úhlový údaj.		
		Typ:	REAL	
		Jednotka:	stupně	
		Rozsah hodnot:	≥ 0:	Hodnota tolerance
			< 0:	Vymazání naprogramované hodnoty tolerance ⇒ Znovu platí hodnoty tolerancí definované ve strojních, příp. nastavovaných parametrech.

ATOL:	Adresa pro programování **tolerance určité osy**				
	Oblast použití:	• všechny funkce kompresoru • Vyhlazování orientace pomocí příkazu ORISON • všechny pohyby na přechodových zaobleních, kromě příkazů G641, G644 a OSD			
	Zastavení předběžného zpracování:	ne			
	Platnost:	modální			
	<Axis>:	Název kanálové osy, pro kterou má naprogramovaná tolerance platit			
	<Value>:	Hodnota pro toleranci osy je v závislosti na typu osy (lineární nebo kruhová osa) buď délkovým nebo úhlovým údajem.			
		Typ:	REAL		
		Jednotka:	pro lineární osy:		Palce/mm (v závislosti na momentálním nastavení měřicích jednotek)
			pro kruhové osy:		stupně
		Rozsah hodnot:	≥ 0:	Hodnota tolerance	
			< 0:	Vymazání naprogramované hodnoty tolerance ⇒ Znovu platí hodnoty tolerancí definované ve strojních, příp. nastavovaných parametrech.	

Poznámka

Kanálové hodnoty tolerancí naprogramované pomocí příkazů CTOL a OTOL mají vyšší prioritu než osové hodnoty tolerancí naprogramované pomocí příkazu ATOL.

Poznámka

Framy provádějící změnu měřítka

Framy provádějící změnu měřítka ovlivňují naprogramované tolerance stejným způsobem, jako polohy os, tzn. relativní tolerance zůstává stejná.

Příklad:

Programový kód	**Komentář**
COMPCAD G645 G1 F10000	; aktivování funkce kompresoru COMPCAD
X... Y... Z...	; zde se uplatňují strojní a nastavované parametry
X... Y... Z...	
X... Y... Z...	
CTOL=0.02	; od tohoto místa se uplatňuje tolerance kontury 0,02 mm
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	

Programový kód	**Komentář**
ASCALE X0.25 Y0.25 Z0.25	; od tohoto místa se uplatňuje tolerance kontury 0,005 mm
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	
CTOL=-1	; od tohoto místa se znovu uplatňují strojní a nastavované parametry
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	

Další informace

Systémové proměnné

Prostřednictvím následujících systémových proměnných je možno načítat momentálně platné tolerance:

- Čtení se zastavením předběžného zpracování (ve výrobním programu a synchronní akci)
  - $AC_CTOL
    Kanálová tolerance kontury, která byla v platnosti při přípravě aktuálního bloku v hlavním zpracování.
    Jestliže žádná tolerance kontury není v platnosti, v proměnné $AC_CTOL se nachází odmocnina ze součtu druhých mocnin tolerancí geometrických os.
  - $AC_OTOL
    Kanálová tolerance orientace, která byla v platnosti při přípravě aktuálního bloku v hlavním zpracování.
    Jestliže žádná tolerance orientace není v platnosti, v proměnné $AC_OTOL se v průběhu aktivní transformace orientace nachází odmocnina ze součtu druhých mocnin tolerancí orientačních os, jinak je tam hodnota "-1".
  - $AA_ATOL[<osa>]
    Osová tolerance kontury, která byla v platnosti při přípravě aktuálního bloku v hlavním zpracování.
    Pokud je aktivní tolerance kontury, obsahuje proměnná $AA_ATOL[<geometrická osa>] toleranci kontury dělenou odmocninou z počtu geometrických os.
    Pokud je aktivní tolerance orientace a pokud je aktivní nějaká transformace orientace, obsahuje proměnná $AA_ATOL[<orientační osa>] toleranci orientace dělenou odmocninou z počtu orientačních os.

Poznámka

Jestliže nebyly naprogramovány žádné hodnoty tolerance, potom proměnné $A... nejsou dostatečně diferencované, aby bylo možné rozlišit tolerance jednotlivých funkcí.

Takové případy se mohou vyskytnout, jestliže jsou pomocí strojních a nastavovaných parametrů nastaveny odlišné tolerance pro funkce kompresoru, přechodová zaoblení a vyhlazování orientace. Systémové proměnné potom obsahují největší hodnotu, jaká se vyskytuje u momentálně aktivních funkcí. Jestliže je např. aktivní funkce kompresoru s tolerancí orientace 0,1° a vyhlazování orientace pomocí funkce ORISON s hodnotou 1°, poskytuje proměnná $AC_OTOL hodnotu "1". Když se vyhlazování orientace vypne, poskytuje proměnná je možné $AC_OTOL hodnotu „0,1".

- Načítání bez zastavení předběžného zpracování (jen ve výrobním programu)
  - $P_CTOL
    Momentálně platná kanálová tolerance kontury.
  - $P_OTOL
    Momentálně platná kanálová tolerance orientace.
  - $PA_ATOL
    Momentálně platná osová tolerance kontury.

Okrajové podmínky

Tolerance naprogramované pomocí příkazů CTOL, OTOL a ATOL platí také pro funkce, které jsou na těchto tolerancích nepřímo závislé:

- Omezení chyby tětivy při výpočtu požadované hodnoty
- Základní funkce režimu pro frézování ploch volných tvarů

Následující funkce pro vyhlazení průběhu **nejsou** naprogramováním příkazů CTOL, OTOL a ATOL ovlivňovány:

- Vyhlazení průběhu orientace pomocí příkazu OSD
  Příkaz OSD nepoužívá žádnou toleranci, nýbrž vzdálenost k přechodu na další blok.
- Přechodová zaoblení u G644
  Příkaz G644 neslouží k obrábění, ale k optimalizaci výměn nástroje a jiných pohybů, při kterých neprobíhá obrábění.
- Přechodová zaoblení u G645
  Příkaz G645 se chová téměř vždy jako příkaz G642, a tudíž naprogramované tolerance používá. Pouze na hranicích bloků s tangenciálně spojitým napojením se skokovým zakřivením, např. na tangenciálním přechodu kruh-přímka, se používá hodnota tolerance ze strojního parametru MD33120 $MA_PATH_TRANS_POS_TOL. Dráha přechodového zaoblení se na těchto místech může nacházet téměř až na vnější straně naprogramované kontury, kde je u mnoha aplikací tolerance menší. Kromě toho zpravidla postačuje malá, napevno nastavená tolerance pro kompenzaci změn zakřivení, s nimiž si programátor NC systému pak už nemusí dělat žádné starosti.

4.14.5 Aktivování/deaktivování programovatelné přesnosti kontury (CPRECON, CPRECOF)

Funkce "Programovatelná přesnost kontury" snižuje prostřednictvím automatického přizpůsobení rychlosti chybu dráhy na zakřivených konturách.

Tuto funkci můžete aktivovat, příp. deaktivovat, v NC programu pomocí příkazů s modální platností z G-skupiny 39 (programovatelná přesnost kontury).

Syntaxe

```
CPRECON
...
CPRECOF
```

Význam

CPRECON:	**Aktivování** funkce „Programovatelná přesnost kontury“
CPRECOF:	**Deaktivování** funkce „Programovatelná přesnost kontury“

Příklad

Programový kód	Komentář
N10 G0 X0 Y0	
N20 CPRECON	; aktivování funkce "programovatelná přesnost kontury"
N30 G1 G64 X100 F10000	; obrábění s rychlostí 10 m/min v režimu řízení pohybu po dráze
N40 G3 Y20 J10	; automatické omezení posuvu v bloku kruhového oblouku
N50 G1 X0	; posuv opět bez omezení (10 m/min)
...	
N100 CPRECOF	; deaktivování funkce "programovatelná přesnost kontury"
N110 G0 ...	

Viz také

Programování tolerance kontury/orientace (CTOL, OTOL, ATOL) (Strana 849)

Další informace

Přesnost kontury

Přesnost kontury, kterou je potřeba dodržet, se zadává v závislosti na konfiguraci stroje (MD20470 $MC_MC_CPREC_WITH_FFW, viz informace od výrobce stroje) buď pomocí nastavovaného parametru SD42450$SC_CONTPREC nebo pomocí naprogramované tolerance kontury CTOL. Čím menší je tato hodnota a čím menší je parametr K_V geometrických os, tím více se rychlost pohybu po dráze na zakřivených konturách snižuje.

Přesnost kontury při pohybech s rychlým posuvem

Při opracovávání obrobků se zakřivenými konturami a s aktivní funkcí „Programovatelná přesnost kontury“ se za účelem dodržení předepsané přesnosti kontury snižuje rychlost pohybu po dráze i u pohybů nástroje rychlým posuvem, např. v rozích a v blocích přechodových zaoblení při objíždění obrobku. Aby se snížení rychlosti pohybu po dráze při pohybech rychlým posuvem minimalizovalo, může být pro funkci „Programovatelná přesnost kontury“ nastavena přesnost kontury pro pohyby rychlým posuvem, která se liší od hodnoty stanovené pro obrábění obrobku:

SD42451 $SC_CONTPREC_G00_ABS (přesnost kontury při pohybu rychlým posuvem)

Když je nastaveno SD42451 = 0, platí pro pohyby rychlým posuvem přesnost kontury definovaná parametrem $SC_CONTPREC[DYNNORM].

Minimální posuv po dráze

Pro funkci „Programovatelná přesnost kontury“ může uživatel prostřednictvím následujícího nastavovaného parametru definovat minimální posuv po dráze:

SD42460 $SC_MINFEED (minimální posuv po dráze, když je aktivní příkaz CPRECON)

Posuv nebude omezen na hodnotu nižší, než je tato hodnota, leda že by byla naprogramována nižší hodnota F nebo si dynamická omezení os vynutila nižší rychlost pohybu po dráze.

Žádný vliv na polohovací osy

Funkce "Programovatelná přesnost kontury" uvažuje jen geometrické osy dané dráhy. Na rychlosti polohovacích os nemá žádný vliv.

Chování na začátku výrobního programu a po resetu kanálu, příp. resetu na konci programu

Na začátku výrobního programu a po resetu kanálu, příp. resetu na konci programu, je v platnosti základní nastavení řídícího systému, které je definováno pro skupinu G-funkcí č. 39 (viz informace od výrobce stroje).

4.14.6 Aktivování/deaktivování automatického přepínání filtru (AFISON, AFISOF)

Díky funkci „Automatické přepínání filtru“ má uživatel možnost v rámci NC programu definovat oblasti, ve kterých jsou všechny pro tuto funkci uvolněné osy při pohybech s příkazem G0 automaticky přepínány na 2. filtrační řetězec. Jestliže jsou parametry 2. filtračního řetězce nastaveny tak, že jsou vibrace silně tlumeny, je možné u pohybů s příkazem G0 nastavit vyšší hodnotu ryvu. V důsledku toho je rychlost pohybu po dráze s příkazy G0, např. v rozích, méně silně omezována a je dosahováno zkrácení doby zpracování programu.

Syntaxe

Příkazy jazyka pro aktivování a deaktivování této funkce se vždy musí nacházet v samostatném bloku.

```
AFISON
...
AFISOF
```

Význam

AFISON	**Aktivování** funkce „Automatické přepínání filtru“
AFISOF	**Deaktivování** funkce „Automatické přepínání filtru“

Příklad

```
Programový kód          Komentář
...
N390 G1 X1100           ; Je v platnosti filtrační řetězec 1
                        ; Pozastavení v bodě X=1100
N400 AFISON             ; Automatické přepínání na 2. filtrační řetězec
N410 G0 X1150           ; Pozastavení v bodě X=1150
N420 G1 X1200           ; Automatické přepínání na 1. filtrační řetězec
N430 AFISOF
N440 G0 X1300           ; Filtrační řetězec 1 je v platnosti i v bloku s G0
N450 G1 X1400
```

```
Programový kód          Komentář
N460 AFISON
N470 G1 X1450
N480 G1 X1500
N490 AFISOF
N500 G1 X1600
N510 G0 X1700           ; Filtrační řetězec 1 je v platnosti i v bloku s G0
N520 AFISON
N530 G1 X1750           ; Pozastavení v bodě X=1750
                        ; Automatické přepínání na 2. filtrační řetězec
N540 G0 X1800           ; Pozastavení v bodě X=1800
                        ; Automatické přepínání na 1. filtrační řetězec
N550 AFISOF
N560 G1 X1900
N570 G0 X2000           ; Filtrační řetězec 1 je v platnosti i v bloku s G0
                        ; Pozastavení v bodě X=2100
N580 AFISON             ; Automatické přepínání na 2. filtrační řetězec
N590 G0 X2050
N600 G0 X2100           ; Pozastavení v bodě X=2100
                        ; Automatické přepínání na 1. filtrační řetězec
N610 AFISOF
N620 G0 X2200           ; Filtrační řetězec 1 je v platnosti i v bloku s G0
...
```

Další informace

Předpoklady

Aby bylo možno funkci „Automatické přepínání filtru" používat, musí být splněny následující předpoklady:

- Musí být aktivována funkce „Přizpůsobení ryvu", pro kterou je nutno mít licenci:
  MD19321 $ON_TECHNO_FUNCTION_MASK_1, Bit 22 = 1
- Funkce musí být pro příslušný kanál uvolněna:
  MD20630 $MC_AFIS_MODE = 1
- Funkce musí být uvolněna pro každou osu, u které se automatické přepínání filtru předpokládá:
  MD32332 $MD_AFIS_ENABLE = 1
- U všech os, které jsou pro tuto funkci uvolněny:
  - musí být aktivní omezení ryvu:
    MD32400 $MA_AX_JERK_ENABLE = 1
  - musí být zvoleny a nastaveny dva typy filtru pro omezení ryvu (→ MD32402 $MA_AX_JERK_MODE).

Jestliže tyto předpoklady nejsou splněny, aktivuje se alarm 14782, příp. 26380.

Pozastavení při aktivování/deaktivování

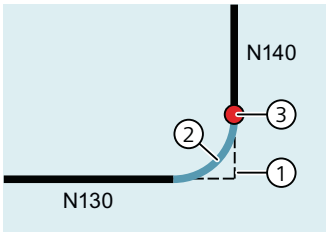
Při aktivování a deaktivování této funkce dochází k pozastavení pohybu po dráze jen tehdy, jestliže je nezbytné přepnutí filtračního řetězce pro požadovanou hodnotu polohy.

Zaoblení

Jestliže se při aktivních přechodových zaobleních (G64x) provádí při přechodu na další blok pozastavení a automatické přepnutí filtračního řetězce, nemá to žádný vliv na konturu zaoblení.

Příklad:

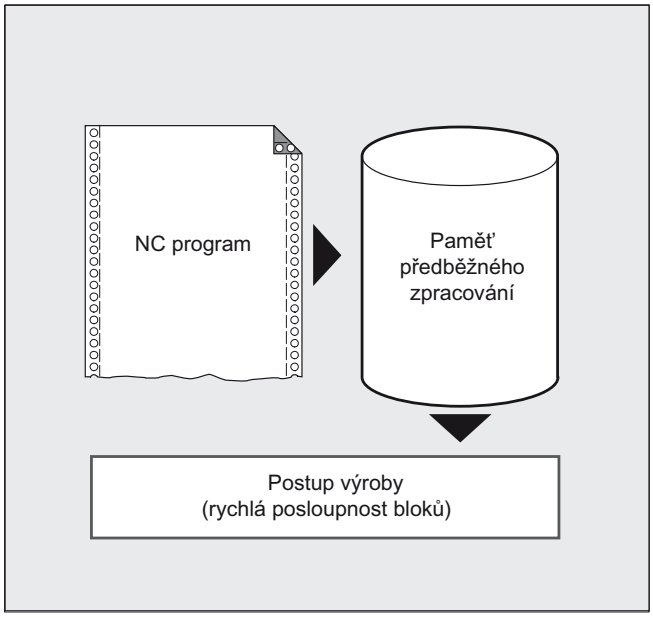
```
N100 G642 F1000 CTOL=10 CTOLG0=10
N110 G0 X0 Y0
N120 AFISON
N130 G1 X100 Y0
N140 G0 X100 Y100
```



① Naprogramovaná dráha
② Kontura zaoblení
③ Pozastavení a přepnutí na 2. filtrační řetězec

4.14.7 Zpracování programu s pamětí předběžného zpracování (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE)

V závislosti na úrovni svého vybavení disponuje řídící systém určitým množstvím tzv. paměti pro předběžné zpracování, do které se ukládají připravené hotové bloky před svým zpracování a odkud jsou v průběhu samotného obrábění předávány jako rychlá posloupnost bloků. Díky tomu mohou být krátké bloky obrobeny s vysokými rychlostmi. Jestliže to doba prodlevy řídícího systému připouští, je paměť předběžného zpracování v zásadě naplněna.



Označení zpracovávaného úseku

Zpracovávaný úsek programu, který má být dočasně uložen do paměti předběžného zpracování, je ve výrobním programu označen na začátku příkazem „STOPFIFO“ a na konci příkazem „STARTFIFO“. Zpracování připravených a do vyrovnávací paměti uložených bloků začíná teprve po příkazu „STARTFIFO“ nebo když je paměť předběžného zpracování plná.

Automatické řízení paměti předběžného zpracování

Automatické řízení paměti předběžného zpracování se vyvolává pomocí příkazu „FIFOCTRL“. Příkaz „FIFOCTRL“ z počátku funguje úplně stejně jako příkaz „STOPFIFO“. Při každé sekvenci programových příkazů se čeká, dokud není paměť předběžného zpracování plná, a potom se spouští zpracování. Rozdíl mezi oběma možnostmi spočívá v chování v případě vyprázdnění paměti předběžného zpracování: V případě příkazu „FIFOCTRL“ se od úrovně 2/3 naplnění postupně snižuje rychlost pohybu po dráze, aby se zabránilo úplnému vyprázdnění a zabrždění pohybu os až do úplného klidu.

Zastavení předběžného zpracování

Když je v bloku naprogramován příkaz „STOPRE“, příprava bloků a ukládání do vyrovnávací paměti se zastaví. Následující blok bude zpracován teprve tehdy, až jsou zcela zpracovány

všechny bloky, které již dříve byly připraveny a uloženy do paměti. Předcházející blok bude zastaven v přesném najetí (jako při G9).

UPOZORNĚNÍ
Přerušení programu Když jsou aktivovány korekce nástroje a splinová interpolace, neměl by být naprogramován žádný příkaz „STOPRE", protože by došlo k přerušení posloupnosti k sobě patřících bloků.

Syntaxe

Tabulka 4-3 Označení zpracovávaného úseku:

```
STOPFIFO
...
STARTFIFO
```

Tabulka 4-4 Automatické řízení paměti předběžného zpracování:

```
...
FIFOCTRL
...
```

Tabulka 4-5 Zastavení předběžného zpracování:

```
...
STOPRE
...
```

Poznámka

Příkazy „STOPFIFO", „STARTFIFO", „FIFOCTRL" a „STOPRE" musí být naprogramovány v samostatném bloku.

Význam

STOPFIFO:	Příkaz "STOPFIFO" označuje začátek zpracovávaného úseku, který má být dočasně uložen do paměti předběžného zpracování. Pomocí příkazu "STOPFIFO" se zpracovávání pozastaví a paměť předběžného zpracování se bude plnit, dokud: • není rozpoznán příkaz „STARTFIFO“ nebo „STOPRE“. nebo • se paměť předběžného zpracování nenaplní nebo • není dosaženo konce programu.
STARTFIFO:	Pomocí příkazu "STARTFIFO" se spouští rychlé zpracování úseku programu; souběžně s tím probíhá plnění paměti předběžného zpracování.
FIFOCTRL:	Spuštění automatického ukládání do paměti předběžného zpracovávání
STOPRE:	Zastavení předběžného zpracování

Poznámka

Plnění paměti předběžného zpracování se neprovádí, příp. je přerušeno, pokud zpracovávaný úsek programu obsahuje příkazy, které vyžadují okamžité zpracování (najíždění na referenční bod, měřicí funkce, ...).

Poznámka

Při přístupu ke stavovým údajům stroje ($SA...) generuje řídící systém interní zastavení předběžného zpracování.

Příklad: Zastavení předběžného zpracování

```
Programový kód                                  Komentář
...
N30 MEAW=1 G1 F1000 X100 Y100 Z50               ; Blok měření s měřicí sondou na prvním
                                                měřicím vstupu a přímková interpolace.
N40 STOPRE                                      ; Zastavení předběžného zpracování.
...
```

4.14.8 Definice oblasti odloženého zastavení (DELAYFSTON, DELAYFSTOF)

Pro definování podmínečně přerušitelných oblastí ve výrobním programu (Stop-Delay) slouží předem definované procedury DELAYFSTON a DELAYFSTOF.

Poznámka

V synchronních akcích je použití funkcí DELAYFSTON a DELAYFSTOF **nepřípustné**!

Syntaxe

```
DELAYFSTON
...
DELAYFSTOF
```

Význam

DELAYFSTON:	Definice začátku oblasti odloženého zastavení	
	Samostatný blok:	ano
DELAYFSTOF:	Definice konce oblasti odloženého zastavení (Stop_Delay)	
	Samostatný blok:	ano

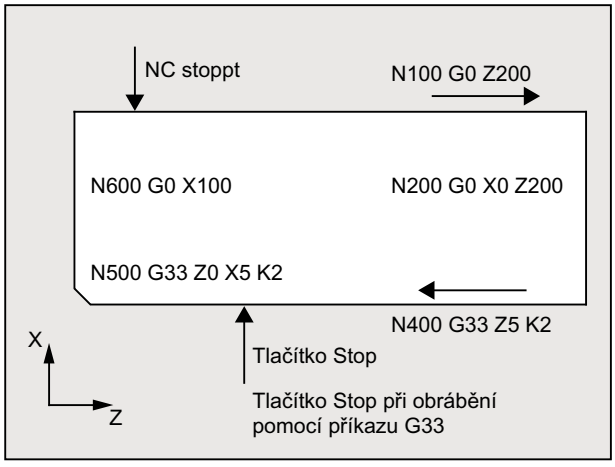
Příklad programování

Ve smyčce se opakuje následující blok:

Programový kód

```
...
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
...
```

V následujícím obrázku je vidět, že když uživatel v oblasti odloženého zastavení (Stop-Delay) stiskne tlačítko "Stop", zahájí NC systém operaci brždění mimo oblast odloženého zastavení, tzn. v bloku N100. Díky tomu se NC systém v oblasti před blokem N100 zastaví.



Další informace

Konec podprogramu

Na konci podprogramu, v němž byl vyvolán příkaz DELAYFSTON, se implicitně aktivuje příkaz DELAYFSTOF.

Vnoření

Pokud podprogram 1 vyvolává v oblasti odloženého zastavení (Stop-Delay) podprogram 2, stává se celý podprogram 2 oblastí odloženého zastavení. Zejména příkazy DELAYFSTOF nemají v podprogramu 2 žádný efekt.

Příklad:

```
Programový kód                  Komentář
N10010 DELAYFSTON               ; bloky s čísly N10xxx z programové úrovně 1
N10020 R1 = R1 + 1
N10030 G4 F1                    ; začátek oblasti odloženého zastavení (Stop-Delay)
...
N10040 Podprogram 2
...
...                             ; interpretace podprogramu 2
N20010 DELAYFSTON               ; žádná platnost, opakovaný začátek, 2. rovina
...
N20020 DELAYFSTOF               ; žádná platnost, konec na jiné úrovni
N20030 RET
N10050 DELAYFSTOF               ; konec oblasti odloženého zastavení (Stop-Delay) na
                                stejné úrovni
...
N10060 R2 = R2 + 2
N10070 G4 F1                    ; oblast odloženého zastavení (Stop-Delay) končí Zasta-
                                vení jsou od tohoto okamžiku uskutečňována okamžitě.
```

Systémové proměnné

To, zda se zpracování výrobního programu v aktuálním okamžiku nachází v oblasti odloženého zastavení, je možno zjistit pomocí následujících systémových proměnných:

- ve výrobním programu pomocí proměnné $P_DELAYFST
- v synchronních akcích pomocí proměnné $AC_DELAYFST

Hodnota	Význam
0	Oblast odloženého zastavení není aktivní
1	Oblast odloženého zastavení je aktivní

4.14.9 Zabránění pozice programu pro SERUPRO (IPTRLOCK, IPTRUNLOCK)

Pro určité komplikované mechanické situace na stroji je zapotřebí, aby se zabránilo vyhledávání bloku pomocí příkazu SEROPRO.

Pomocí programovatelného ukazatele přerušení existuje možnost, jak při operaci "Vyhledávání místa přerušení" zasáhnout a přesunout se před prohledávané místo.

Ve výrobním programu mohou být definovány také oblasti vyhledávání, do kterých NC systém ještě nemůže znovu vstoupit. Při přerušení programu se v NC systému zaznamená naposled zpracovávaný blok, který potom může být pomocí uživatelského rozhraní HMI vyhledán.

Syntaxe

```
IPTRLOCK
IPTRUNLOCK
```

Tyto příkazy musí být uvedeny na samostatném řádku výrobního programu a umožňují pracovat s programovatelným ukazatelem přerušení.

Význam

IPTRLOCK:	Začátek úseku programu pro vyhledávání
IPTRUNLOCK:	Konec úseku programu pro vyhledávání

Oba příkazy jsou přípustné jen ve výrobních programech, **nikoli** v synchronních akcích.

Příklad:

Vnoření úseků programu, které mohou být prohledávány, na dvou programových úrovních s implicitním příkazem „IPTRUNLOCK". Implicitní příkaz „IPTRUNLOCK" v podprogramu 1 končí oblast určenou pro vyhledáváním.

Programový kód	**Komentář**
N10010 IPTRLOCK()	
N10020 R1 = R1 + 1	
N10030 G4 F1	; blok s pozastavením, úsek programu určený pro vyhledávání začíná
...	
N10040 Podprogram 2	
...	; interpretace podprogramu 2
N20010 IPTRLOCK ()	; žádná platnost, opakovaný začátek
...	
N20020 IPTRUNLOCK ()	; žádná platnost, konec v jiné rovině
N20030 RET	
...	
N10060 R2 = R2 + 2	
N10070 RET	; konec úseku programu pro vyhledávání
N100 G4 F2	; hlavní program pokračuje

Přerušení na řádku 100 potom znovu vytvoří ukazatel přerušení.

Další informace

Zjišťování a prohledávání oblastí určených pro vyhledávání

Úseky programu určené pro vyhledávání jsou označeny příkazy NC jazyka „IPTRLOCK“ a „IPTRUNLOCK“.

Příkaz „IPTRLOCK“ zmrazí ukazatele přerušení na určitém bloku (SB1), který může být realizován v hlavním zpracování. Tento blok se v následujícím označí jako blok se zastavením. Pokud se po příkazu „IPTRLOCK“ vyskytne přerušení programu, pak je možné na uživatelském rozhraní HMI tento tak zvaný blok zastavení vyhledat.

Opětovné nastavení na aktuální blok

Ukazatel přerušení se pomocí příkazu „IPTRLOCK“ pro následující úsek programu nastaví na aktuální blok k bodu, kde došlo k přerušení.

Po nalezení cíle vyhledávání může být pomocí stejného bloku zastavení cíl vyhledávání opakován.

Uživatelem editovaný ukazatel přerušení musí být pomocí HMI znovu odstraněn.

Pravidla pro vnoření

Chování příkazů NC jazyka „IPTRLOCK“ a „IPTRUNLOCK“ ve spojení s vnořeními a konci podprogramů se řídí následujícími pravidly:

1. Na konci podprogramu, v němž byl vyvolán příkaz "IPTRLOCK", se implicitně aktivuje příkaz "IPTRUNLOCK".
2. Oblast určená pro vyhledávání nastavená příkazem „IPTRLOCK“ nemá žádný efekt.
3. Pokud podprogram 1 vyvolá v oblasti určené pro vyhledávání podprogram 2, zůstává celý podprogram 2 určený pro vyhledávání. Zejména příkaz "IPTRUNLOCK" v podprogramu 2 nemá žádný efekt.

Další informace: Příručka k funkcím, Základní funkce

Systémová proměnná

Oblast určená pro vyhledávání může být ve výrobním programu rozpoznána pomocí proměnné "$P_IPTRLOCK".

Automatický ukazatel vyhledávání

Funkce automatického ukazatele vyhledávání automaticky definuje dříve nastavený typ vazby jako objekt určený pro vyhledávání. Prostřednictvím strojního parametru se pro

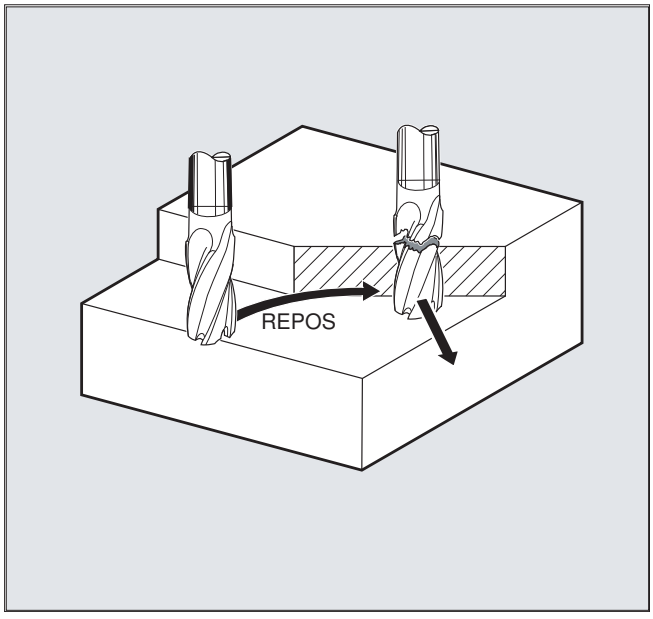
- Elektronickou převodovku s příkazem „EGON“
- Osovou vazbu řídící hodnotou s příkazem „LEADON“

aktivuje automatický ukazatel přerušení. Jestliže se naprogramované a prostřednictvím strojního parametru aktivované automatické ukazatele přerušení překrývají, vytvoří se největší možná oblast určená pro vyhledávání.

4.14.10 Opětovné najíždění na konturu (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)

Jestliže v průběhu zpracovávání běžící program přerušíte a s nástrojem odjedete od obrobku - např. kvůli zlomení nástroje nebo kvůli změření obrobku - můžete potom na konturu znovu najet ve zvoleném bodě, přičemž tento pohyb je řízený programem.

Příkaz `REPOS` způsobuje ASUP, stejně jako návrat zpět z podprogramu (např. M17). Následující bloky se už neuskuteční. Pokud budete potřebovat informace o přerušování programu, viz také "Rutiny přerušení (ASUP) (Strana 544)".



Syntaxe

```
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=…
```

Význam

Volba najížděcí dráhy

REPOSA:	Najetí zpět na konturu geometrickými osami po přímkách. Svou polohu mění také všechny ostatní kanálové osy.
REPOSL:	Najetí zpět na konturu geometrickými osami po přímkách. Všechny ostatní kanálové osy musí být explicitně naprogramovány.
REPOSQ DISR=… :	Najetí zpět na konturu geometrickými osami po čtvrtkruhu s rádiusem DISR. Všechny ostatní kanálové osy musí být explicitně naprogramovány.
REPOSQA DISR=… :	Najetí zpět na konturu geometrickými osami po čtvrtkruhu s rádiusem DISR. Svou polohu mění také všechny ostatní kanálové osy.
REPOSH DISR=… :	Najetí zpět na konturu geometrickými osami po půlkruhu s průměrem DISR. Všechny ostatní kanálové osy musí být explicitně naprogramovány.
REPOSHA DISR=… :	Najetí zpět na konturu geometrickými osami po půlkruhu s rádiusem DISR. Svou polohu mění také všechny ostatní kanálové osy.

Volba bodu pro opětovné najetí

RMIBL:	Najíždění na bod, kde došlo k přerušení
RMIBL DISPR=…:	Vstupní bod ve vzdálenosti DISPR v mm/palcích před bodem, kde došlo k přerušení
RMBBL:	Najíždění na počáteční bod bloku
RMEBL:	Najíždění na koncový bod bloku
RMEBL DISPR=… :	Najíždění na koncový bod bloku ve vzdálenosti DISPR před koncovým bodem
RMNBL:	Najíždění na nejblíže ležící bod dráhy
A0 B0 C0 :	Osy, kterými se má najíždět

Poznámka

Kompatibilita

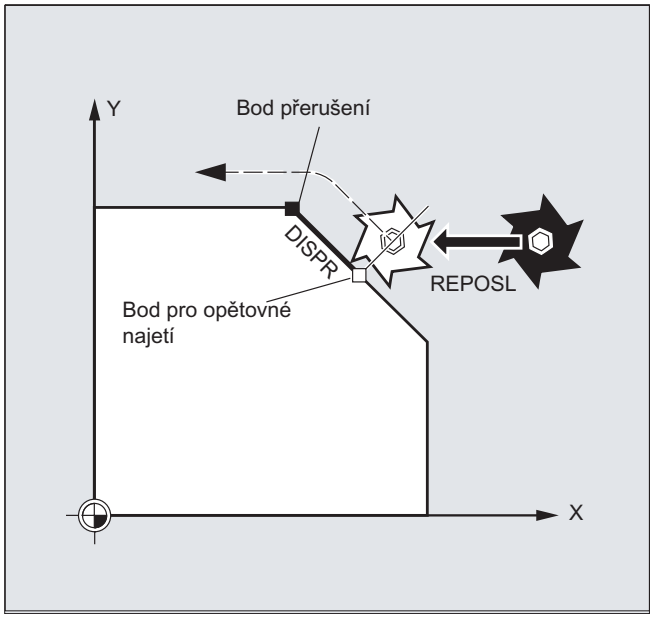
Aby byla zajištěna kompatibilita se staršími verzemi programového vybavení, můžete režim najíždění pro funkci REPOS i nadále programovat pomocí modálních příkazů RMI, RMB, RME a RMN. Při použití ve stavu ASUP by měl být tento příkaz opatřen atributem SAVE v příkazu PROC. Jinak se bude uplatňovat v režimu ASUP modální režim najíždění REPOS, a pokud se liší od standardního nastavení RMI, použije se i v následujících operacích REPOS.

Najetí zpět na konturu po přímkách, REPOSA, REPOSL

Nástroj najíždí na bod pro opětovné najetí přímo po přímce.

Příklad

```
REPOSL RMIBL DISPR=6 F400
```

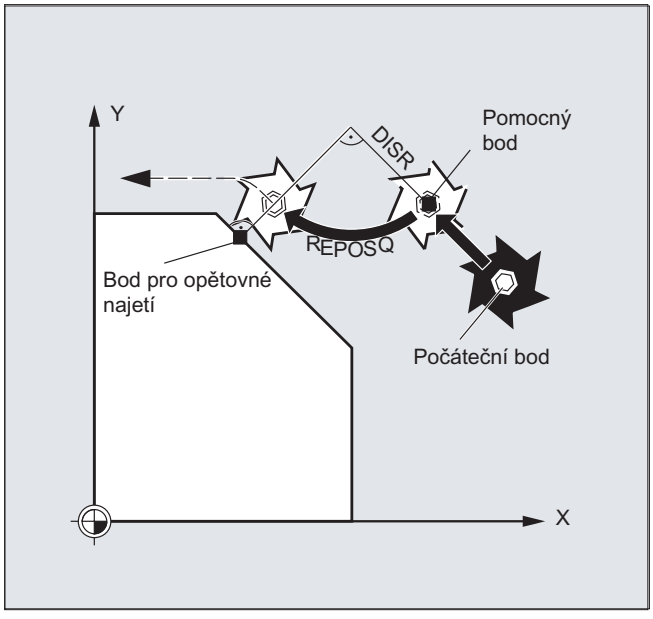


Najetí zpět na konturu po čtvrtkruhu, REPOSQ, REPOSQA

Nástroj najíždí na bod pro opětovné najetí po čtvrtkruhové dráze s rádiusem `DISR=...`. Potřebný pomocný bod mezi počátečním bodem a bodem pro opětovné najetí vypočítává řídící systém automaticky.

Příklad

```
REPOSQ RMIBL DISR=10 F400
```

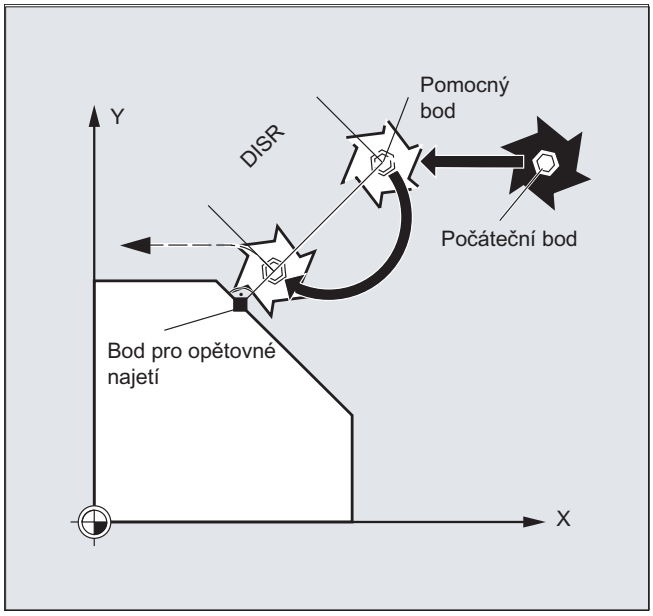


Najetí zpět na konturu po půlkruhu, REPOSH, REPOSHA

Nástroj najíždí na bod pro opětovné najetí po půlkruhové dráze s průměrem `DISR=...`. Potřebný pomocný bod mezi počátečním bodem a bodem pro opětovné najetí vypočítává řídící systém automaticky.

Příklad

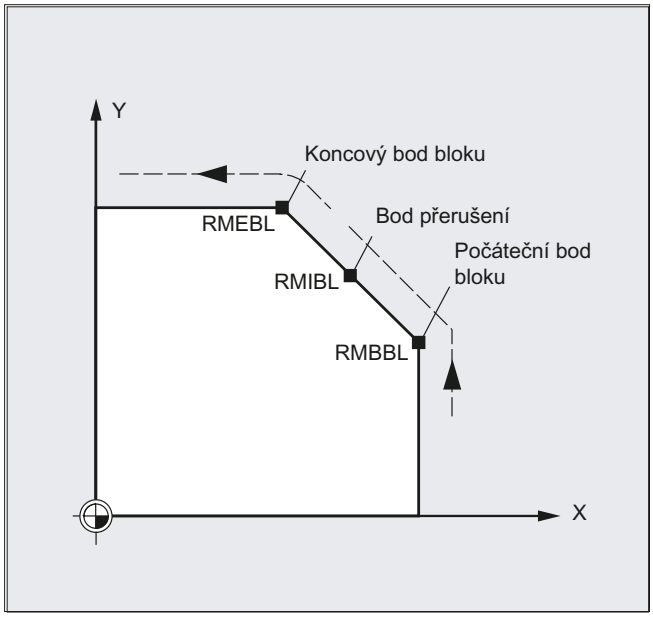
```
REPOSH RMIBL DISR=20 F400
```



Definice bodu pro opětovné najíždění (nikoli pro najíždění SERUPRO s RMNBL)

V souvislosti s NC blokem, ve kterém bylo zpracování programu přerušeno, si můžete vybrat mezi třemi body pro opětovné najetí:

- RMIBL, bod, kde došlo k přerušení
- RMBBL, počáteční bod bloku, příp. poslední koncový bod
- RMEBL, koncový bod bloku



Pomocí příkazů RMIBL DISPR=..., příp. RMEBL DISPR=... můžete definovat bod pro opětovné najíždění, který leží před bodem, v němž došlo k přerušení, příp. před koncovým bodem bloku.

Pomocí příkazu DISPR=... zadáváte dráhu na kontuře v mm/palcích, o kterou bod pro opětovné najíždění leží před bodem, v němž došlo k přerušení, příp. před koncovým bodem. Tento bod může - i v případě větších hodnot - ležet maximálně v počátečním bodu bloku.

Jestliže není naprogramována žádná hodnota parametru DISPR=..., platí nastavení DISPR=0 a v důsledku toho se použije bod, kde došlo k přerušení (v případě příkazu RMIBL), příp. koncový bod bloku (v případě RMEBL).

Znaménko parametru DISPR

Znaménko parametru DISPR se vyhodnocuje. Je-li znaménko kladné, funkce se chová, jak bylo popsáno výše.

V případě záporného znaménka se nástroj znovu nasazuje za bod, kde došlo k přerušení, příp. v případě příkazu RMBBL za počáteční bod.

Vzdálenost bodu přerušení a bodu nasazení nástroje vyplývá z hodnoty uložené v parametru DISPR. Tento bod může i v případě větších hodnot ležet maximálně v počátečním bodu bloku.

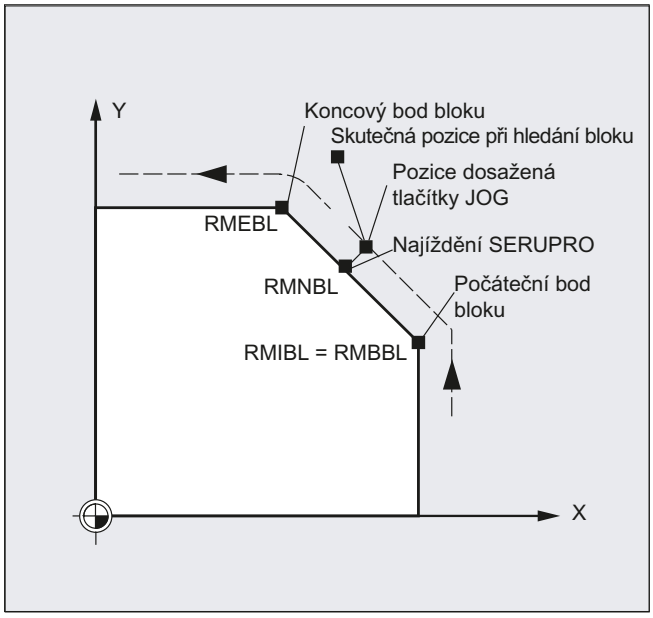
Příklad použití:

Pomocí senzoru bylo zjištěno přiblížení k upínací čelisti. Aktivuje se program ASUP, pomocí kterého se upínací čelist objede.

Nakonec se znovu najede na bod za upínací čelistí se zápornou hodnotou parametru DISPR a program bude pokračovat.

Najíždění pomocí příkazu SEROPRO s RMNBL

Jestliže si v průběhu obrábění okolnosti vyžádají na libovolném místě přerušení, pak se pomocí najíždění SERUPRO s RMNBL najede co možno nejblíže před místo, kde došlo k přerušení, aby se nakonec opracovala pouze zbývající dráha. Za tím účelem spouští uživatel operaci SERUPRO na bloku, kde došlo k přerušení, a pomocí tlačítek JOG najíždí před poškozené místo cílového bloku.



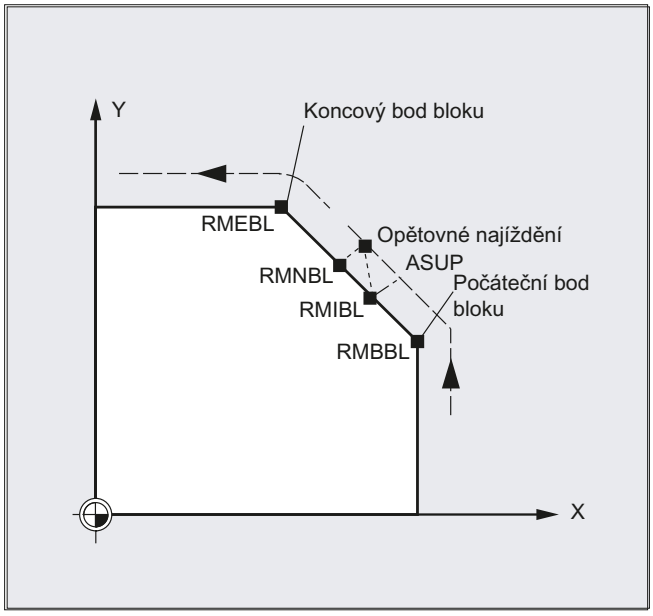
Poznámka

SERUPRO

Pro operaci SERUPRO jsou parametry RMIBL a RMBBL identické. Parametr RMNBL není omezen pouze na operaci SERUPRO, nýbrž platí všeobecně.

Najíždění před nejblíže ležící bod dráhy RMNBL

Pokud jde o okamžik interpretace příkazu REPOSA, po přerušení není blok pro opětovné najetí s příkazem RMNBL zahájen celý ještě jednou, nýbrž je opracovávána pouze zbytková dráha. Najíždí se na bod dráhy, který leží co možno nejblíže bloku, v němž došlo k přerušení.



Stavové informace pro platný režim REPOS

Platný režim funkce REPOS přerušeného bloku je možné načíst pomocí synchronních akcí prostřednictvím proměnné $AC_REPOS_PATH_MODE.

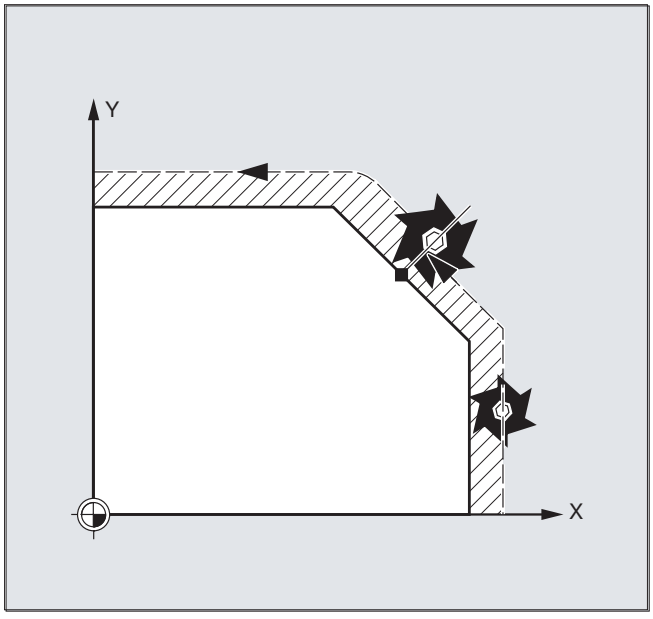
0 Najíždění není definováno
1 RMBBL: Najíždění na začátek
2 RMIBL: Najíždění na místo, kde došlo k přerušení
3 RMEBL: Najíždění na koncový bod bloku
4 RMNBL: Najíždění na bod dráhy, který leží nejblíže bloku, kde došlo k přerušení

Najíždění s novým nástrojem

Jestliže jste zpracovávání programu přerušili kvůli zlomení nástroje:

Naprogramováním nového D_čísla bude zpracování programu pokračovat od bodu opětovného najetí se změněnými hodnotami korekčních parametrů nástroje.

V případě změněných hodnot korekčních parametrů nástroje se může stát, že na místo, v němž došlo k přerušení, nebude možné najet. V tomto případě se bude najíždět na bod na nové kontuře, který leží co možno nejblíže bodu, v němž došlo k přerušení (v případě potřeby s úpravou podle parametru DISPR).



Najíždění na konturu

Pohyb, se kterým nástroj znovu najíždí na konturu, může být naprogramován. Pro adresy os, které se mají pohybovat, zadejte nulovou hodnotu.

Pomocí příkazů REPOSA, REPOSQA a REPOSHA jsou automaticky všechny osy nastavovány na původní pozici. Zadání os není zapotřebí.

Jestliže je naprogramován příkaz REPOSL, REPOSQ a REPOSH, pohybují se všechny geometrické osy automaticky, aniž by byly v příkazu zadány. Všechny ostatní osy musí být v příkazu uvedenu.

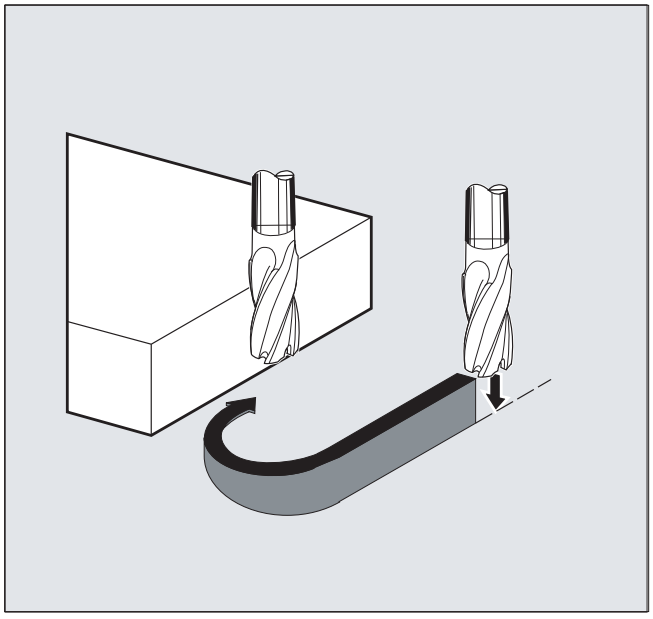
Pro kruhové pohyby s příkazy REPOSH a REPOSQ platí:

Kruhový pohyb se uskutečňuje v zadané pracovní rovině G17 až G19.

Jestliže je v najížděcím bloku uvedena třetí geometrická osa (směr přísuvu), potom se v případě, že si poloha nástroje a naprogramovaná poloha ve směru přísuvu neodpovídají, uskuteční zpětný pohyb na bod pro opětovné najetí po spirále.

V následujících případech dochází k automatickému přepnutí na lineární najíždění REPOSL:

- Nezadali jste žádnou hodnotu pro parametr DISR.
- Neexistuje žádný definovaný směr najíždění (přerušení programu v bloku bez informací o posuvech).
- V případě směru najíždění kolmo na aktuální pracovní rovinu.



4.14.11 Ovlivňování vedení pohybu

4.14.11.1 Přizpůsobení maximální rychlosti osy nebo otáček vřetena (VELOLIM)

Pomocí příkazu VELOLIM je možné ve výrobním programu snížit prostřednictvím strojního parametru nastavenou maximální možnou rychlost osy, příp. maximální možné otáčky vřetena závislé na stupni převodovky.

Platnost

Funkce VELOLIM se uplatňuje:

- v automatických provozních režimech
- na dráhové a polohovací osy
- na vřetena v režimu vřetena/osy

Syntaxe

```
VELOLIM[<osa>]=<hodnota>
```

Význam

VELOLIM:	Přizpůsobení mezní hodnoty rychlosti nebo otáček
<osa>:	Osa nebo vřeteno, jejichž mezní hodnota rychlosti nebo otáček má být přizpůsobena. Prostřednictvím parametru MD30455 $MA_MISC_FUNCTION_MASK, bit 6, může být nastaveno, zda má být funkce VELOLIM uplatňována nezávisle na momentálním použití jako vřeteno nebo jako osa (bit 6 = 1) nebo zda může být naprogramována odděleně pro každý provozní režim (bit 6 = 0). Jestliže je v konfiguraci nastaveno oddělené působení, je možné při programování pomocí identifikátorů vybírat z následujících možností: • Identifikátor vřetena pro provozní režimy vřetena: S<n> • Identifikátor osy pro režim osy, např.: C
<hodnota>:	Procentuální hodnota korekce U os, případně vřeten v režimu osy, pro které je nastaven MD30455 Bit 6 = 0, se hodnota korekce vztahuje na maximální rychlost osy nastavenou v konfiguraci (MD32000 $MA_MAX_AX_VELO). U vřeten v režimu vřetena nebo v režimu osy a při nastavení MD30455 Bit 6 = 1 se hodnota korekce vztahuje na maximální otáčky momentně aktivního stupně převodovky (MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[<n>]).
	Rozsah hodnot: 1 ... 100
	Hodnota 100 má za následek, že rychlost, příp. otáčky nejsou ovlivňovány.

Poznámka

Chování na konci výrobního programu a při resetu kanálu

Chování příkazu VELOLIM na konci výrobního programu a při resetu kanálu závisí na nastavení bitu 0 strojního parametru MD32320 $MA_DYN_LIMIT_RESET_MASK.

Poznámka

Omezení otáček v režimu vřetena

Omezení otáček pomocí příkazu VELOLIM (menší jak 100%) může být v režimu vřetena rozpoznáno prostřednictvím následujících systémových proměnných:

- $AC_SMAXVELO (maximální možné otáčky vřetena)
- $AC_SMAXVELO_INFO (identifikace příčiny omezení otáček)

Příklady

Příklad 1: Omezení rychlosti osy stroje

```
Programový kód              Komentář
...
N70 VELOLIM[X]=80           ; osové obrábění načisto ve směru osy X má být uskutečňováno s rychlostí odpovídající jen s maximálně 80% rychlosti přípustné pro osu
...
```

Příklad 2: Omezení otáček vřetena 1 (AX5)

Konfigurace:

- MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[1, AX5] = 1000
  (maximální otáčky stupně převodovky 1 = 1000 ot/min)
- MD30455 $MA_MISC_FUNCTION_MASK[AX5], Bit 6 = 1
  (naprogramování příkazu VELOLIM se uplatňuje společně pro režim osy a režim vřetena nezávisle na naprogramovaném identifikátoru)

Programování:

Programový kód	**Komentář**
`N05 VELOLIM[S1]=90`	`; omezení maximálních otáček vřetena 1 na 90% z 1000 ot/min`
`...`	
`N50 VELOLIM[C]=45`	`; omezení maximálních otáček vřetena 1 na 45% z 1000 ot/min, pokud C je osový identifikátor S1.`
`...`	

4.14.11.2 Přizpůsobení maximálního osového ryvu (JERKLIM)

Pomocí příkazu JERKLIM může být ve výrobním programu snížen nebo zvýšen prostřednictvím strojního parametru nastavený maximální možný ryv osy při pohybu po dráze v kritických úsecích programu.

Předpoklady

Musí být aktivní režim zrychlení SOFT.

Platnost

Funkce JERKLIM se uplatňuje:

- v automatických provozních režimech
- jen pro dráhové osy

Syntaxe

```
JERKLIM[<osa>]=<hodnota>
```

Význam

`JERKLIM`:	Přizpůsobení mezní hodnoty ryvu	
`<osa>`:	Osa stroje, jejíž mezní hodnota ryvu má být přizpůsobena	
`<hodnota>`:	Procentuální hodnota korekce Hodnota korekce se vztahuje na v konfiguraci nastavený maximální ryv osy při pohybu po dráze (MD32431 $MA_MAX_AX_JERK).	
	Rozsah hodnot:	1 ... 200
	Hodnota 100 má za následek žádné ovlivňování ryvu.	

Poznámka

Chování na konci výrobního programu a při resetu kanálu

Chování příkazu JERKLIM na konci výrobního programu a při resetu kanálu závisí na nastavení bitu 0 strojního parametru MD32320 $MA_DYN_LIMIT_RESET_MASK.

Příklad:

Programový kód	**Komentář**
`N1000 G0 X0 Y0 F10000 SOFT G64`	
`N1100 G1 X20 RNDM=5 ACC[X]=20 ACC[Y]=30`	
`N1200 G1 Y20 JERKLIM[Y]=200`	`; osové obrábění načisto ve směru osy Y může být uskutečňováno se zrychlením/ zpomalením odpovídajícím s maximálně 200% ryvu přípustného pro osu`
`N1300 G1 X0 JERKLIM[X]=2`	`; osové obrábění načisto ve směru osy X má být uskutečňováno se zrychlením/zpoma- lením odpovídajícím jen s max. 2% ryvu přípustného pro osu`
`N1400 G1 Y0`	
`M30`	

4.14.11.3 Přizpůsobení maximální rychlosti pohybu po dráze (FLIM)

Prostřednictvím funkce „Přizpůsobení maximální rychlosti pohybu po dráze" může být ve výrobním programu v jeho kritických úsecích omezena, příp. snížena, rychlost pohybu po dráze vyplývající z hodnot omezení pro jednotlivé osy. Hodnota rychlosti, na kterou má být maximální rychlost pohybu po dráze omezena, se programuje prostřednictvím adresy FLIM. Tato naprogramovaná hodnota je vždy v platnosti pouze jen do následujícího resetu NC systému nebo do konce výrobního programu.

Na posuv omezený příkazem FLIM se i nadále uplatňují korekce (override). Prostřednictvím ovladače Override se proto dá hodnota daná příkazem FLIM mírně překročit.

Platnost

Funkce se uplatňuje:

- v automatických provozních režimech
- jen pro dráhové osy
- jen tehdy, je-li v platnosti G94
- nikoli při rychlém posuvu

Poznámka

Funkce „Přizpůsobení maximální rychlosti pohybu po dráze“ je v platnosti pouze v kombinaci s lineárním posuvem G94. V kombinaci s jinými druhy posuvu (G93, G931, G95, G96, G97, G971, G972) není možné tuto funkci použít.

Syntaxe

```
...
FLIM=<hodnota>
...
FLIM=-1
...
```

Význam

FLIM	Adresa pro přizpůsobení maximální rychlosti pohybu po dráze	
<hodnota>	Hodnota rychlosti, na kterou má být maximální rychlost pohybu po dráze omezena	
	Datový typ:	REAL
	Rozsah hodnot:	$1{,}0 * 10^{-6}$ ... $1{,}0 * 10^{38}$
	Jednotka:	mm/min nebo palce/min (v závislosti na aktivním systému měřicích jednotek)
FLIM=-1	Omezení, které bylo naprogramováno prostřednictvím příkazu FLIM=<hodnota>, se opět deaktivuje.	

Příklad

Programový kód	**Komentář**
...	
N1000 G0 X0 Y0 F10000 G64 G710	
N1100 G1 X20 RNDM=5	
N1200 G1 Y20 **FLIM=5000**	; Pohyb při obrábění načisto ve směru osy Y probíhá s rychlostí max. 5 m/min.
N1300 G1 X0 Y40	; Pohyb po dráze při obrábění načisto ve směru osy X a osy Y probíhá s rychlostí max. 5 m/min.
N1400 G1 Y0 **FLIM=-1**	; Pohyb při obrábění načisto ve směru osy Y probíhá s rychlostí 10 m/min.
M30	

4.14.11.4 Přizpůsobení maximálního zrychlení při pohybu po dráze (PACCLIM)

Prostřednictvím funkce „Přizpůsobení maximálního zrychlení pohybu po dráze" může být ve výrobním programu v jeho kritických úsecích omezeno, příp. sníženo, zrychlení pohybu po dráze vyplývající z hodnot omezení pro jednotlivé osy. Hodnota zrychlení, na kterou má být maximální zrychlení pohybu po dráze sníženo, se programuje prostřednictvím adresy PACCLIM. Tato naprogramovaná hodnota je vždy v platnosti pouze jen do následujícího resetu NC systému nebo do konce výrobního programu.

Předpoklady

Správa licencí

Tato funkce je volitelným doplňkem, k jehož použití je nutno zakoupit licenci („Omezení zrychlení pohybu po dráze"; typové číslo: 6FC5800-0xP26-0YB0), která musí být prostřednictvím správy licencí přiřazena danému hardwaru.

Platnost

Funkce se uplatňuje:

- v automatických provozních režimech
- jen pro dráhové osy

Syntaxe

```
...
PACCLIM=<hodnota>
...
PACCLIM=-1
...
```

Význam

PACCLIM	Adresa pro přizpůsobení maximálního zrychlení pohybu po dráze	
<hodnota>	Hodnota zrychlení, na kterou má být maximální zrychlení pohybu po dráze sníženo	
	Datový typ:	REAL
	Rozsah hodnot:	$1{,}0 * 10^{-6}$ ... $1{,}0 * 10^{38}$
	Jednotka:	m/s^2 nebo $palce/s^2$ (v závislosti na aktivním systému měřicích jednotek)
PACCLIM=-1	Omezení, které bylo naprogramováno prostřednictvím příkazu PACCLIM=<hodnota>, se opět deaktivuje.	

Poznámka

Pokud jde o funkce, podobá se příkaz PACCLIM nastavovanému parametru SD42500 $SC_SD_MAX_PATH_ACCEL (maximální zrychlení při pohybu po dráze). Na rozdíl od parametru SD42500 však příkaz PACCLIM pracuje blokově synchronním způsobem.

Poznámka

Při výpočtu hodnoty omezení se hodnota z parametru SD42500 $SC_SD_MAX_PATH_ACCEL zohledňuje jen tehdy, pokud je parametr SD42502 $SC_IS_SD_MAX_PATH_ACCEL nastaven na hodnotu „TRUE“.

Jestliže jsou aktivní jak parametr PACCLIM, tak také parametr SD42500 $SC_SD_MAX_PATH_ACCEL, je v platnosti menší z obou omezujících hodnot.

Příklad

Programový kód	**Komentář**
...	
N1000 G0 X0 Y0 F10000 G64 G710	
N1100 G1 X20 RNDM=5	
N1200 G1 Y20 **PACCLIM=0.5**	; Pohyb při obrábění načisto ve směru osy Y je zrychlen/zpomalen na hodnotu max. 0,5 m/s2.
N1300 G1 X0 Y40	; Pohyb po dráze při obrábění načisto ve směru osy X a osy Y je zrychlen/zpomalen na hodnotu max. 0,5 m/s2.
N1400 G1 Y0	
M30	

4.14.12 Chování při přechodu na další blok, když je aktivní vazba (CPBC)

Pomocí příkazu CPBC se zadává kritérium pro přechod na další blok, které musí být splněno, aby se ve výrobním programu s aktivní vazbou uskutečnil přechod na další blok.

Syntaxe

```
CPBC[<vlečná osa >] = <Kritérium>
```

Význam

CPBC:	Kritérium pro přechod na další blok, když je aktivní vazba
<vlečná osa>:	Identifikátor vlečné osy

<Kritérium>:	Kritérium přechodu na další blok	
	Typ:	STRING
	Hodnota	**Význam:** Přechod na další blok se uskuteční
	"NOC"	Nezávisle na stavu vazby
	"IPOSTOP"	Při synchronním zpracování na základě žádané hodnoty
	"COARSE"	Při synchronním zpracování na základě "hrubé" skutečné hodnoty
	"FINE"	Při synchronním zpracování na základě "jemné" skutečné hodnoty

Příklad

Programový kód

```
; Na další blok se přechází, když:
;  - Vazba na vlečnou osu X2 == aktivní
;  - Synchronní zpracování na základě žádané hodnoty ==
aktivní
CPBC[X2]="IPOSTOP"
```

4.15 Funkce pro osy

4.15.1 Výměna osy, výměna vřetena (RELEASE, GET, GETD)

Interpolace jedné nebo více os, příp. vřeten se smí uskutečňovat vždy jen v jednom kanálu. Pokud musí osa pracovat ve dvou různých kanálech střídavě (např. podavač pro výměnu palet), potom tato osa musí být napřed v aktuálním kanálu uvolněna a pak v jiném kanálu převzata. Kanály si osu vyměňují.

Rozšíření výměny os

Osa/vřeteno mohou být mezi vedlejším a hlavním programem vyměněny se zastavením předběžného zpracování a se synchronizací nebo alternativně i bez zastavení předběžného zpracování. Kromě toho je výměna os možná také pomocí následujících operací:

- Rotace zásobníku os AXCTSWE příp. AXCTWED pomocí implicitních příkazů `GET/GETD`.
- Frame s otočením, jestliže je tato osa spojena vazbou s jinou osou.
- Synchronní akce, viz "Pohybové synchronní akce", "Výměna os `RELEASE`, `GET`".

Výrobce stroje

Věnujte prosím pozornost informacím od výrobce stroje. Prostřednictvím v konfiguraci nastavitelných strojních parametrů musí být osa pro výměnu os ve všech kanálech jednoznačně definována a chování při výměně os může být pomocí strojních parametrů nastaveno tak, aby je bylo možné změnit.

Syntaxe

`RELEASE(název osy, název osy, ...)` nebo `RELEASE(S1)`

`GET(název osy, název osy, ...)` nebo `GET(S2)`

`GETD(název osy, název osy, ...)` nebo `GETD(S3)`

Pomocí příkazu GED (GET Directly) se osa přímo vyzvedne z nějakého jiného kanálu. To znamená, že k tomuto příkazu GETD nemusí být naprogramován žádný odpovídající příkaz REELASE v nějakém jiném kanálu. Znamená to ale také, že nyní musí být s oním kanálem navázána komunikace (např. značky pro čekání).

Význam

`RELEASE (název osy, název osy, ...):`	Uvolnění jedné nebo více os
`GET (název osy, název osy, ...):`	Převzetí jedné nebo více os
`GETD (název osy, název osy, ...):`	Přímé převzetí jedné nebo více os
`název osy:`	Přiřazení osy v systému: AX1, AX2, ... nebo zadání názvů os stroje
`RELEASE(S1):`	Uvolnění vřetena S1, S2, ...
`GET(S2):`	Převzetí vřetena S1, S2, ...
`GETD(S3):`	Přímé převzetí vřetena S1, S2, ...

Požadavek GET bez zastavení předběžného zpracování

Pokud je po požadavku GET **bez** zastavení předběžného zpracování osa opět uvolněna některým z příkazů `RELEASE(osa)` nebo `WAITP(osa)`, má následující příkaz `GET` za následek `GET` **se** zastavením předběžného zpracování.

⚠ POZOR

Přiřazení os změněno

Osa, příp. vřeteno, které byly příkazem GET převzaty, zůstávají tomuto kanálu přiřazeny i po RESETU, ať už byl vyvolán tlačítkem nebo programem.

Jestliže je osa zapotřebí ve svém základním kanálu, při spuštění nového programu se přiřazení vyměňovaných os, příp. vřeten musí programově-technicky ošetřit.

Při vypnutí a zapnutí systému (Power-On) jsou osy a vřetena přiřazena kanálům podle definice ve strojních parametrech.

Příklady

Příklad 1: Výměna os mezi dvěma kanály

Ze 6 os jsou v kanálu 1 pro obrábění používány: 1., 2., 3. a 4. osa.
5. a 6. osa jsou používány v kanálu 2 pro výměnu obrobku.

Osu 2 má být možno vyměňovat mezi oběma kanály a po zapnutí systému (Power-On) má být tato osa přiřazena kanálu 1.

Program "MAIN" v kanálu 1:

Programový kód	**Komentář**
INIT (2,"TAUSCH2")	; Aktivování programu TAUSCH2 v kanálu 2.
N… START (2)	; Spuštění programu v kanálu 2.
N… GET (AX2)	; Převzít osu AX2.
...	
N… RELEASE (AX2)	; Uvolnit osu AX2.
N… WAITM (1,1,2)	; Čekání na značku WAIT v kanálu 1 a 2 kvůli synchronizaci v obou kanálech.
...	; Další postup po výměně osy.
N… M30	

Program "TAUSCH2" v kanálu 2:

Programování	**Komentář**
N… RELEASE (AX2)	
N160 WAITM(1,1,2)	; Čekání na značku WAIT v kanálu 1 a 2 kvůli synchronizaci v obou kanálech.
N150 GET(AX2)	; Převzít osu AX2.
...	; Další postup po výměně osy.
N… M30	

Příklad 2: Výměna osy bez synchronizace

Jestliže osa nemusí být synchronizována, není příkazem GET generováno žádné zastavení předběžného zpracování.

Programování	**Komentář**
N01 G0 X0	
N02 RELEASE(AX5)	
N03 G64 X10	
N04 X20	
N05 GET(AX5)	; Jestliže není zapotřebí žádná synchronizace, nejedná se o zpracovatelný blok.
N06 G01 F5000	; Žádný zpracovatelný blok.
N07 X20	; Žádný zpracovatelný blok, protože X má stejnou pozici jako v bloku N04.
N08 X30	; První zpracovatelný blok po bloku N05.
...	

Příklad 3: Aktivování výměny os bez zastavení předběžného zpracování

Předpoklad: Výměna os bez zastavení předběžného zpracování musí být nastavena v konfiguraci pomocí strojního parametru.

Programování	Komentář
N010 M4 S100	
N011 G4 F2	
N020 M5	
N021 SPOS=0	
N022 POS[B]=1	
N023 WAITP(B)	; Osa B je nyní neutrální osou.
N030 X1 F10	
N031 X100 F500	
N032 X200	
N040 M3 S500	; Osa nespouští zastavení předběžného zpracování/ příkaz REORG.
N041 G4 F2	
N050 M5	
N099 M30	

Jestliže se vřeteno, příp. osa B bezprostředně po bloku N023 jako **osa PLC** pohybuje na pozici 180 stupňů a zpět na pozici 1 stupeň, potom se tato osa opět stane neutrální osou a v bloku `N40` nebude spouštět žádné zastavení předběžného zpracování.

Další informace

Předpoklady pro výměnu osy

- Prostřednictvím strojních parametrů musí být osa definována ve všech kanálech, ve kterých se používá.
- Pomocí strojního parametru pro specifickou **osu** musí být definováno, kterému kanálu má být osa po zapnutí systému (Power-On) přiřazena.

Popis

Uvolnění osy: RELEASE

Při uvolnění osy je potřeba dávat pozor na následující záležitosti:

1. Osa se nesmí podílet na žádné transformaci.
2. V případě spojení os (tangenciální řízení) musí být všechny do svazku seskupené osy uvolněny.
3. Konkurenční polohovací osa nemůže být v tomto stavu vyměňována.
4. V případě řídící osy gantry jsou vyměněny také všechny vlečné osy.
5. V případě vazeb mezi osami (vlečení, spojení řídící hodnotou, elektronická převodovka) může být uvolněna pouze řídící osa vazby.

Převzetí osy: GET

Pomocí tohoto příkazu se uskutečňuje vlastní výměna osy. Veškerá odpovědnost za osu se tím přenáší na kanál, v němž byl příkaz naprogramován.

Důsledky použití příkazu GET:

Výměna osy se synchronizací:

Osa musí být synchronizována vždy, když byla v mezičase přiřazena v nějakém jiném kanálu nebo v PLC a pokud před příkazem GET nebyla uskutečněna žádná synchronizace pomocí příkazu "WAITP", G74 nebo vymazáním zbytkové dráhy.

- Předběžné zpracování se zastaví (stejně jako při příkazu STOPRE).
- Zpracování bude pozastaveno tak dlouho, dokud není výměna úplně dokončena.

Automatický příkaz "GET"

Jestliže je osa v principu k dispozici v nějakém kanálu, v daném okamžiku ale není "kanálovou osou", provede se příkaz "GET" automaticky. Jestliže už jsou osy synchronizovány, k zastavení předběžného zpracování nedochází.

Nastavení měnitelného chování při výměně osy

Okamžik předání osy může být nastaven pomocí strojního parametru, a to následujícím způsobem:

- Automatická výměna osy mezi dvěma kanály se uskuteční také tehdy, pokud byla osa pomocí příkazu WAITP uvedena do neutrálního stavu (stejné chování jak bylo popsáno výše).
- V případě požadavku na rotaci zásobníku os jsou všechny osy zásobníku os, které jsou přiřazeny uvedenému kanálu, převedeny pomocí implicitních příkazů GET, příp. GETD do daného kanálu. Následující výměna osy je znovu umožněna teprve po dokončení rotace zásobníku os.
- Po posunutém pomocném bloku v hlavní větvi programu se zkontroluje, zda je či není nezbytná reorganizace. Jedině v případě, že stavy os v tomto bloku **neodpovídají** momentálním stavům os, je reorganizace potřebná.
- Místo bloku s příkazem GET se zastavením předběžného zpracování a se synchronizací mezi vedlejším a hlavním programem mohou být osy vyměněny bez zastavení předběžného zpracování. Potom je generován jen jeden pomocný blok s příkazem GET. V hlavním programu se při zpracovávání tohoto bloku kontroluje, zda se stavy os v bloku shodují s aktuálními stavy os.

4.15.2 Předávání osy jinému kanálu (AXTOCHAN)

Pomocí příkazu NC-jazyka `AXTOCHAN` může být pro určitou osu přikázáno, aby se tato osa předala do jiného kanálu. Osa může být do odpovídajícího kanálu přenesena jak z výrobního NC programu, tak i ze synchronní akce.

Syntaxe

```
AXTOCHAN(název osy,číslo kanálu[,název osy,číslo kanálu[,...]])
```

Význam

Prvek	Popis
`AXTOCHAN`:	Vyžádání osy pro určitý kanál
`název osy`:	Přiřazení osy v systému: X, Y, ... nebo zadání názvů os stroje podílejících se na operaci. Jako kanál, který je v příkazu uveden, nesmí být veden vlastní kanál a nesmí se jednat ani o kanál, který v daném okamžiku disponuje interpolačními oprávněními pro danou osu.
`číslo kanálu`:	Číslo kanálu, jemuž má být v dané chvíli osa přiřazena.

Poznámka

Konkurenční polohovací osa a osa ovládaná výlučně prostřednictvím PLC

Osa PLC nesmí být jako konkurenční polohovací osa v kanálu vyměňována. Osa, která je ovládána výlučně prostřednictvím PLC, nesmí být NC programu přiřazena.

Další informace: Příručka k funkcím, Osy a vřetena

Příklad:

Příkaz AXTOCHAN v NC programu

Osy X a Y jsou známy v kanálu 1 a v kanálu 2. Momentálně disponuje interpolačními oprávněními kanál 1 a v kanálu 1 je spuštěn následující program:

```
Programový kód                  Komentář
N110 AXTOCHAN(Y,2)              ; Osa Y je přesunuta do kanálu 2.
N111 M0
N120 AXTOCHAN(Y,1)              ; osa Y je přenesena opět zpátky (neutrální stav)
N121 M0
N130 AXTOCHAN(Y,2,X,2)          ; Osa Y a osa X jsou přesunuty do kanálu 2 (osy jsou
                                neutrální).
N131 M0
N140 AXTOCHAN(Y,2)              ; Osa Y je přesunuta do kanálu 2 (NC program).
N141 M0
```

Další informace

Příkaz AXTOCHAN v NC programu

V tomto případě se příkaz `GET` uskutečňuje a tím pádem se také čeká na skutečnou změnu stavu jen tehdy, vyskytne-li se požadavek na tuto osu pro NC program ve vlastním kanálu. Pokud je osa vyžádána pro jiný kanál nebo jestliže má být ve vlastním kanálu neutrální osou, potom je požadavek odpovídajícím způsobem pouze odložen.

Příkaz AXTOCHAN ze synchronní akce

Jestliže je osa vyžádána pro vlastní kanál, potom se příkaz `AXTOCHAN` ze synchronní akce převádí na příkaz `GET` ze synchronní akce. V tomto případě se osa v rámci prvního vyžádání pro vlastní kanál stává neutrální osou. V případě druhého vyžádání osy z NC programu je osa přiřazena analogicky k příkazu GET v NC programu. Pokud budete potřebovat informace k příkazu GET v synchronních akcích, viz kapitola "Pohybové synchronní akce".

4.15.3 Osové funkce (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL)

Příkaz „AXNAME" se používá např. při sestavování všeobecně platných cyklů, pokud názvy os nejsou známy.

Příkaz „AX" se používá pro nepřímé programování geometrických a synchronních os. Identifikátor osy je přitom uložen v proměnné typu AXIS nebo je zjištěn pomocí příkazu, jako např. „AXNAME" nebo „SPI".

Příkaz „SPI" se použije, jestliže se programují funkce osy pro vřeteno, např. pro synchronní vřeteno.

Příkaz „AXTOSPI" slouží v situacích, kdy potřebujete identifikátor osy převést na index vřetena (inverzní funkce k „SPI").

Příkaz „AXSTRING" slouží v situacích, kdy potřebujete identifikátor osy (datový typ AXIS) převést na řetězec (inverzní funkce k „AXNAME").

Příkaz „ISAXIS" se používá ve všeobecně platných cyklech, aby se zjistilo, že určitá geometrická osa je k dispozici a aby následující volání proměnné $P_AXNX nebylo přerušeno s chybou.

Funkce „MODAXVAL" se používá pro zjišťování polohy kruhových os typu Modulo po zpracování funkcí Modulo.

Syntaxe

```
AXNAME("řetězec")
AX[AXNAME("řetězec")]
SPI (n)

AXTOSPI(A) nebo AXTOSPI(B) nebo AXTOSPI(C)
AXSTRING(SPI(n))
ISAXIS(<číslo geometrické osy>)
<pozice modulo>=MODAXVAL(<osa>,<poloha osy>)
```

Význam

`AXNAME`:	Převádí vstupní řetězec na identifikátor osy, vstupní řetězec musí obsahovat platný název osy.
`AX`:	Proměnný identifikátor osy
`SPI`:	Převádí číslo vřetena na identifikátor osy; předávaný parametr musí obsahovat platné číslo vřetena.
`n`:	Číslo vřetena
`AXTOSPI`:	Převádí identifikátor osy na index vřetena typu INTEGER. Příkaz "AXTOSPI" odpovídá inverzní funkci k příkazu „SPI".

X, Y, Z:	Identifikátor osy typu AXIS jako proměnná nebo jako konstanta.
AXSTRING:	Na výstupu bude řetězec, jemuž je přiřazeno číslo vřetena.
ISAXIS:	Zkontroluje, zda je uvedená geometrická osa k dispozici.
MODAXVAL:	Zjišťuje polohu kruhových os typu Modulo po zpracování funkcí Modulo, tato hodnota odpovídá zbytku po celočíselném dělení vzhledem k rozsahu Modulo (v případě standardního nastavení je to 0° až 360°, pomocí strojních parametrů MD30340 MODULO_RANGE_START a MD30330 $MA_MODULO_RANGE mohou být začátek s velikost rozsahu Modulo změněny).

Poznámka

Rozšíření SPI

Osová funkce SPI(n) m§že být použita také pro čtení a zápis komponentů framu. Tímto způsobem je možné zapisovat údaje do framu, např. pomocí syntaxe `$P_PFRAME[SPI(1),TR]=2.22`.

Prostřednictvím doplňkového naprogramování pozice os pomocí adresy `AX[SPI(1)]=<pozice osy>` je možné osou pohybovat. Předpokladem pro tuto operaci ale je, že se vřeteno nachází v režimu polohování nebo v režimu osy.

Příklady

Příklad 1: AXNAME, AX, ISAXIS

```
Programový kód                                  Komentář
OVRA[AXNAME("Planachse")]=10                    ; korekce (override) pro příčnou osu
AX[AXNAME("Planachse")]=50.2                    ; koncová pozice pro příčnou osu
OVRA[SPI(1)]=70                                 ; korekce (override) pro vřeteno 1
AX[SPI(1)]=180                                  ; koncová pozice pro vřeteno 1
IF ISAXIS(1) == FALSE GOTOF WEITER              ; Je k dispozici abscisa?
AX[$P_AXN1]=100                                 ; pohyb abscisy
WEITER:
```

Příklad 2: AXSTRING

Při programování příkazu AXSTRING[SPI(n)] se nezadává index osy pro osu, která je přiřazena vřetenu, jako číslo vřetena, nýbrž místo toho se zadává řetězec "Sn".

```
Programový kód                  Komentář
AXSTRING[SPI(2)]                ; na výstupu bude řetězec "S2"
```

Příklad 3: MODAXVAL

Tímto způsobem se má zjistit poloha kruhové osy A typu Modulo po zpracování funkcí modulo.

Výchozí hodnota pro výpočet je poloha osy 372,55.

Rozsah Modulo nastavený v parametrech je 0 až 360 stupňů:

MD30340 MODULO_RANGE_START = 0

MD30330 $MA_MODULO_RANGE = 360

Programový kód	**Komentář**
`R10=MODAXVAL(A,372.55)`	`; vypočítaná pozice Modulo je R10 = 12,55`

Příklad 4: MODAXVAL

Jestliže se naprogramovaný identifikátor osy nevztahuje na kruhovou osu typu Modulo, potom bude převáděná hodnota (<poloha osy>) vrácena zpět nezměněná.

Programový kód	**Komentář**
`R11=MODAXVAL(X,372.55)`	`; X je lineární osa; R11 = 372,55`

4.15.4 Přepínatelné geometrické osy (GEOAX)

Pomocí funkce "Přepínatelné geometrické osy" můžete geometrické osy nastavené v konfiguraci pomocí strojních parametrů vyměnit za jiné kanálové osy.

Syntaxe

```
GEOAX(<n>,<kanálová osa>,<n>,<kanálová osa>,<n>,<kanálová osa>)
GEOAX()
```

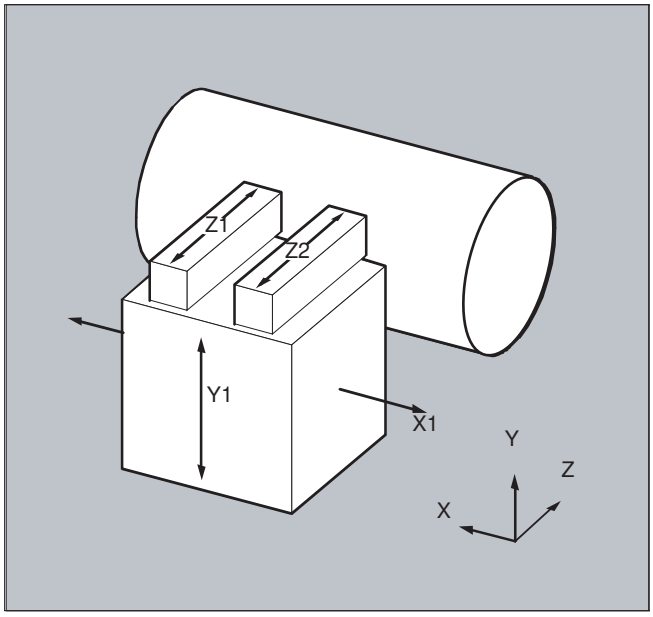
Význam

`GEOAX(...)`	Funkce pro přepnutí geometrických os. **Poznámka**: `GEOAX)` bez udání parametrů znovu aktivuje základní konfiguraci geometrických os nastavenou ve strojních parametrech.
`<n>`	Číslo geometrické osy, která má být nahrazena za uvedenou kanálovou osu. Rozsah hodnot: 0, 1, 2, 3 **Poznámka**: 0: Uvedená kanálová osa bude bez náhrady odstraněna ze svazku geometrických os 1: 1. Geometrická osa ≙ Souřadná osa X (abscisa) WCS 2: 2. Geometrická osa ≙ Souřadná osa Y (ordináta) WCS 3: 3. Geometrická osa ≙ Souřadná osa Z (aplikáta) WCS
`<kanálová osa>`	Název kanálové osy, která má být přijata do svazku geometrických os.

Příklady

Příklad 1: Střídavé přepínání dvou os na místě jedné geometrické osy

Nástrojovými saněmi je možné pohybovat pomocí kanálových os X1, Y1, Z1 a Z2.



Konfigurace geometrických os je taková, že po zapnutí systému se napřed jako platnou 3. geometrickou osou pod názvem geometrické osy "Z" stává osa Z1 a spolu s osami X1 a Y1 vytváří svazek geometrických os.

Ve výrobním programu se nyní mají osy Z1 a Z2 střídavě používat jako geometrická osa Z.

Programový kód	**Komentář**
...	
N100 GEOAX(3,Z2)	; jako 3. geometrická osa (Z) funguje kanálová osa Z2
N110 G1 ...	
N120 GEOAX(3,Z1)	; jako 3. geometrická osa (Z) funguje kanálová osa Z1
...	

Příklad 2: Přepínání geometrických os v případě 6 kanálových os

Stroj je vybaven 6 kanálovými osami s názvy XX, YY, ZZ, U, V, W.

Základní nastavení konfigurace geometrických os prostřednictvím strojních parametrů je:

Kanálová osa XX = 1. geometrická osa (osa X)

Kanálová osa YY = 2. geometrická osa (osa Y)

Kanálová osa ZZ = 3. geometrická osa (osa Z)

Programový kód	**Komentář**
N10 GEOAX()	; základní konfigurace geometrických je v platnosti
N20 G0 X0 Y0 Z0 U0 V0 W0	; všechny osy rychlým posuvem na pozici 0
N30 GEOAX(1,U,2,V,3,W)	; přiřazení kanálové osy U první (X), osy V druhé (Y)
	; a osy W třetí geometrické ose (Z)
N40 GEOAX(1,XX,3,ZZ)	; Přiřazení kanálové osy XX první (X) a osy ZZ třetí
	; geometrické ose (Z). Kanálová osa V zůstává druhou
	; geometrickou osou (Y).

Programový kód	**Komentář**
N50 G17 G2 X20 I10 F1000	; Plná kružnice v rovině X/Y. Pohybují se
	; kanálové osy XX a V.
N60 GEOAX(2,W)	; kanálová osa W se stává druhou geometrickou osou (Y)
N80 G17 G2 X20 I10 F1000	; Plná kružnice v rovině X/Y. Pohybují se
	; kanálové osy XX a W.
N90 GEOAX()	; obnovení původního stavu
N100 GEOAX(1,U,2,V,3,W)	; přiřazení kanálové osy U první (X), osy V druhé
	; (Y) a osy W třetí geometrické ose (Z)
N110 G1 X10 Y10 Z10 XX=25	; Kanálové osy U, V a W najíždějí na své
	; pozice 10. Osa XX jako přísuvná osa najíždí na pozici 25.
N120 GEOAX(0,V)	; osa V je vyjmuta ze svazku geometrických os
	; osy U a W jsou nadále první (X) a třetí
	; geometrickou osou (Z)
	; druhá geometrická osa (Y) zůstává neobsazena
N130 GEOAX(1,U,2,V,3,W)	; kanálová osa U zůstává přiřazena první (X), osa V
	; druhé (Y) a osa W třetí geometrické ose (Z)
N140 GEOAX(3,V)	; osa V se stává třetí geometrickou osou (Z), přičemž přepisuje osu W
	; a tím pádem ji vyjímá
	; ze svazku geometrických os ; druhá geometrická osa (Y)
	; je stejně jako předtím neobsazena.

Strojní parametry

Konfigurace osy

Přiřazení geometrických, doplňkových a strojních os kanálovým osám:

- 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

Chování při resetu

Chování změněného přiřazení geometrických os při resetu:

- MD20110 $MC_RESET_MODE_MASK, Bit 12
- MD20118 $MC_GEOAX_CHANGE_RESET

Chování při spuštění NC systému

- MD20112 $MC_START_MODE_MASK, Bit 12

Předávání uživatelskému programu PLC

Možnost nastavení parametrů příkazu M-funkce, která je vysílána na rozhraní NC-systému/PLC při přepnutí geometrických os:

- MD22532 $MC_GEOAX_CHANGE_M_CODE

Okrajové podmínky

Žádné přepínání geometrických os

- Pokud je aktivní některá z následujících funkcí, přepínání geometrických os není možné:
  - Transformace
  - Interpolace spliny
  - Korekce rádiusu nástroje
  - Jemná korekce nástroje
- Geometrická osa a další kanálová osa mají stejný Název.
- Jedna z os, které se podílejí na operaci přepnutí geometrických os, je zapojena do akce, která svým trváním přesahuje hranice bloku. Např. polohovací osa, jejíž pohyb trvá přes hranice bloku, nebo vlečná osa začleněná do vazby mezi osami.

Kruhové osy

Kruhové osy se nemohou stát geometrickými osami.

Stav os po nahrazení

Osa, která byla ve svazku geometrických os nahrazena pomocí přepnutí, může být po přepnutí naprogramována jako doplňková osa pomocí svého kanálového názvu.

Framy, chráněné oblasti, ohraničení pracovního pole

Přepnutím geometrických os jsou všechny framy, chráněné oblasti a ohraničení pracovního pole vymazány.

Polární souřadnice

Výměna geometrických os pomocí příkazu `GEOAX` nastavuje analogicky se změnou roviny pomocí příkazů `G17`-`G19` modální polární souřadnice na hodnotu 0.

DRF, NPV

Případná posunutí pomocí ručního kolečka (DRF) nebo externí posunutí počátku (NPV) zůstávají i po přepnutí v platnosti.

Základní konfigurace geometrických os

Příkaz `GEOAX()` vyvolá základní konfiguraci svazku geometrických os.

Po náběhu systému (Power-On) a při přepnutí do provozního režimu "Najíždění na referenční bod" se automaticky nastavuje základní konfigurace.

Korekce délky nástroje

Aktivní korekce délky nástroje je v platnosti i po operaci přepnutí. Platí však pro geometrické osy, které byly nově připojeny, příp. které si vyměnili pozici, jako hodnota, na kterou se zatím nenajelo. Při prvním pohybovém příkazu pro tyto geometrické osy vznikne výsledná dráha posuvu, která je odpovídajícím způsobem složena z korekce délky nástroje a naprogramované dráhy posuvu.

Geometrické osy, které při přepnutí své pozice zůstaly ve svazku os zachovány, si ponechávají i svůj stav týkající se korekce délky nástroje.

Konfigurace geometrických os při aktivní transformaci

- Konfigurace geometrických os, která je pro aktivní transformaci definována pomocí strojních parametrů transformací, nemůže být pomocí funkce "Přepínatelné geometrické osy" změněna.
- Pro odlišné konfigurace geometrických os pro transformace musí být ve strojních parametrech transformací vytvořeny různé datové bloky.
- Konfigurace geometrických os změněná pomocí funkce `GEOAX` se aktivováním transformace vymaže.
- Pokud jde o geometrické osy, mají nastavení parametrů těchto geometrických os vztahující se ke specifické aktivní transformaci přednost před nastaveními parametrů, která se týkají přepínání geometrických os.
  Příklad: Je aktivní transformace. Podle strojních parametrů má tato transformace při resetu kanálu zůstat zachována. Současně však ale při resetu kanálu má být obnovena původní základní konfigurace geometrických os. Zůstane zachována konfigurace geometrických os, která byla definována pro transformaci.
- Po deaktivování transformace bude opět v platnosti základní nastavení konfigurace geometrických os stanovené v parametrech.

Provozní režim JOG, funkce stroje REF

Když přepnete do provozního režimu JOG a spustíte strojní funkci REF (najíždění na referenční bod), vstoupí v platnost konfigurace geometrických os nastavená ve strojních parametrech.

4.15.5 Osový zásobník (AXCTSWE, AXCTSWED, AXCTSWEC)

Příkaz „AXCTSWE“, příp. „AXCTSWED“ provádí odblokování otáčení uvedeného osového zásobníku.

Pomocí příkazu „AXCTSWEC“ může být již aktivované uvolnění pro otáčení osového zásobníku opět odvoláno.

Syntaxe

```
AXCTSWE(<ID>)
AXCTSWED(<ID>)
AXCTSWEC(<ID>)
```

Význam

AXCTSWE:	Uvolnění otáčení osového zásobníku Zpracovávání programu není příkazem „AXCTSWE" pozastaveno. Otočení se uskuteční, jakmile je všem kanálům podílejícím se na osovém zásobníku poskytnuto uvolnění.	
AXCTSWED:	Uvolnění osového zásobníku pro otáčení bez ohledu na jiné kanály začleněné do osového zásobníku **Upozornění** • Varianta příkazu určená pro zjednodušení uvádění do provozu výrobního programu, příp. synchronní akce. • Chování v souvislosti s jinými kanály začleněnými do osového zásobníku může být předem nastaveno pomocí strojního parametru: MD12760 $MN_ AXCT_FUNCTION_MASK, Bit 0	
AXCTSWEC:	Deaktivování uvolnění otáčení osového zásobníku **Upozornění:** Uvolnění osového zásobníku za účelem otáčení může být deaktivováno jen tehdy, pokud toto otáčení ještě nezačalo. $AN_AXCTSWA[<osový zásobník>] == 0 Pokud potřebujete informace o systémových proměnných, viz "Osový zásobník (AXCTSWE, AXCTSWED, AXCTSWEC) (Strana 893)"	
<ID>:	Identifikátor osového zásobníku nebo osy ze zásobníku:	
	CT<Číslo>:	Předdefinovaný identifikátor osového zásobníku: MD12750 $MN_AXCT_NAME_TAB Příklad: "CT1"
	<Zásobník>:	Uživatelský identifikátor osového zásobníku: MD12750 $MN_AXCT_NAME_TAB Příklad: "CONTAINER_1"
	<osa>:	Identifikátor osy ze zásobníku, která je v kanálu známá

Poznámka

Velikosti kroku

Velikost kroku při otáčení osového zásobníku se nastavuje pomocí nastavovaného parametru:

SD41700 $SN_AXCT_SWWIDTH

Další informace

Diagnostika

Aktuální stavové informace o osovém zásobníku je možné načítat pomocí následující systémové proměnné:

Systémové proměnné	Typ	Popis
$AC_AXCTSWA[<Název>]	BOOL	Status osového zásobníku specifického kanálu
$AN_AXCTSWA[<osový zásobník>]	BOOL	Status osového zásobníku specifické NCU

Systémové proměnné	Typ	Popis
$AN_AXCTSWE[<osový zásobník>]	INT	Status otáčení osového zásobníku specifického slotu Tato systémová proměnná poskytuje informace o stavu slotů osového zásobníku prostřednictvím jednotlivých **bitů**. Každý bit odpovídá jednomu slotu.
$AN_AXCTAS[<osový zásobník>]	INT	Počet míst (slotů), o které byl osový zásobník momentálně pootočený.

Otočení osového zásobníku s implicitními příkazy GET/GETD

Prostřednictvím následujícího strojního parametru může být nastaveno, že pomocí příkazu „AXCTSWE“ jsou všechny zásobníkové osy kanálu převedeny prostřednictvím implicitních příkazů "GET“ / „GETD" do daného kanálu. Výměna osy je opět možná teprve po úspěšném otočení zásobníku.

MD10722 $MN_AXCHANGE_MASK, Bit 1 = 1

Poznámka

Otočení osového zásobníku pomocí implicitních příkazů „GET“ / „GETD“ **nebude** pro osu nacházející se ve stavu "osy hlavního zpracování" (např. pro osu PLC) provedeno, protože osa by potom musela tento stav kvůli otáčení osového zásobníku opustit.

4.15.6 Čekání na platnou pozici osy (WAITENC)

Pomocí příkazu NC jazyka „WAITENC“ je možné v NC programu počkat, dokud pro osy nastavené v konfiguraci pomocí parametru MD34800 $MA_WAIT_ENC_VALID = 1 nebude k dispozici synchronizovaná, příp. obnovená poloha osy.

V čekacím stavu může dojít k přerušení, např. v důsledku spuštění programu ASUP nebo přepnutím do provozního režimu JOG. Při pokračování programu bude čekací stav v případě potřeby opět obnoven.

Poznámka

Na uživatelském rozhraní se čekací stav zobrazuje prostřednictvím stavového hlášení o čekání "Čekání na měřicí systém".

Syntaxe

Příkaz „WAITENC“ může být naprogramován v programové části libovolného NC programu.

Programování se musí provádět v samostatném bloku:

```
...
WAITENC
...
```

Příklad:

Příkaz „WAITENC" se používá např. v uživatelském programu .../_N_CMA_DIR/_N_PROG_EVENT_SPF řízeném prostřednictvím událostí, jak ukazuje následující příklad použití.

Příklad použití: Zpětný pohyb nástroje po POWER-OFF s transformací orientace

Obrábění s orientací nástroje bylo v důsledku výpadku proudu přerušeno.
Při následujícím opětovném náběhu systému se vyvolává uživatelský program .../_N_CMA_DIR/_N_PROG_EVENT_SPF řízený na základě událostí.

V uživatelském programu řízeném na základě událostí se pomocí příkazu „WAITENC" čeká na synchronizovanou, příp. obnovenou polohu os, aby potom bylo možné vypočítat frame, který nasměruje WCS podle směru nástroje.

```
Programový kód                  Komentář
...
IF $P_PROG_EVENT == 4           ; náběh systému
    IF $P_TRAFO <> 0            ; transformace byla deaktivována
        WAITENC                 ; čekání na platné pozice orientačních os
        TOROTZ                  ; otočení osy Z systému WCS ve směru osy nástroje
    ENDIF
    M17
ENDIF
...
```

Potom může nástroj v provozním režimu JOG volně vyjet pomocí zpětného pohybu ve směru osy nástroje.

4.15.7 Programovatelné přepnutí sady parametrů (SCPARA)

Pomocí příkazu SCPARA je možno si pro danou osu vyžádat přepnutí na určitou sadu parametrů.

Poznámka

Během opracovávání závitu není přepnutí na jinou sadu parametrů možné

Při řezání závitů pomocí příkazu G33 a vrtání závitu pomocí příkazu G331 / G332 je sada parametrů vybírána řídícím systémem a není možní ji změnit.

Zablokované přepnutí sady parametrů

Přepnutí sady parametrů je možné si vyžádat také pomocí rozhraní NC/PLC. Aby se zabránilo konfliktům, které by při přepínání mohly vzniknout, je možné pomocí rozhraní NC/PLC přepínání sady parametrů NC systému (SCPARA) zablokovat.

DB31, ... DBX9.3 (zadaná sada parametrů prostřednictvím NC systému blokována)

Poznámka

Pokud je pomocí příkazu SCPARA vyžádáno přepnutí sady parametrů, zatímco je přepínání sady parametrů pomocí rozhraní NC/PLC zablokováno, přepnutí se neuskuteční, aniž by se vypisovalo chybové hlášení.

Syntaxe

```
SCPARA[<osa>]=<hodnota>
```

Význam

SCPARA:	Příkaz: Přepnutí sady parametrů	
<osa>:	Identifikátor osy (kanálová osa)	
	Typ:	AXIS
<hodnota>:	Číslo sady parametr: 1, 2, 3, ... max. číslo sady parametrů	

Příklad:

Programový kód	Komentář
...	
N110 SCPARA[X]= 3	; aktivování: Osa X, 3. sada parametrů
...	

Další informace

Odblokování přepnutí sady parametrů

Přepínání sady parametrů pro danou osu musí být explicitně odblokováno:

MD35590 $MA_PARAMSET_CHANGE_ENABLE[<osa>]

Načtení čísla sady parametrů

Číslo zvolené sady parametrů (požadovaná sada parametrů) je možné načíst prostřednictvím systémové proměnné $AA_SCPAR.

4.15.8 Aktivování/deaktivování přizpůsobování (CADAPTON, CADAPTOF)

Pomocí předem definovaných procedur CADAPTON() a CADAPTOF() můžete z výrobního programu aktivovat, aktualizovat a deaktivovat předem definovaná přizpůsobení dynamiky nebo regulačních parametrů. Tyto funkce jsou používány také v cyklu CYCLE782 (Strana 1105).

Syntaxe

```
...
CADAPTON(<Výsledek>,<osa>,<InVar>[,<InVal>])
...
CADAPTOF(<výsledek>,<osa>,<InVar>)
...
```

Význam

`CADAPTON()`:	Aktivování přizpůsobovacího vztahu		
`CADAPTOF()`:	Deaktivování přizpůsobovacího vztahu **Poznámka:** Když jsou trvale nastavené přizpůsobovací vztahy (MD16501 = 1) ignorovány `CADAPTOF()`, případně předtím naprogramovaná vstupní hodnota (`<InVal>`) zůstává v platnosti.		
`<výsledek>`:	Proměnná pro výsledek: Výsledná hodnota pro stav (parametr typu Call by reference)		
	Datový typ:	INT	
	Hodnota:	0	žádná chyba
		1	v parametrech není definována žádná platná tabulka přizpůsobení
		2	Parametr `<osa>` je neplatný
		3	Parametr `<InVar>` je neplatný
		4	rezervováno
		5	Parametr `<InVal>` je neplatný
`<osa>`:	Název strojní osy, která je vstupní osou pro přizpůsobovací vztah		
	Datový typ:	AXIS	
	Rozsah hodnot:	Identifikátory strojních os definovaných v kanálu	
	Poznámka: Pomocí tohoto parametru jsou vybírána přizpůsobení, která mají v parametru MD16504 $MN_CADAPT_INPUT_AX zadánu hodnotu, která odpovídá parametru `<osa>`. Další parametry `<InVar>` a `<InVal>` jsou přiřazeny této ose.		
`<InVar>`:	Vstupní veličina pro přizpůsobovací vztah		
	Datový typ:	INT	
	Hodnota:	1	Setrvačnost osy
		2	Poloha osy
		3	Rychlost osy

<InVal>:	Vstupní hodnota pro přizpůsobovací vztah Volitelný parametr pro inteligentní přizpůsobování zátěži (<InVar>=1).		
	Datový typ:	REAL	
	Hodnota:	> 0	Aktuální setrvačnost
		= 0	Zátěž/zatížení nejsou definovány, příp. nejsou známy. Přiřazená přizpůsobení v tomto případě poskytují jako výstupní veličinu náhradní hodnotu 1,0.
		< 0	Volání je ukončeno, přičemž je nastaveno <Status> = 5 (Parametr <InVal> je neplatný). Naposled aktivovaná vstupní hodnota zůstává i nadále aktivní.
	Poznámka: Pokud vstupní hodnota není naprogramována, zůstává v platnosti naposled naprogramovaná hodnota, příp. standardní hodnota po náběhu řídícího systému (= 0).		

Příklad

Programový kód	**Komentář**
DEF INT RESULT	; Definice proměnné pro výsledky
...	
CADAPTON(RESULT,MX1,2)	; Aktivování přizpůsobování
IF RESULT <> 0	; Vyhodnocování proměnné pro výsledky (je zapotřebí po každém příkazu CADAPTON/ CADAPTOF)
MSG("CADAPT-RESULT=" << RESULT)	
STOPRE	
SETAL(61000)	
ENDIF	
...	
CADAPTOF(RESULT,MX1,2)	; Deaktivování přizpůsobování
...	; Vyhodnocování proměnné pro výsledky
...	

Další informace

Spřažené osy

Pro spřažené osy (funkce NCU-Link) není možné tuto funkci používat. Proměnná pro výsledek by bývala v tomto případě poskytovala výslednou hodnotu 2 ("Parametr <osa> je neplatný").

Vyhledávání bloku

- **Vyhledávání bloku bez výpočtu**
  Příkazy CADAPTON/CADAPTOF, které jsou naprogramovány mezi začátkem programu s cílem vyhledávání, jsou ignorovány. Uživatel musí cíl vyhledávání zvolit tak, aby přizpůsobení, která jsou pro daný úsek opracování důležitá, byla aktivována.
- **Vyhledávání bloku s výpočtem na konturu, příp. na koncový bod bloku**
  Příkazy pro aktivování, příp. deaktivování, stejně jako případně naprogramované vstupní hodnoty jsou automaticky shromažďovány a aktuální stav na konci vyhledávání je poskytnut spolu s příslušnými akčními bloky.
- **Vyhledávání bloku s výpočtem v režimu testování programu (SERUPRO)**
  Spolu s deaktivováním příkazu SERUPRO se stav přizpůsobování dosažený prostřednictvím příkazů CADAPTON/CADAPTOF přenese do reálného provozu.

4.15.9 Přizpůsobení filtru ryvu FIR režimu dynamiky (CALCFIR)

Aby bylo po změně režimu dynamiky pro všechny osy v kanálu, pro které je aktivován filtr ryvu FIR typu dolní propust, dosaženo identického tlumicího efektu a přesnosti kontury, je možno pomocí příkazu CALCFIR jazyka NC systému vypočítat na dynamice závislá nastavení parametrů filtru FIR a současně je aktivovat.

Použití

Procedura CALCFIR se používá v cyklu výrobce CUST_832.

Standardní postup

Na začátku NC programu pro obrábění ploch libovolných tvarů je naprogramován cyklus CYCLE832. Při zpracovávání programu volá cyklus CYCLE832 cyklus výrobce CUST_832 a příkaz CALCFIR jazyka NC systému, který je výrobcem stroje do cyklu CUST_832 vložen, se uskuteční.

Platnost

Nastavení parametrů filtru přepisovaná procedurou CALCFIR zůstávají v platnosti tak dlouho, dokud nejsou přepsána znovu následujícím voláním procedury CALCFIR. To je zapotřebí vždy, když je prostřednictvím příkazu ze skupiny G-funkcí 59 (režim dynamiky pro dráhovou interpolaci) naprogramována změna aktuálního režimu dynamiky.

Chování při rozběhu řídícího systému / resetu kanálu / resetu na konci programu

Při rozběhu řídícího systému a při resetu kanálu/na konci programu se automaticky aktivují nastavení filtru FIR, která odpovídají deaktivování skupiny G-funkcí 59.

Předpoklady

- Je instalován volitelný doplněk „Top Speed".

  Poznámka

  Aby bylo pomocí funkce „Top Speed" dosahováno optimálních výsledků, doporučujeme vám současně používat funkci „Top Surface".

- Pro osy v interpolačním svazku musí být splněny následující podmínky:
  - Filtr ryvu FIR typu dolní propust je aktivní.
  - Parametry filtru FIR, které závisí na dynamice, jsou v konfiguraci definovány.
  - Přepisování nastavení parametrů filtru pomocí procedury CALCFIR je odblokováno.
- Omezení ryvu SOFT/SOFTA je aktivní.

Syntaxe

Příkaz CALCFIR je naprogramován v samostatném bloku. Volání cyklu výrobce CUST_832 se uskutečňuje bezprostředně po naprogramování režimu dynamiky:

```
...
DYN...
CALCFIR
...
```

Význam

DYN...:	Příkaz ze skupiny G-funkcí 59 pro volbu režimu dynamiky	
	DYNNORM:	Aktivování **normální dynamiky**
	DYNPOS:	Aktivování dynamiky pro **režim polohování, vrtání závitu**
	DYNROUGH:	Aktivování dynamiky pro **obrábění nahrubo**
	DYNSEMIFIN:	Aktivování dynamiky pro **předběžné obrábění načisto**
	DYNFINISH:	Aktivování dynamiky pro **obrábění načisto**
	DYNPREC:	Aktivování dynamiky pro **jemné obrábění načisto**
CALCFIR:	Předdefinovaná procedura pro přizpůsobení filtru ryvu FIR typu dolní propust režimu dynamiky	

Poznámka

Procedura CALCFIR provádí implicitní proceduru NEWCONFIG, aby byl aktivován výsledek výpočtu parametrů filtru FIR, stejně jako specifické osové strojní parametry pro nastavení filtru ryvu FIR typu dolní propust a pro příkaz CPRECON (Strana 854). Implicitní procedura NEWCONFIG má za následek, že je možno aktivovat ještě i další strojní parametry, které jsou v proceduře NEWCONFIG zahrnuty.

Okrajové podmínky

Použití v synchronních akcích

Příkaz CALCFIR **nemůže** být naprogramován v synchronních akcích.

4.15.10 Čtení/zápis do parametrů pohonu ve výrobním programu (DRVPRD, DRVPWR)

Příkazy jazyka DRVPRD a DRVPWR umožňují výrobci stroje provádět na úrovni výrobního programu čtení parametrů pohonu pro specifické osy a zápis do těchto parametrů.

Protože se v případě příkazů DRVPRD a DRVPWR jedná o předem definované procedury, které se musí nacházet v samostatném bloku NC programu, je možno v jednom okamžiku číst **nebo** zapisovat právě jeden parametr pohonu v jednom kanálu.

Příkazy DRVPRD a DRVPWR jsou definovány v cyklu vytvořeném výrobcem stroje.

Předpoklady

Platí následující předpoklady:

- Uvádění do provozu pohonu řízeného NC systémem je ukončeno.
- Cykly v adresáři CST (Siemens) nebo v adresáři CMA (výrobce) mají vždy všechna potřebná oprávnění ke čtení a k zápisu.

Syntaxe

```
DRVPRD(<Výsledek>,<Osa>,<DrvParNo>,<DrvParIdx>,<Hodnota>)
DRVPWR(<Výsledek>,<Osa>,<DrvParNo>,<DrvParIdx>,<Hodnota>)
```

Význam

`DRVPRD()`:	Předdefinovaná procedura pro **čtení** parametrů pohonu		
`DRVPWR()`:	Předdefinovaná procedura pro **zápis** do parametrů pohonu		
`<výsledek>`:	Proměnná pro výsledek: Výsledná hodnota pro stav		
	Datový typ:	INT	
	Hodnota:	= 0	Funkce byla zpracována bezchybně
		> 0	Funkce skončila s chybou Pokud budete potřebovat podrobnosti, viz „Další informace“ > „Diagnostika prostřednictvím výstupní hodnoty“.
`<osa>`:	Název osy stroje		
	Datový typ:	AXIS	
	Rozsah hodnot:	Identifikátory strojních os definovaných v kanálu	
`<DrvParNo>`:	Číslo parametru pohonu		
	Datový typ:	INT	
`<DrvParIdx>`:	Index parametru pohonu		
	Datový typ:	INT	

<hodnota>:	Hodnota parametru pohonu DRVPRD: čtená hodnota DRVPWR: hodnota, která se má zapsat	
	Datový typ:	REAL

Příklad

Načtení hodnoty parametru p1460[0] „Zesílení P regulátoru otáček, úprava hodnoty otáček" pro osu X:

Programový kód	**Komentář**
`N100 DEF INT _Status`	`; Definice proměnné pro výstupní hodnotu příkazu jazyka`
`N110 DEF REAL _Result`	`; Definice proměnné pro načítanou hodnotu parametru`
`N120 DRVPRD(_Status,AX1,1460,0,_Result)`	`; Čtení parametru p1460`

Další informace

Diagnostika pomocí výstupních hodnot

Přístup k parametrům pohonu je neplatný, jestliže se výstupní hodnota příkazu jazyka nerovná nule.

Pro tuto výstupní hodnotu jsou rezervovány dva intervaly čísel:

Výstupní hodnota při volání příkazu jazyka:

0 Žádná chyba při přístupu.
1 Hodnota není přípustná.
2 Osa není k dispozici.
3 Pohon není k dispozici.
4 Chybějící přístupová oprávnění.
5 Ve vyhledávání bloku/testování programu není možné.
6 Přístup k pohonu není možný.

Výstupní hodnoty při zápisu/čtení parametru pohonu:

1000 Nepřípustné číslo parametru.
1001 Hodnotu parametru nelze změnit.
1002 Překročení dolní nebo horní mezní hodnoty.
1003 Nesprávný dílčí index.
1004 Žádné pole, žádný dílčí index.
...
1107 Přístup pro zápis je u uvolněného regulátoru nepřípustný.
1110 Přístup pro zápis je povolen jen ve stavu, kdy probíhá uvádění do provozu: Motor (p0010 = 3).

...
1204 Přístup za účelem zápisu není dovolen.

Aby se výstupní hodnoty příkazů jazyka odlišily od výstupních hodnot pohonu, přičítá se hodnota 1000.

Další informace: Příručka pro diagnostiku systému SINUMERIK, alarm 201042 „Chyba parametru při načítání projektu (Download)"

Chování při simulaci programu

Parametry pohonu, které jsou k dispozici, závisí na formě a rozsahu použité simulace pohonu. V případě, že simulace pohonu skončí neúspěchem, poskytuje příkaz jazyka výstupní hodnotu 3.

Pomocí proměnné $P_SIM má výrobce stroje možnost stanovit pro simulaci programu určité zvláštní chování.

Vzájemné ovlivňování s jinými funkcemi

- Souběžně probíhající přístupy za účelem čtení/zápisu
  Souběžně probíhající přístupy za účelem čtení/zápisu z HMI nebo z jiných klientů mohou mít vliv na dobu, po kterou trvá zpracování příkazu jazyka.
- REPOS
  V případě přerušení zpracovávání příkazu jazyka událostí typu REORG-je příkaz opakován.
- Vyhledávání bloku / zkušební zpracování programu
  Při vyhledávání bloku a zkušebním zpracování programu (i s příkazem SERUPRO) nejsou příkazy jazyka pro zápis a čtení parametrů pohonu podporovány. Výsledná proměnná obsahuje chybu přístupu 5.
  Jestliže je ve výrobním programu pomocí systémových proměnných $P_SEARCH a $P_ISTEST zjišťováno, zda je aktivní vyhledávání bloku nebo zkušební zpracování programu, je například možné příkazy DRVPRD a DRVPWR přeskakovat a implementovat náhradní strategii.

Okrajové podmínky

Při programování je potřeba mít na paměti následující podmínky:

- Použití v synchronních akcích
  Příkaz jazyka nemůže být naprogramován v synchronních akcích, protože synchronní akce nečeká na zpracování necyklických přístupů.
  Když jsou tyto příkazy jazyka použity v synchronní akci, aktivuje se alarm 12571.
- Parametry pohonu, které lze číst a do nichž lze zapisovat
  Je možné zapisovat a číst pouze osové parametry pohonu SERVO-DO nebo HLA-DO, jakož i parametry normovaných pohonů PROFIDRIVE, které jsou přiřazeny osám NC systému.
  Není možné čtení/zápis do parametrů následujících DO: CU_I, CU_NX, CU3x, TM, HUB, INFEED.

4.16 Spojení os vazbou

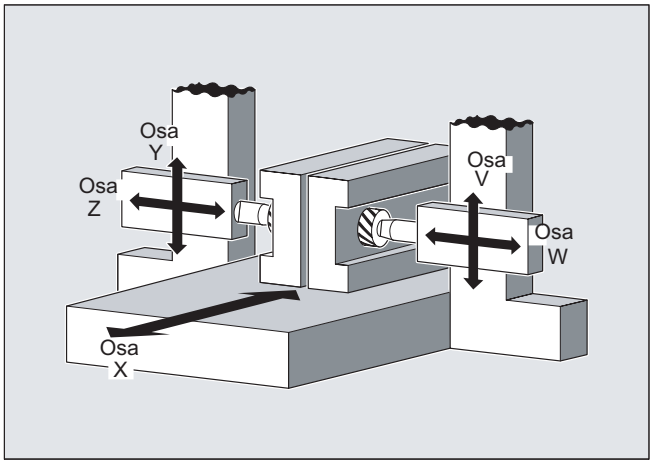
4.16.1 Vlečení (TRAILON, TRAILOF)

Při pohybech definované řídící osy se pohybují k ní přiřazené vlečené osy (= závislé osy) po drahách, které jsou odvozeny od pohybů řídící osy, přičemž je zohledňován ještě faktor vazby.

Řídící osa a vlečné osy spolu dohromady tvoří vlečné spojení os.

Oblasti použití

- Ovládání pohybu osy pomocí simulované osy. Řídící osa je simulovanou osou a vlečná osa je reálnou osou. Tímto způsobem je možné ovládat pohyb reálné osy, přičemž je zohledněn faktor vazby.
- Obrábění na dvou stranách se 2 vlečnými spojeními os:
  1. řídící osa Y, vlečná osa V
  2. řídící osa Z, vlečná osa W



Syntaxe

```
TRAILON(<vlečná osa>,<řídící osa>,<faktor vazby>)
TRAILOF(<vlečná osa>,<řídící osa>,<řídící osa 2>)
TRAILOF(<vlečná osa>)
```

Význam

TRAILON:	Příkaz pro aktivování a definici vlečného spojení os.	
	Platnost:	modální
<vlečná osa>:	Parametr 1: Identifikátor vlečné osy **Upozornění:** Vlečná osa může být také řídící osou pro další vlečené osy. Tímto způsobem je možné vytvářet různá další vlečná spojení os.	
<řídící osa>:	Parametr 2: Identifikátor řídící osy	

`<faktor vazby>`:	Parametr 3: Faktor vazby Faktor vazby udává požadovaný poměr mezi dráhou vlečné osy a dráhou řídící osy: <Faktor vazby> = Dráha vlečné osy / Dráha řídící osy	
	Typ:	REAL
	Předdefinované nastavení:	1
	Zadání záporné hodnoty má za následek, že se vlečná osa bude pohybovat v opačném směru než osa řídící. Jestliže faktor vazby není při programování uveden, automaticky potom platí faktor vazby 1.	
`TRAILOF`:	Příkaz pro deaktivování vlečného spojení os	
	Platnost:	modální

	Příkaz `TRAILOF` se 2 parametry vypíná pouze vazbu k uvedené řídící ose:
	`TRAILOF(<vlečná osa>,<řídící osa>)`
	Jestliže je vlečná osa řízena 2 řídícími osami, pro zrušení obou vazeb je možné vyvolat příkaz `TRAILOF` se 3 parametry:
	`TRAILOF(<vlečná osa>,<řídící osa>,<řídící osa 2>)`
	Stejného výsledku dosáhnete tím, že příkaz `TRAILOF` naprogramujete bez udání řídící osy:
	`TRAILOF(<vlečná osa>)`	

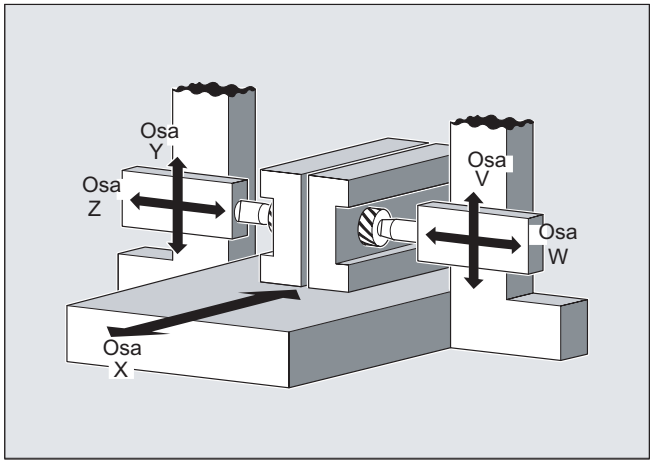
Poznámka

Vlečení se vždy uskutečňuje v základním souřadném systému (BCS).

Počet vlečných spojení, která mohou být aktivována současně, je omezen pouze kombinačními možnostmi os, které jsou na stroji k dispozici.

Příklad:

Obrobek má být opracován na obou stranách s konfigurací os uvedenou na obrázku. Za tím účelem jsou vytvořena 2 vlečná spojení os.



Programový kód	Komentář
…	
N100 TRAILON(V,Y)	; aktivování 1. vlečného spojení os
N110 TRAILON(W,Z,-1)	; aktivování 2. vlečného spojení os Faktor vazby je záporný: Vlečená osa se pohybuje vždy v opačném směru než osa řídící.
N120 G0 Z10	; přísuv osy Z a osy W v opačných směrech os
N130 G0 Y20	; přísuv osy Z a osy V ve stejném směru
…	
N200 G1 Y22 V25 F200	; superpozice závislého a nezávislého pohybu vlečné osy V
…	
TRAILOF(V,Y)	; deaktivování 1. vlečného spojení os
TRAILOF(W,Z)	; deaktivování 2. vlečného spojení os

Další informace

Typy os

Vlečné spojení os se může skládat z libovolné kombinace lineárních a kruhových os. Jako řídící osa přitom může být definována také simulovaná osa.

Vlečené osy

Vlečné ose je možné přiřadit maximálně 2 řídící osy současně. Přiřazení se provádí v různých vlečných spojeních.

Pro programování vlečné osy je možné využívat všech pohybových příkazů, které jsou k dispozici (G0, G1, G2, G3, ...). Pohyby vlečné osy se skládají z nezávisle definovaných drah, které se skládají s drahami odvozenými od řídících os, jež jsou upraveny pomocí faktoru vazby.

Omezení dynamiky

Omezení dynamiky závisí na druhu aktivování vlečné vazby:

- Aktivování ve výrobním programu
  Pokud je aktivování uskutečněno ve výrobním programu a pokud jsou všechny řídící osy programovými osami v aktivovaném kanálu, potom se při pohybech řídících os zohledňuje dynamika vlečných os tak, aby žádná z vlečných os nebyla přetížena.
  Jestliže je aktivování provedeno ve výrobním programu s řídícími osami, které nejsou aktivní jako programové osy v aktivovaném kanálu ($AA_TYP ≠ 1), potom se při pohybech řídících os na dynamiku vlečné osy nebere zřetel. V důsledku toho může u vlečných os s menší dynamikou, než jaká je pro vazbu potřebná, dojít k přetížení.
- Aktivování v synchronních akcích
  Pokud se aktivování uskuteční v synchronní akci, nebude se při pohybech řídících os dynamika vlečných os zohledňovat. V důsledku toho může u vlečných os s menší dynamikou, než jaká je pro vazbu potřebná, dojít k přetížení.

⚠ POZOR

Přetížení osy

Pokud je vlečné spojení os aktivováno

- v synchronních akcích
- ve výrobním programu s řídícími osami, které nejsou programovými osami v kanálu vlečné osy,

potom uživatel/výrobce stroje v plné míře odpovídají za to, že budou přijata vhodná opatření, aby v důsledku posuvových pohybů řídící osy nemohlo dojít k přetížení vlečných os.

Stav vazby

Stav vazby dané osy může být ve výrobním programu zjištěn pomocí následujících systémových proměnných:

$AA_COUP_ACT[<osa>]

Hodnota	Význam
0	Žádná vazba není aktivní
8	Je aktivní vlečení

Výpis zbytkové dráhy vlečené osy u kruhových os typu modulo

Pokud jsou řídící a vlečná osa kruhovými osami typu modulo, sečítají se posuvy při pohybech řídící osy n * 360°, kde n = 1, 2, 3... , ve vypisovaném údaji zbytkové dráhy vlečené osy až do deaktivování vazby.

Příklad: Úsek programu s příkazem TRAILON, řídící osou B a vlečnou osou C

```
Programový kód                  Komentář
TRAILON(C,B,1)                  ; Aktivování vazby
G0 B0                           ; Výchozí pozice
                                ; Údaj zbytkové dráhy na začátku bloku:
G91 B360                        ; B=360, C=360
```

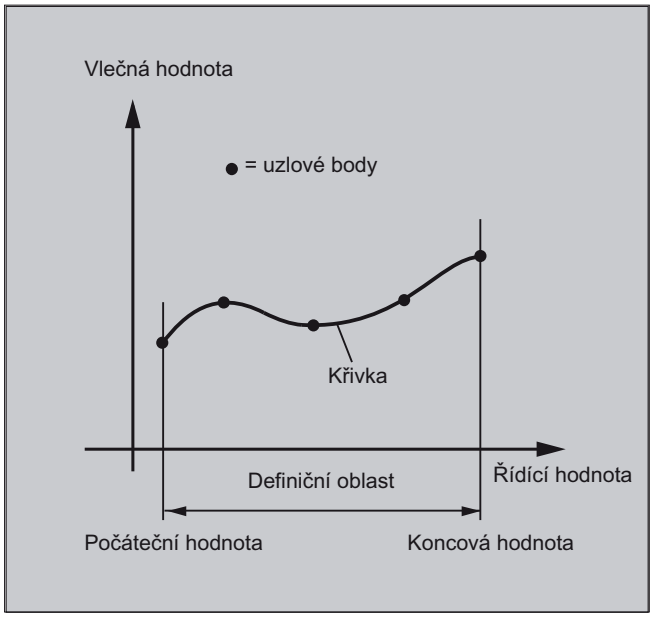
Programový kód	Komentář
G91 B720	; B=720, C=1080
G91 B360	; B=360, C=1440

4.16.2 Tabulky křivek (CTAB)

4.16.2.1 Definice tabulek křivek (CTABDEF, CATBEND)

Tabulka křivky představuje výrobní program nebo úsek výrobního programu, který je na začátku uveden příkazem CTABDEF a ukončen závěrečným příkazem CTABEND.

V rámci tohoto úseku výrobního programu jsou prostřednictvím pohybových příkazů přiřazeny jednotlivým pozicím řídící osy polohy závislé osy, které se pak používají jako uzlové body pro výpočet aproximační křivky ve formě polynomu maximálně 5. stupně.



Předpoklady

Pro definici tabulky křivky musí být prostřednictvím nastavení odpovídajících konfiguračních strojních parametrů vyhrazen patřičný paměťový prostor (--> výrobce stroje!).

Syntaxe

```
CTABDEF(<vlečná osa>,<řídící osa>,<n>,<periodicita>[,<paměťové
místo>])
...
CTABEND
```

Význam

CTABDEF( ):	Začátek definice tabulky křivky	
CTABEND:	Konec definice tabulky křivky	
<vlečná osa>:	Osa, jejíž pohyb se má vypočítávat prostřednictvím tabulky křivky.	
<řídící osa>:	Osa, která poskytuje řídící hodnotu pro výpočet pohybu vlečné osy.	
<n>:	Číslo (**ID**) tabulky křivky Číslo tabulky křivky je jednoznačné a nezávisí na umístění v paměti. Ani ve statické, ani v dynamické paměti NC systému se nemohou nacházet tabulky se stejnými čísly.	
<periodicita>:	Periodicita tabulky	
	0	Tabulka není periodická (bude zpracována jen jednou, a to i v případě kruhových os)
	1	Tabulka je vzhledem k řídící ose periodická.
	2	Tabulka je periodická vzhledem k řídící ose i vlečné ose.
<paměťové místo>:	Údaj místa v paměti (nepovinné)	
	"SRAM"	Tabulka křivky se ukládá ve **statické** paměti NC systému.
	"DRAM"	Tabulka křivky se ukládá v **dynamické** paměti NC systému.
	Upozornění: Pokud pro tento parametr není naprogramována žádná hodnota, potom se použije standardní místo v paměti nastavené parametrem MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE.	

Poznámka

Přepisování

Tabulka křivky se přepíše, jakmile je při nové definici tabulky použito její číslo (<n>) (výjimka: Tabulka křivky je aktivní v rámci vazby mezi osami nebo je blokována příkazem CTABLOCK). **Při přepisování tabulky se nevypisuje žádná odpovídající výstraha!**

Příklady

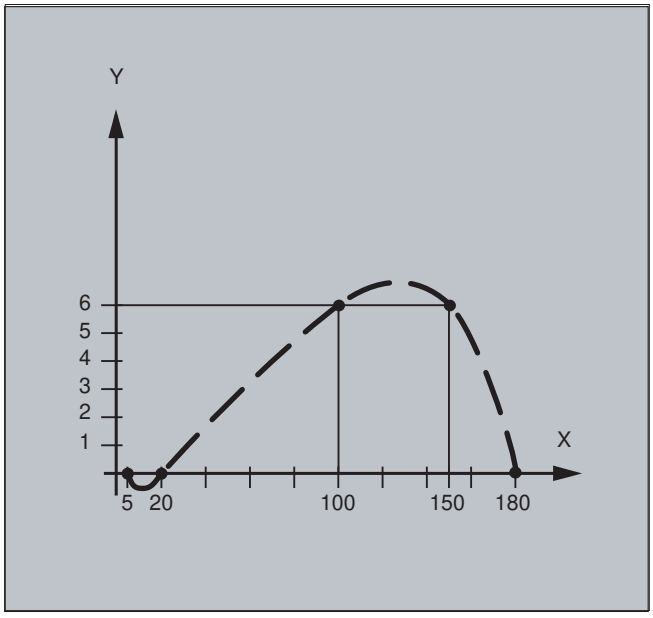
Příklad 1: Úsek programu jako definice tabulky křivky

Úsek programu má být použit nezměněn pro definici tabulky křivky. Příkaz pro zastavení předběžného zpracování STOPRE, který se v něm vyskytuje, může zůstat na svém místě a okamžitě se znovu aktivuje, jakmile se úsek programu přestane používat pro definici tabulky a jakmile je pomocí příkazů CTABDEF a CTABEND odstraněn.

```
Programový kód                  Komentář
…
CTABDEF(Y,X,1,1)                ; definice tabulky křivky
…
IF NOT ($P_CTABDEF)
STOPRE
ENDIF
…
```

Programový kód	Komentář
CTABEND	

Příklad 2: Definice neperiodické tabulky křivky



Programový kód	Komentář
N100 CTABDEF(Y,X,3,0)	; začátek definice neperiodické tabulky křivky s číslem 3
N110 X0 Y0	; 1. pohybový příkaz, definuje počáteční hodnotu a 1. uzlové místo:
	Řídící hodnota: 0, vlečná hodnota: 0
N120 X20 Y0	; 2. uzlové místo:
	Řídící hodnota: 0…20, vlečná hodnota: Počáteční hodnota… 0
N130 X100 Y6	; 3. uzlové místo:
	Řídící hodnota: 20…100, vlečná hodnota: 0…6
N140 X150 Y6	; 4. uzlové místo:
	Řídící hodnota: 100…150, vlečná hodnota: 6…6
N150 X180 Y0	; 5. uzlové místo:
	Řídící hodnota: 150…180, vlečná hodnota: 6…0
N200 CTABEND	; konec definice Tabulka křivky se ve své interní reprezentaci převede na polynom maximálně 5. stupně. Výpočet křivkového průběhu se zadanými uzlovými body závisí na zvoleném modálním druhu interpolace (kruhová, lineární, splinová interpolace). Opět se obnoví stav výrobního programu před zahájením definice.

Příklad 3: Definice periodické tabulky křivky

Definice periodické tabulky křivky s číslem 2, rozmezí řídící hodnoty od 0 do 360, pohyb vlečné osy od 0 do 45 a zpátky do 0.

```
Programový kód                                          Komentář
N10 DEF REAL DEPPOS
N20 DEF REAL GRADIENT
N30 CTABDEF(Y,X,2,1)                                    ; začátek definice
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                                            ; konec definice
;Text křivky pomocí vazby mezi Y a 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)                       ; načtení tabulkové funkce s řídí-
                                                          cí hodnotou 75,0
N180 G0 X75 Y=DEPPOS                                    ; polohování řídící a vlečné osy
;Po aktivování vazby už není žádná synchronizace vlečné osy zapotřebí.
N190 LEADON(Y,X,2)
N200 G1 X110 F1000
N210 LEADOF(Y,X)
N220 M30
```

Další informace

Počáteční a koncová hodnota tabulky křivky

Jako počáteční hodnota pro začátek definiční oblasti tabulky křivky platí první údaj k sobě patřících poloh os (první pohybový příkaz) uvnitř definice tabulky křivky. Koncová hodnota definiční oblasti tabulky křivky je analogicky určena prostřednictvím posledního pohybového příkazu.

Rozsah jazykových prostředků, které jsou k dispozici

V rámci definice tabulky křivky je k dispozici celý rozsah NC jazyka.

Poznámka

Následující příkazy jsou v definicích tabulek křivek nepřípustné:

- Zastavení předběžného zpracování
- Skokové pohyby řídící osy (např. při změně transformací)
- Příkaz pohybu pro samotnou závislou osu
- Změna směru pohybu řídící osy na opačný, tzn. poloha řídící osy musí být vždy jednoznačná.
- Příkazy CTABDEF a CTABEND na různých programových úrovních.

Platnost modálních příkazů

Veškeré příkazy s modální platností, které se vyskytují v rámci definice tabulky křivky, se stávají po ukončení definice tabulky neplatnými. Výrobní program, ve kterém se definice tabulky nachází, se takto nachází před a po definici tabulky ve stejném stavu.

Přiřazování R-Parametrů

Přiřazení R-parametrů v rámci definici tabulky se po příkazu CTABEND resetuje.

Příklad:

Programový kód	**Komentář**
...	
R10=5 R11=20	; R10=5
...	
CTABDEF	
G1 X=10 Y=20 F1000	
R10=R11+5	; R10=25
X=R10	
CTABEND	
...	; R10=5

Aktivování funkcí ASPLINE, BSPLINE, CSPLINE

Jestliže je v rámci definice tabulky křivky pomocí příkazů CTABDEF ... CTABEND aktivován některý z příkazů ASPLINE, BSPLINE nebo CSPLINE, potom by měl být před tímto příkazem splinu naprogramován minimálně jeden počáteční bod. Okamžitému aktivování po příkazu CTABDEF byste se měli vyhnout, protože jinak by spline závisel na momentální poloze osy před definicí tabulky křivky.

Příklad:

Programový kód
...
CTABDEF(Y,X,1,0)
X0 Y0
ASPLINE

Programový kód
X=5 Y=10
X10 Y40
...
CTABEND

Opakované použití tabulky křivky

Funkční závislost mezi řídící a vlečnou osou, která byla vypočítána pomocí tabulky křivky, zůstává pod vybraným číslem tabulky i po skončení výrobního programu a po vypnutí a zapnutí systému (Power off), jestliže je ale tabulka uložena ve statické paměti NC systému.

Tabulka, která byla uložena v dynamické paměti (DRAM), se při vypnutí a zapnutí systému (power-on) vymaže a v případě potřeby musí být sestavena ještě jednou.

Jedenkrát sestavená tabulka křivky může být využita libovolnou kombinací řídící a vlečné osy a je nezávislá na tom, které osy byly pro její sestavení použity.

Přepisování tabulek křivek

Tabulka křivky se přepíše, jakmile je při definici tabulky znovu použito její číslo.

Výjimka: Tabulka křivky je aktivní v rámci vazby mezi osami nebo je blokována příkazem CTABLOCK.

Poznámka

Při přepisování tabulky křivky se nevypisuje žádná odpovídající výstraha!

Je definice tabulky křivky aktivní?

Prostřednictvím systémové proměnné `$P_CTABDEF` je možné z výrobního programu kdykoli zjistit, zda je definice tabulky křivky aktivní.

Ukončení definice tabulky křivky

Úsek výrobního programu je po vymezení definice tabulky křivky, které označují její začátek a konec, opět použitelný jako reálný výrobní program.

Načítání tabulek křivek pomocí funkce "Zpracování z externího zdroje"

Při zpracovávání tabulek křivek z externího zdroje musí být velikost používané vyrovnávací paměti (DRAM) nastavena pomocí parametru MD18360 $MN_MM_EXT_PROG_BUFFER_SIZE tak, aby bylo možné do této vyrovnávací paměti uložit celou definici tabulky naráz. Zpracovávání výrobního programu se jinak přeruší a aktivuje se alarm.

Skokové změny vlečné osy

V závislosti na nastavení strojního parametru
MD20900 $MC_CTAB_ENABLE_NO_LEADMOTION
mohou být skokové změny vlečné osy v případě chybějícího pohybu řídící osy tolerovány.

4.16.2.2 Kontrola existence tabulky křivky (CTABEXISTS)

Pomocí příkazu CTABEXISTS může být zkontrolováno, zda se tabulka s určitým číslem křivky vyskytuje v paměti NC systému.

Syntaxe

```
CTABEXISTS(<n>)
```

Význam

CTABEXISTS:	Zkontroluje se, zda je tabulka křivky s číslem <n> k dispozici ve statické nebo dynamické paměti NC systému.	
	0	Tabulka neexistuje
	1	Tabulka existuje
<n>:	Číslo (**ID**) tabulky křivky	

4.16.2.3 Mazání tabulek křivek (CTABDEL)

Pomocí příkazu CTABDEL je možné mazat tabulky křivek.

Poznámka

Tabulky křivky, které jsou aktivní ve vazbě mezi osami, nemohou být vymazány.

Syntaxe

```
CTABDEL(<n>)
CTABDEL(<n>,<m>)
CTABDEL(<n>,<m>,<paměťové místo>)
CTABDEL()
CTABDEL(,,<paměťové místo>)
```

Význam

CTABDEL:	Příkaz pro vymazání tabulky křivky
<n>:	Číslo (**ID**) tabulky křivky, která má být vymazána Při mazání intervalu tabulek křivek pomocí příkazu CTABDEL(<n>,<m>) se pomocí parametru <n> zadává číslo první tabulky křivky tohoto intervalu.
<m>:	Při mazání intervalu tabulek křivek pomocí příkazu CTABDEL(<n>,<m>) se pomocí parametru <m> zadává číslo poslední tabulky křivky tohoto intervalu. Parametr <m> musí být větší než <n>!

<paměťové místo>:	Údaj místa v paměti (nepovinné) Při mazání **bez** zadání paměťového místa jsou uvedené tabulky křivek vymazány ze statické a dynamické paměti NC systému. Při mazání **se** zadáním paměťového místa jsou vymazány jen ty uvedené tabulky křivek, které se nacházejí v zadané paměti. Zbývající tabulky zůstanou nedotčeny.	
	"SRAM"	Mazání ve **statické** paměti NC systému
	"DRAM"	Mazání v **dynamické** paměti NC systému

Pokud je naprogramován příkaz CTABDEL bez udání tabulky křivky, která má být vymazána, potom se vymažou **všechny** tabulky křivek, příp. všechny tabulky křivek z uvedené paměti.

CTABDEL():	Vymažou se všechny tabulky křivek ve statické a dynamické paměti NC systému.
CTABDEL(,,"SRAM"):	Vymažou se všechny tabulky křivek ve statické paměti NC systému.
CTABDEL(,,"DRAM"):	Vymažou se všechny tabulky křivek v dynamické paměti NC systému.

Poznámka

Pokud je v případě mazání většího počtu křivek pomocí příkazů CTABDEL(<n>,<m>) nebo CTABDEL() alespoň jedna z mazaných tabulek křivek součástí aktivní vazby, potom se příkaz mazání neuskuteční, tzn. nevymaže se **žádná** ze zadaných tabulek křivek.

4.16.2.4 Blokování tabulek křivek proti mazání a přepisování (CTABLOCK, CTABUNLOCK)

Tabulky křivek mohou být aktivováním blokování chráněny proti nechtěnému vymazání a přepsání. Aktivované blokování může být kdykoli opět zrušeno.

Syntaxe

Aktivování blokování:

```
CTABLOCK(<n>)
CTABLOCK(<n>,<m>)
CTABLOCK(<n>,<m>,<paměťové místo>)
CTABLOCK()
CTABLOCK(,,<paměťové místo>)
```

Deaktivování blokování:

```
CTABUNLOCK(<n>)
CTABUNLOCK(<n>,<m>)
CTABUNLOCK(<n>,<m>,<paměťové místo>)
CTABUNLOCK()
CTABUNLOCK(,,<paměťové místo>)
```

Význam

CTABLOCK:	Příkaz pro **aktivování** blokování proti vymazání/přepsání
CTABUNLOCK:	Příkaz pro **deaktivování** blokování proti vymazání/přepsání Příkazem CTABUNLOCK se tabulky křivek zablokované příkazem CTABLOCK opět odblokují. Tabulky křivky, které se uplatňují v aktivní vazbě, zůstanou i nadále zablokovány a nemohou být vymazány. Blokování příkazem CTABLOCK je odstraněno, jakmile pomine ochrana vyvolávaná aktivní vazbou v důsledku toho, že je vazba deaktivována. Potom může být tato tabulka vymazána. Opětovné vyvolání příkazu CTABUNLOCK není zapotřebí.
<n>:	Číslo (**ID**) tabulky křivky, která má být blokována/odblokována Při aktivování/deaktivování blokování intervalu tabulek křivek pomocí příkazů CTABLOCK(<n>,<m>)/CTABUNLOCK(<n>,<m>) se pomocí parametru <n> zadává číslo první tabulky křivky tohoto intervalu.
<m>:	Při aktivování/deaktivování blokování intervalu tabulek křivek pomocí příkazů CTABLOCK(<n>,<m>)/CTABUNLOCK(<n>,<m>) se pomocí parametru <m> zadává číslo poslední tabulky křivky tohoto intervalu. Parametr <m> musí být větší než <n>!
<paměťové místo>:	Údaj místa v paměti (nepovinné) Při aktivování/deaktivování blokování **bez** zadání paměťového místa jsou uvedené tabulky křivek blokovány/odblokovány ve statické a dynamické paměti NC systému. Při aktivování/deaktivování blokování **se** zadáním paměťového místa jsou blokovány/odblokovány jen ty uvedené tabulky křivek, které se nacházejí v zadané paměti. Zbývajících se blokování/odblokování netýká.
"SRAM"	Aktivování/deaktivování blokování ve **statické** paměti NC systému
"DRAM"	Aktivování/deaktivování blokování v **dynamické** paměti NC systému

Pokud je naprogramován příkaz CTABLOCK/CTABUNLOCK bez udání tabulky křivky, jejíž blokování má být aktivováno/deaktivováno, potom se aktivování/deaktivování blokování týká **všech** tabulek křivek, příp. všech tabulek křivek z uvedené paměti.

CTABLOCK():	Zablokují se všechny tabulky křivek ve statické a dynamické paměti NC systému.
CTABLOCK(,,"SRAM"):	Zablokují se všechny tabulky křivek ve statické paměti NC systému.
CTABLOCK(,,"DRAM"):	Zablokují se všechny tabulky křivek v dynamické paměti NC systému.
CTABUNLOCK():	Odblokují se všechny tabulky křivek ve statické a dynamické paměti NC systému.
CTABUNLOCK(,,"SRAM"):	Odblokují se všechny tabulky křivek ve statické paměti NC systému.
CTABUNLOCK(,,"DRAM"):	Odblokují se všechny tabulky křivek v dynamické paměti NC systému.

4.16.2.5 Tabulky křivek: Zjišťování vlastností tabulek (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD)

Pomocí těchto příkazů mohou být zjišťovány důležité vlastnosti tabulek křivek (číslo tabulka, stav blokování, místo uložení, periodicita).

Syntaxe

```
CTABID(<p>)
CTABID(<p>,<paměťové místo>)
CTABISLOCK(<n>)
CTABMEMTYP(<n>)
TABPERIOD(<n>)
```

Význam

CTABID:	Výsledkem je **číslo tabulky**, která je v zadané paměti uložena jako <p>-tá tabulka křivky. Příklad: Výsledkem příkazu `CTABID(1,"SRAM")` bude číslo první tabulky křivky ve statické paměti NC systému. První tabulka křivky přitom odpovídá tabulce křivky s nejvyšším číslem tabulky. **Upozornění:** Jestliže se mezi dvěma po sobě následujícími voláními funkce `CTABID` posloupnost tabulek křivek v paměti změní, např. v důsledku vymazání tabulek křivek pomocí příkazu `CTABDEL`, může se stát, že výsledkem příkazu `CTABID(<p>,...)` se stejným číslem <p> bude jiná tabulka křivky než předtím.	
CTABISLOCK:	Výsledkem je **stav blokování** tabulky křivky s číslem <n>.	
	0	Tabulka není blokována
	1	Tabulka je blokována příkazem `CTABLOCK`
	2	Tabulka je blokována díky aktivní vazbě
	3	Tabulka je blokována příkazem `CTABLOCK` a díky aktivní vazbě
	-1	Tabulka neexistuje
CTABMEMTYP:	Výsledkem je **místo uložení** tabulky křivky s číslem <n>.	
	0	Tabulka ve statické paměti NC systému
	1	Tabulka v dynamické paměti NC systému
	-1	Tabulka neexistuje
CTABPERIOD:	Výsledkem je **periodicita** tabulky křivky s číslem <n>.	
	0	Tabulka není periodická
	1	Tabulka je vzhledem k řídící ose periodická.
	2	Tabulka je periodická vzhledem k řídící ose i vlečné ose.
	-1	Tabulka neexistuje
<p>:	Číslo položky v paměti	
<n>:	Číslo (**ID**) tabulky křivky	

<paměťové místo>:	Údaj místa v paměti (nepovinné)	
	"SRAM"	**statická** paměť NC systému
	"DRAM"	**dynamická** paměť NC systému
	Upozornění: Pokud pro tento parametr není naprogramována žádná hodnota, potom se použije standardní místo v paměti nastavené parametrem MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE.	

4.16.2.6 Čtení hodnot z tabulky křivky (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX)

Ve výrobním programu mohou být čteny následující hodnoty z tabulky křivky:

- Hodnoty vlečné a řídící osy na začátku a na konci tabulky křivky
- Hodnoty vlečné osy na začátku a na konci úseku křivky
- Hodnota vlečné osy pro určitou hodnotu řídící osy
- Hodnota řídící osy pro určitou hodnotu vlečné osy
- Minimální a maximální hodnota vlečné osy
  - v celkovém definičním rozsahu tabulky křivky nebo
  - v definovaném intervalu tabulky křivky

Syntaxe

```
CTABTSV(<n>,<gradient>[,<vlečná osa>])
CTABTEV(<n>,<gradient>[,<vlečná osa>])
CTABTSP(<n>,<gradient>[,<řídící osa>])
CTABTEP(<n>,<gradient>[,<řídící osa>])
CTABSSV(<řídící hodnota>,<n>,<gradient>[,<vlečná osa>])
CTABSEV(<řídící hodnota>,<n>,<gradient>[,<vlečná osa>])
CTAB(<řídící hodnota>,<n>,<gradient>[,<vlečná osa>,<řídící osa>]
CTABINV(<vlečná hodnota>,<hodnota
přiblížení>,<n>,<gradient>[,<vlečná osa>,<řídící osa>]
CTABTMIN(<n>[,<vlečná osa>])
CTABTMAX(<n>[,<vlečná osa>])
CTABTMIN(<n>,<a>,<b>[,<vlečná osa>,<řídící osa>])
CTABTMAX(<n>,<a>,<b>[,<vlečná osa>,<řídící osa>])
```

Význam

CTABTSV:	Načtení hodnoty vlečné osy na **začátku** tabulky křivky s číslem <n>.
CTABTEV:	Načtení hodnoty vlečné osy na **konci** tabulky křivky s číslem <n>.
CTABTSP:	Načtení hodnoty řídící osy na **začátku** tabulky křivky s číslem <n>.
CTABTEP:	Načtení hodnoty řídící osy na **konci** tabulky křivky s číslem <n>.
CTABSSV:	Načtení hodnoty vlečné osy na **začátku** úseku křivky, který patří k zadané hodnotě řídící osy (<řídící hodnota>)

CTABSEV:	Načtení hodnoty vlečné osy na **konci** úseku křivky, který patří k zadané hodnotě řídící osy (<řídící hodnota>)
CTAB:	Načtení hodnoty vlečné osy k zadané hodnotě řídící osy (<řídící hodnota>)
CTABINV:	Načtení hodnoty řídící osy k zadané hodnotě vlečné osy (<vlečná hodnota>)
CTABTMIN:	Stanovení **minimální hodnoty** vlečné osy: • v celkovém definičním rozsahu tabulky křivky nebo • v definovaném intervalu <a> ... <b>
CTABTMAX:	Stanovení **maximální hodnoty** vlečné osy: • v celkovém definičním rozsahu tabulky křivky nebo • v definovaném intervalu <a> ... <b>
<n>:	Číslo (**ID**) tabulky křivky
<gradient>:	V parametru <gradient> systém poskytuje údaj o **stoupání** funkce popisované tabulkou křivky v bodě, který je předmětem zájmu
<vlečná osa>:	Osa, jejíž pohyb se má vypočítávat prostřednictvím tabulky křivky (nepovinné)
<řídící osa>:	Osa, která poskytuje řídící hodnotu pro výpočet pohybu vlečné osy (nepovinné)
<vlečná hodnota>:	Hodnota vlečné osy pro načtení k ní náležící hodnoty řídící osy pomocí příkazu CTABINV
<řídící hodnota>:	Hodnota řídící osy • pro načtení k ní náležící hodnoty vlečné osy pomocí příkazu CTAB nebo • pro volbu úseku křivky pomocí příkazu CTABSSV/CTABSEV
<hodnota přiblížení>:	Přiřazení hodnoty řídící osy hodnotě vlečné osy u příkazu CTABINV nemusí být vždy jednoznačné. Příkaz CTABINV potřebuje proto jako parametr hodnotu, která se blíží očekávané hodnotě řídící osy.
<a>:	**Spodní** mezní hodnota intervalu řídících hodnot u příkazů CTABTMIN/CTABTMAX
<b>:	**Horní** mezní hodnota intervalu řídících hodnot u příkazů CTABTMIN/CTABTMAX
	Upozornění: Interval řídících hodnot <a> ... <b> musí ležet v rámci definičního rozsahu tabulky křivky.

Příklady

Příklad 1:

Stanovení hodnot vlečné osy a řídící osy na začátku a na konci tabulky křivky, jakož i maximální a minimální hodnoty vlečné osy v celém definičním rozsahu tabulky křivky.

Programový kód	**Komentář**
N10 DEF REAL STARTPOS	
N20 DEF REAL ENDPOS	

Programový kód	Komentář
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)	; Začátek definice tabulky
N110 X0 Y10	; Počáteční pozice 1. úseku tabulky
N120 X30 Y40	; koncová pozice 1. úseku tabulky = počáteční pozice 2. úseku tabulky
N130 X60 Y5	; Koncová pozice 2. úseku tabulky = ...
N140 X70 Y30	
N150 X80 Y20	
N160 CTABEND	; Konec definice tabulky.
...	
N200 STARTPOS=CTABTSV(1,GRADIENT)	; Hodnota vlečné osy na začátku tabulky křivky = 10
N210 ENDPOS=CTABTEV(1,GRADIENT)	; Hodnota vlečné osy na konci tabulky křivky = 20
N220 STARTPARA=CTABTSP(1,GRADIENT)	; Hodnota řídící osy na začátku tabulky křivky = 0
N230 ENDPARA=CTABTEP(1,GRADIENT)	; Hodnota řídící osy na konci tabulky křivky = 80
N240 MINVAL=CTABTMIN(1)	; Minimální hodnota vlečné osy v bodě Y=5
N250 MAXVAL=CTABTMAX(1)	; Maximální hodnota vlečné osy v bodě Y=40

Příklad 2:

Stanovení hodnoty vlečné osy na začátku a na konci úseku křivky, který patří k hodnotě řídící osy X=30.

Programový kód	Komentář
N10 DEF REAL STARTPOS	
N20 DEF REAL ENDPOS	
N30 DEF REAL GRADIENT	
...	
N100 CTABDEF(Y,X,1,0)	; Začátek definice tabulky.
N110 X0 Y0	; Počáteční pozice 1. úseku tabulky
N120 X20 Y10	; koncová pozice 1. úseku tabulky = počáteční pozice 2. úseku tabulky
N130 X40 Y40	; Koncová pozice 2. úseku tabulky = ...
N140 X60 Y10	
N150 X80 Y0	
N160 CTABEND	; Konec definice tabulky.
...	
N200 STARTPOS=CTABSSV(30.0,1,GRADIENT)	; Počáteční pozice Y ve 2. úseku = 10
N210 ENDPOS=CTABSEV(30.0,1,GRADIENT)	; Koncová pozice Y ve 2. úseku = 40

Další informace

Použití v synchronních akcích

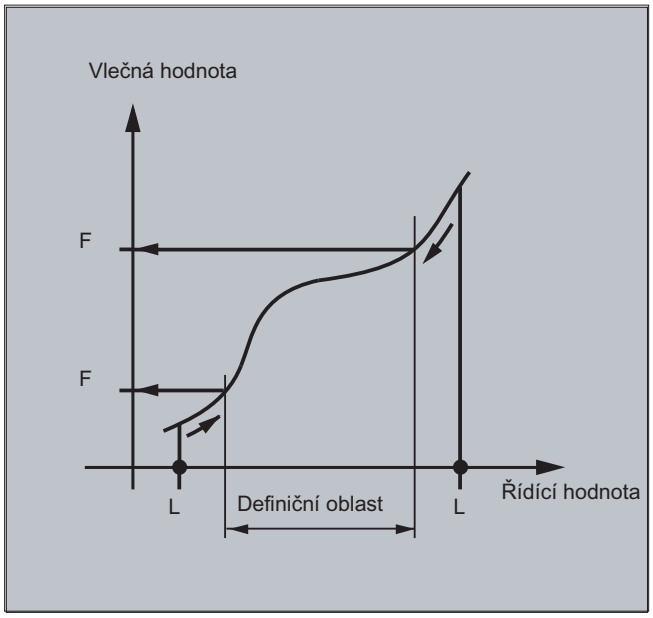
Všechny příkazy pro čtení hodnot z tabulky křivky se mohou používat také v synchronních akcích (viz také kapitola "Pohybové synchronní akce").

Při použití příkazů CTABINV, CTABTMIN a CTABTMAX je zapotřebí mít na paměti, aby:

- v okamžiku zpracování těchto příkazů byl k dispozici dostatečný výkon NC systému nebo
- aby před vyvoláním příkazu byl zjištěn počet úseků v tabulce křivky, aby bylo možné v případě potřeby příslušnou tabulku rozdělit

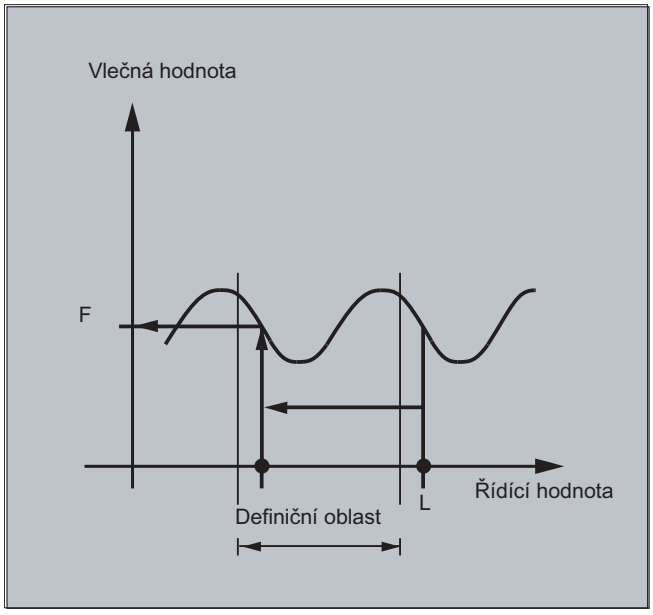
Příkaz CTAB v neperiodických tabulkách křivky

Jestliže hodnota zadaná v parametru <řídící hodnota> leží mimo definiční rozsah, bude jako vlečná hodnota uvedena horní, příp. dolní mezní hodnota:



Příkaz CTAB v periodických tabulkách křivky

Jestliže hodnota zadaná v parametru <řídící hodnota> leží mimo definiční rozsah, bude řídící hodnota upravena funkcí Modulo pro definiční rozsah a potom se zjistí odpovídající vlečná hodnota:



Hodnota přiblížení pro příkaz CTABINV

Příkaz CTABINV potřebuje hodnotu, která se blíží očekávané řídící hodnotě. Výsledkem funkce CTABINV je řídící hodnota, která leží nejblíže hodnotě přiblížení. Hodnotou přiblížení může být např. řídící hodnota z předcházejícího taktu interpolátoru.

Stoupání funkce dané tabulkou křivky

Zadání stoupání pomocí parametru (<gradient>) umožňuje započítat rychlost řídící nebo vlečné osy na odpovídajícím místě charakteristiky.

Zadání řídící nebo vlečné osy

Nepovinné zadání řídící a/nebo vlečné osy je důležité, jestliže jsou pro řídící a vlečnou osu v konfiguraci nastaveny různé jednotky délky.

CTABSSV, CTABSEV

Příkazy CTABSSV a CTABSEV **nejsou** v následujících případech vhodné k tomu**,** aby se zjišťovaly informace o naprogramovaných segmentech:

- Jsou naprogramovány kruhy nebo evolventy.
- Jsou aktivní fasety, příp. zaoblení pomocí příkazů CHF/RND.
- Jsou aktivní přechodová zaoblení aktivovaná příkazem G643.
- Je aktivní komprese NC-bloků (COMP...).

4.16.2.7 Tabulky křivek: Kontrola využití systémových zdrojů (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL)

Pomocí těchto příkazů má programátor možnost získat aktuální informace o využití systémových zdrojů pro tabulky křivek, úseky tabulek a polynomy.

Syntaxe

```
CTABNO
CTABNOMEM(<paměťové místo>)
CTABFNO(<paměťové místo>)
CTABSEGID(<n>,<paměťové místo>)
CTABSEG(<paměťové místo>,<druh úseku>)
CTABFSEG(<paměťové místo>,<druh úseku>)
CTABMSEG(<paměťové místo>,<druh úseku>)
CTABPOLID(<n>)
CTABPOL(<paměťové místo>)
CTABFPOL(<paměťové místo>)
CTABMPOL(<paměťové místo>)
```

Význam

CTABNO:	Stanovení celkového počtu **definovaných** tabulek křivek (ve statické a dynamické paměti NC systému)	
CTABNOMEM:	Stanovení počtu **definovaných** tabulek křivek v zadané paměti <paměťové místo>	
CTABFNO:	Stanovení počtu **ještě možných** tabulek křivek v zadané paměti <paměťové místo>	
CTABSEGID:	Stanovení počtu křivkových úseků odpovídajících zadanému <druhu úseku>, které jsou využívány tabulkou křivky s číslem <n>	
CTABSEG:	Stanovení počtu **používaných** úseků křivky odpovídajících zadanému <druhu úseku>, které jsou uloženy v zadané paměti <paměťové místo>	
CTABFSEG:	Stanovení počtu **ještě možných** úseků křivky odpovídajících zadanému <druhu úseku>, které jsou uloženy v zadané paměti <paměťové místo>	
CTABMSEG:	Stanovení počtu **maximálně možných** úseků křivky odpovídajících zadanému <druhu úseku>, které jsou uloženy v zadané paměti <paměťové místo>	
CTABPOLID:	Stanovení počtu křivkových polynomů, které jsou využívány tabulkou křivky s číslem <n>	
CTABPOL:	Stanovení počtu **používaných** křivkových polynomů v zadané paměti <paměťové místo>	
CTABFPOL:	Stanovení počtu **ještě možných** křivkových polynomů v zadané paměti <paměťové místo>	
CTABMPOL:	Stanovení počtu **maximálně možných** křivkových polynomů v zadané paměti <paměťové místo>	
<n>:	Číslo (**ID**) tabulky křivky	
<paměťové místo>:	Údaj místa v paměti (nepovinné)	
	"SRAM"	**statická** paměť NC systému
	"DRAM"	**dynamická** paměť NC systému
	Upozornění: Pokud pro tento parametr není naprogramována žádná hodnota, potom se použije standardní místo v paměti nastavené parametrem MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE.	

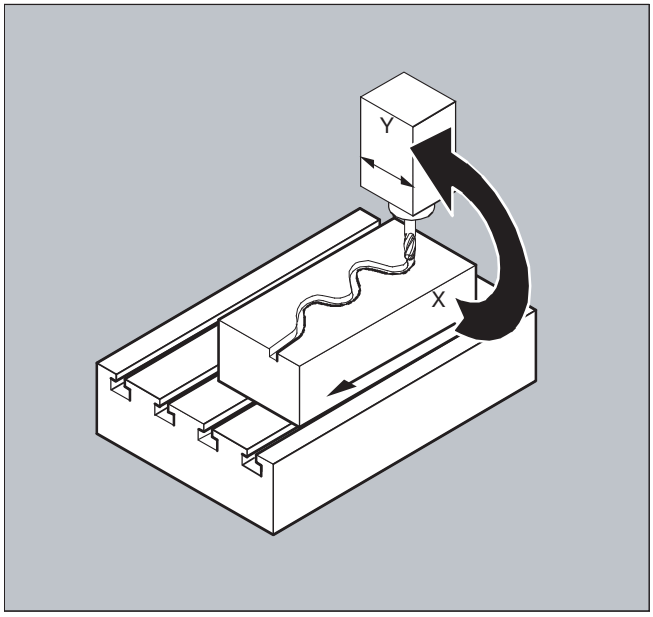
<druh úseku>:	Zadání druhu úseku (nepovinné)	
	"L"	Lineární úseky
	"P"	Polynomický úsek
	Upozornění: Jestliže pro tento parametr není naprogramována žádná hodnota, bude výsledkem součet lineárních a polynomických úseků.	

4.16.3 Osová vazba řídící hodnotou (LEADON, LEADOF)

Poznámka

U systému SINUMERIK 828D není tato funkce k dispozici!

V případě osové vazby řídící hodnotou se řídící a vlečná osa pohybují synchronně. Příslušná poloha vlečné osy je přitom prostřednictvím tabulky křivky, příp. z ní vypočítaného polynomu jednoznačně přiřazena poloze - v případě potřeby simulované - řídící osy.



Řídící osou je ta osa, která poskytuje vstupní hodnoty pro tabulku křivky. **Vlečnou osou** je nazývána osa, která zaujímá polohy vypočítané prostřednictvím tabulky křivky.

Vazba pomocí skutečné a požadované hodnoty

Jako řídící hodnoty se mohou používat také výstupní hodnoty pro zjišťování polohy vlečné osy.

- Skutečné hodnoty polohy řídící osy: Vazba pomocí skutečné hodnoty
- Požadované hodnoty polohy řídící osy: Vazba pomocí požadované hodnoty

Vazba pomocí řídící hodnoty platí vždy v základním souřadném systému.

Pokud budete potřebovat informace o tabulkách křivek, viz kapitola "Tabulky křivek".

Syntaxe

```
LEADON(<vlečná osa>,<řídící osa>,<n>)
LEADOF(<vlečná osa>,<řídící osa>)
```

nebo zrušení příkazem bez uvedení řídící osy:
`LEADOF(<vlečná osa>)`

Vazba řídící hodnotou může být aktivována a deaktivována jak z výrobního programu, tak také v průběhu pohybu ze synchronních akcí.

Význam

`LEADON`:	Aktivování vazby řídící hodnotou
`LEADOF`:	Deaktivování vazby řídící hodnotou
`<vlečná osa>`:	Vlečná osa
`<řídící osa>`:	Řídící osa
`<n>`:	Číslo tabulky křivky
`$SA_LEAD_TYPE`:	Přepnutí mezi vazbou pomocí požadované a skutečné hodnoty

Deaktivování vazby řídící hodnotou, LEADOF

Po deaktivování vazby řídící hodnotou se vlečná osa znovu stává normální příkazovou osou!

Osová vazba řídící hodnotou a různé provozní stavy, RESET

V závislosti na nastavení ve strojním parametru se vazby řídící hodnotou při RESETu vypínají.

Příklad vazby řídící hodnotou ze synchronní akce

U lisovacího zařízení má být obvyklá mechanická vazba mezi řídící osou (hřídel razníku) a osami podávacího systému, který se skládá z os podavače a pomocných os, nahrazena systémem elektronických vazeb.

Na tomto příkladu je ukázáno, jak se u lisovacího zařízení nahrazuje mechanický podávací systém elektronickým podávacím systémem. Operace vytváření a rozpojování vazeb jsou realizovány jako **statické synchronní akce**.

Na základě řídící osy LW (osa razníku) jsou definovaným způsobem prostřednictvím tabulek křivek řízeny osy podavače a pomocné osy jako vlečné osy.

Vlečné osy

X posuv, příp. podélná osa
YL zavírání, příp. příčná osa
ZL osa zdvihu
U posuv válce, pomocná osa
V vyrovnávací hlava, pomocná osa
W mazání, pomocná osa

Akce

V rámci synchronních akcí se uskutečňují např. následující činnosti:

- Aktivování vazby, `LEADON(<vlečná osa>,<řídící osa>,<číslo tabulky křivky>)`
- Deaktivování vazby, `LEADOF(<vlečná osa>,<řídící osa>)`
- Nastavení skutečné hodnoty, `PRESETON(<osa>, <hodnota>)`
- Nastavení značky, `$AC_MARKER[i] = <hodnota>`
- Druh vazby: reálná/virtuální řídící hodnota
- Najíždění osami na pozice, `POS[<osa>]=<hodnota>`

Podmínky

Jako podmínky jsou vyhodnocovány rychlé digitální vstupy, proměnné v reálném čase `$AC_MARKER` a porovnávání pozic, které jsou spojeny pomocí logického operátoru AND.

Poznámka

V následujícím příkladu byly řádkování, řazení bloků a bloky **mazání** použity výlučně k tomu účelu, aby se zvýšila čitelnost programu. Z hlediska řídícího systému je všechno pod jedním číslem bloku považováno za jeden řádek.

Komentář

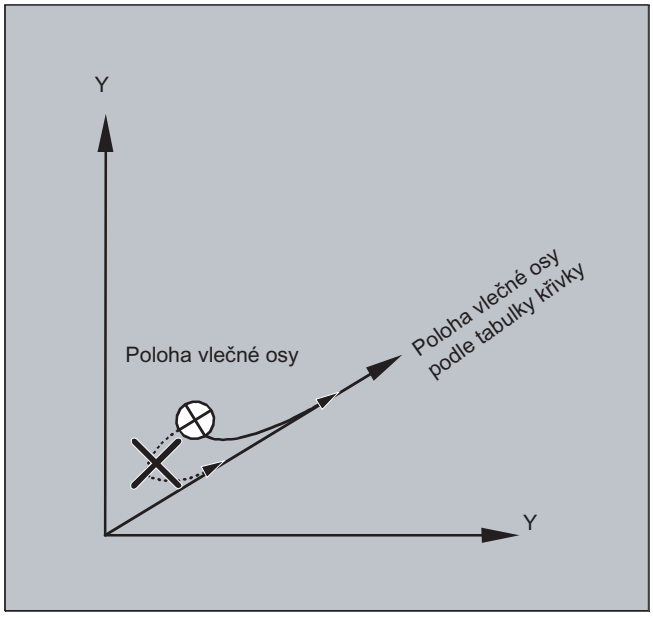
Programový kód	**Komentář**
	; definuje veškeré statické synchronní akce
	; ****Vynulování návěští
N2 $AC_MARKER[0]=0 $AC_MARKER[1]=0 $AC_MARKER[2]=0 $AC_MARKER[3]=0 $AC_MARKER[4]=0 $AC_MARKER[5]=0 $AC_MARKER[6]=0 $AC_MARKER[7]=0	
	; **** E1 0=>1 Zapnutí vazby podavače
N10 IDS=1 EVERY ($A_IN[1]==1) AND ($A_IN[16]==1) AND ($AC_MARKER[0]==0) DO LEADON(X,LW,1) LEADON(YL,LW,2) LEADON(ZL,LW,3) $AC_MARKER[0]=1	
	; **** E1 0=>1 Zapnutí vazby posuvu válce
N20 IDS=11 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[5]==0) DO LEADON(U,LW,4) PRESETON(U,0) $AC_MARKER[5]=1	
	; **** E1 0->1 Zapnutí vazby vyrovnávací hlavy
N21 IDS=12 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[6]==0) DO LEADON(V,LW,4) PRESETON(V,0) $AC_MARKER[6]=1	
	; **** E1 0->1 Zapnutí vazby mazání
N22 IDS=13 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[7]==0) DO LEADON(W,LW,4) PRESETON(W,0) $AC_MARKER[7]=1	
	; **** E2 0=>1 Vypnutí vazby
N30 IDS=3 EVERY ($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	

Popis

Vazba řídící hodnotou vyžaduje synchronizaci řídící a vlečné osy. Této synchronizace může být dosaženo jen tehdy, pokud se vlečná osa při zapnutí vazby řídící hodnotou nachází v rámci toleranční oblasti průběhu kontury vypočítaného z tabulky křivky.

Toleranční oblast pro polohu vlečné osy je definována prostřednictvím strojního parametru `MD 37200: COUPLE_POS_POL_COARSE A_LEAD_TYPE`.

Jestliže se vlečná osa v okamžiku zapnutí vazby řídící hodnotou ještě nenachází na odpovídající pozici, automaticky se uskuteční synchronizační operace, která zajistí, aby se vypočítaná požadovaná poloha pro vlečnou osu blížila její skutečné poloze. Vlečná osa se přitom v průběhu synchronizační operace pohybuje ve směru, který je definován prostřednictvím požadované hodnoty rychlosti vlečné osy (vypočítá se na základě rychlosti řídící osy a podle tabulky křivky CTAB).

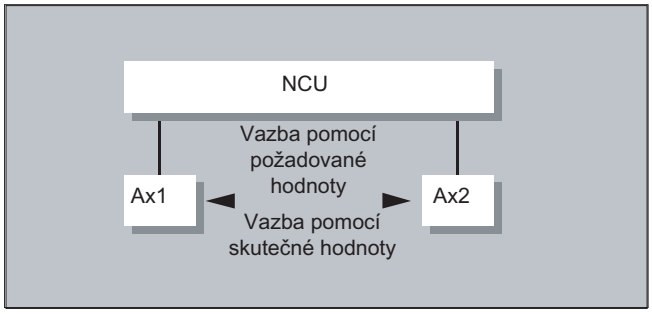


Žádná synchronizační operace

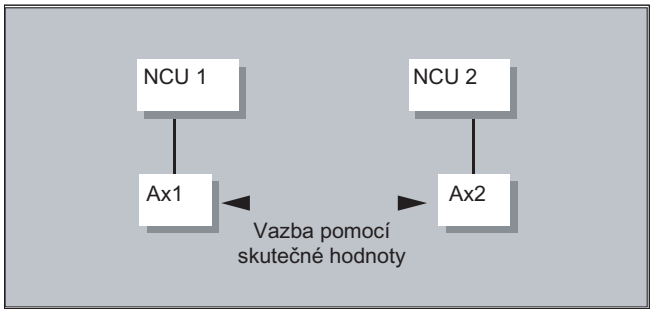
Jestliže se vypočítaná požadovaná poloha vlečné osy při zapnutí vazby řídící hodnotou vzdaluje od aktuální polohy vlečné osy, žádná synchronizační operace se neuskutečňuje.

Vazba pomocí skutečné a požadované hodnoty

Vazba požadovanou hodnotou poskytuje ve srovnání s vaznou pomocí skutečné hodnoty lepší synchronizaci mezi řídící a vlečnou osou a je proto předem nastavena jako standardní.



Vazba pomocí požadované hodnoty je možná jedině tehdy, pokud interpolace řídící a vlečné probíhá ve stejné NCU. V případě externí řídící osy může být vytvořena vazba mezi touto řídící osou a vlečnou osou jen pomocí skutečné hodnoty.



Přepnutí je možné uskutečnit pomocí nastavovaného parametru $SA_LEAD_TYPE.

Přepnutí mezi vazbou pomocí skutečné a požadované hodnoty by se mělo vždy provádět, když je vlečná osa v klidu. Důvodem je, že po přepnutí proběhne nová synchronizace jen v klidovém stavu.

Příklad použití

Může se stát, že při velkých otřesech stroje nebude čtení skutečné hodnoty bezchybné. Při použití vazby řídící hodnotou se v případě podavačů u lisů může ukázat jako nezbytné v pracovních krocích s velkými otřesy přepnout z vazby pomocí skutečné hodnoty na vazbu pomocí požadované hodnoty.

Simulace řídící hodnoty ve vazbách pomocí požadované hodnoty

Pomocí strojního parametru je možné oddělit interpolátor pro řídící osu od příslušného servomechanismu. Díky tomu je potom možné v případě vazby pomocí požadované hodnoty vytvořit požadovanou hodnotu, aniž by se řídící osa ve skutečnosti pohybovala.

Řídící hodnoty, které jsou vytvořeny prostřednictvím vazby pomocí požadované hodnoty a které se využívají např. v synchronních akcích, je možné načítat z následujících proměnných:

- $AA_LEAD_P	Řídící hodnota polohy
- $AA_LEAD_V	Řídící hodnota rychlosti

Vytváření řídících hodnot

Řídící hodnoty mohou být generovány pomocí libovolných posuvů, které jste sami naprogramovali. Tyto řídící hodnoty vytvořené tímto způsobem se zapisují do proměnných

- $AA_LEAD_SP	Řídící hodnota polohy
- $AA_LEAD_SV	Řídící hodnota rychlosti

a je možné je odtud také číst. Aby bylo možné tyto proměnné využívat, musí být nastavovanému parametru přiřazena hodnota $SA_LEAD_TYPE = 2.

Stav vazby

Ve výrobním programu v NC systému můžete pomocí následujících systémových proměnných zjistit stav vazby:

`$AA_COUP_ACT[osa]`
0: Žádná vazba není aktivní
16: Vazba řídící hodnotou je aktivní

Správa stavových proměnných při synchronních akcích

Spínací operace a operace s vazbami jsou spravovány pomocí proměnných v reálném čase:

`$AC_MARKER[i] = n`
, přičemž:
i = číslo značky
n = stavová hodnota

4.16.4 Elektronická převodovka (EG)

Pomocí funkce "Elektronická převodovka" je možné ovládat pohyb jedné **vlečné osy** po lineárním pohybovém bloku v závislosti na až pěti **řídících osách**. Vzájemné závislosti mezi řídícími osami a vlečnou osou jsou pro každou řídící osu zvlášť definovány pomocí faktoru vazby.

Vypočítaná pohybová složka vlečné osy se vypočítává součtem pohybových složek jednotlivých lineárních os vynásobených příslušnými faktory vazby. Při aktivování svazku os typu "Elektronická převodovka" se může ukázat jako nezbytné synchronizovat vlečnou osu na určitou definovanou pozici. S převodovou vazbou je možné ve výrobním programu uskutečňovat následující operace:

- definice
- aktivování
- deaktivování
- vymazání

.

Podle Vašeho přání může být pohyb vlečné osy odvozen od následujících veličin:

- Požadované hodnoty řídících os
- Skutečné hodnoty řídících os

Jako rozšíření mohou být prostřednictvím **tabulek křivek** (viz kapitola "Chování při pohybu po dráze) realizovány také nelineární vztahy mezi řídícími osami a vlečnou osou. Elektronické převodovky umožňují kaskádovité řazení, tzn. vlečná osa jedné elektronické převodovky se může stát řídící osou pro další elektronickou převodovku.

4.16.4.1 Definice elektronické převodovky (EGDEF)

Svazek os typu "Elektronická převodovka" je definován stanovením vlečné osy a minimálně jedné, avšak nejvýše pěti řídících os s příslušnými faktory vazby.

Předpoklady

Předpoklad pro definici svazku os typu "Elektronická převodovka":

Pro vlečnou osu nesmí být definována žádná další vazba mezi osami (v případě potřeby musí předem být předcházející vazba vymazána pomocí příkazu EGDEL).

Syntaxe

```
EGDEF(vlečná osa,řídící osa1,typ vazby1,řídící osa2,typ vazby2,...)
```

Význam

`EGDEF`:	Definice elektronické převodovky	
`Vlečná osa`:	Osa, která je řídícími osami ovlivňována	

`řídící osa1`
`,...,`
`řídící osa5`
	Osy, které ovlivňují vlečnou osu	

`typ vazby1`
`,...,`
`typ vazby5`
	Typ vazby
	Typ vazby nemusí být pro všechny řídící osy stejný a proto je potřeba jej zadat pro každou řídící osu zvlášť.	

	Hodnota:	**Význam:**
	0	Vlečná osa je ovlivňována **skutečnou hodnotou** odpovídající řídící osy.
	1	Vlečná osa je ovlivňována **požadovanou hodnotou** odpovídající řídící osy.

Poznámka

Při definici svazku os typu "Elektronická převodovka" jsou faktorům vazby dosazeny nuly.

Poznámka

Příkaz EGDEF zastavuje předběžné zpracování. Definici převodovky pomocí příkazu EGDEF je potřeba v nezměněné podobě používat také v případě systémů, u kterých jedna nebo více řídících os ovlivňují vlečnou osu na základě **tabulky křivky**.

Příklad:

Programový kód	Komentář
`EGDEF(C,B,1,Z,1,Y,1)`	`; definice svazku os typu "Elektronická převodovka" Řídící osy B, Z, Y ovlivňují vlečnou osu C pomocí požadované hodnoty.`

4.16.4.2 Zapnutí elektronické převodovky (EGON, EGONSYN, EGONSYNE)

Pro zapínání svazku os typu "Elektronická převodovka" existují 3 varianty:

Syntaxe

Varianta 1:

Svazek os typu "Elektronická převodovka" se zapíná selektivně bez synchronizace pomocí příkazu:

```
EGON(FA,"režim přechodu na další
blok",LA1,Z1,N1,LA2,Z2,N2,...,LA5,Z5,N5)
```

Varianta 2:

Svazek os typu "Elektronická převodovka" se zapíná selektivně se synchronizací pomocí příkazu:

```
EGONSYN(FA,"režim přechodu na další blok",SynPosFA,
[,LAi,SynPosLAi,Zi,Ni])
```

Varianta 3:

Svazek os typu "Elektronická převodovka" se zapíná selektivně se synchronizací a s předem zadaným režimem najíždění pomocí příkazu:

```
EGONSYNE(FA,"režim přechodu na další blok",SynPosFA,režim
najíždění[,LAi,SynPosLAi,Zi,Ni])
```

Význam

Varianta 1:

FA	Vlečná osa	
`Režim přechodu na další blok`:	Mohou se používat následující režimy:	
	`"NOC"`	Na další blok se přechází okamžitě
	`"FINE"`	Přechod na další blok při „jemném chodu synchronizace“
	`"COARSE"`	Přechod na další blok při „hrubém chodu synchronizace“
	`"IPOSTOP"`	Přechod na další blok při dosažení synchronizace s požadovanou hodnotou
LA1, ... LA5	Řídící osy	
Z1, ... Z5	Čitatel pro faktor vazby i	
N1, ... N5	jmenovatel pro faktor vazby i Faktor vazby i = čitatel i/jmenovatel i	

Smí být naprogramovány jedině řídící osy, které byly předtím specifikovány pomocí příkazu EGDEF. Musí být naprogramována nejméně jedna řídící osa.

Varianta 2:

FA	Vlečná osa	
Režim přechodu na další blok:	Mohou se používat následující režimy:	
	"NOC"	Na další blok se přechází okamžitě
	"FINE"	Přechod na další blok při „jemném chodu synchronizace“
	"COARSE"	Přechod na další blok při „hrubém chodu synchronizace“
	"IPOSTOP"	Přechod na další blok při dosažení synchronizace s požadovanou hodnotou
[,LAi,SynPosLAi,Zi,Ni]	(hranaté závorky nezapisovat) Minimálně. 1, max. 5 posloupností s těmito prvky:	
LA1, ... LA5	Řídící osy	
SynPosLAi	Synchronizační pozice pro i-tou řídící osu	
Z1, ... Z5	Čitatel pro faktor vazby i	
N1, ... N5	jmenovatel pro faktor vazby i Faktor vazby i = čitatel i/jmenovatel i	

Smí být naprogramovány jedině řídící osy, které byly předtím specifikovány pomocí příkazu EGDEF. Prostřednictvím naprogramovaných "Synchronizačních pozic" pro vlečnou osu (SynPosFA) a pro řídící osy (SynPosLA) jsou definovány polohy, ve kterých je svazek os považován za *synchronizovaný*. Jestliže se elektronická převodovka při svém aktivování nenachází v synchronizovaném stavu, najede vlečná osa na svou definovanou synchronizační pozici.

Varianta 3:

Parametry odpovídají variantě 2, a kromě toho ještě:

Způsob odjíždění:	Mohou se používat následující režimy:	
	"NTGT"	V následujícím mezizubí najíždět s časovou optimalizací
	"NTGP"	V následujícím mezizubí najíždět s dráhovou optimalizací
	"ACN"	Kruhovou osou najíždět v záporném směru otáčení na absolutní pozici
	"ACP"	Kruhovou osou najíždět v kladném směru otáčení na absolutní pozici
	"DCT"	Na naprogramovanou synchronizační pozici s časovou optimalizací
	"DCP"	Na naprogramovanou synchronizační pozici s dráhovou optimalizací

Varianta 3 se uplatňuje pouze u vlečných os typu Modulo, které jsou navázány na řídící osy typu Modulo. Časová optimalizace bere v úvahu mezní hodnoty rychlosti vlečné osy.

Další informace

Popis variant zapnutí

Varianta 1:

Pozice jak řídící osy, tak i vlečné osy v okamžiku aktivování se ukládají jako "synchronizační pozice". "Synchronizační pozice" mohou být načteny pomocí systémových proměnných `$AA_EG_SYN`.

Varianta 2:

Jestliže jsou ve svazku osy typu modulo, jsou hodnoty jejich poloh zmenšeny pomocí funkce Modulo. Tímto způsobem je zajištěno, že se najíždí na nejbližší možnou synchronizační polohu (tzv. *relativní synchronizace*: např. následující mezizubí). Jestliže pro vlečnou osu není nastaven signál rozhraní DB(30 + číslo osy), DBX 26 bit 4 "Uvolnění superpozice vlečné osy", nebude se na synchronní pozici najíždět. Místo toho se program bloku s příkazem EGONSYN pozastaví a aktivuje se alarm 16771 s automatickým potvrzováním, dokud není výše uvedený signál nastaven.

Varianta 3:

Vzdálenost mezi zuby vyplývá z následujícího vztahu: 360 * Zi/Ni. V případě, že je vlečná osa v okamžiku vyvolání zastavena, je chování v případě dráhové a časové optimalizace stejné.

Pokud se vlečná osa už pohybuje, pomocí příkazu NTGP se uskuteční synchronizace na následující mezizubí nezávisle na momentální rychlosti vlečné osy. Pokud se vlečná osa už pohybuje, pomocí příkazu NTGT se uskuteční synchronizace na následující mezizubí v závislosti na momentální rychlosti vlečné osy. Za tím účelem se osa v případě potřeby zabrzdí.

Tabulky křivek

Jestliže se pro jednu z řídících os používá **tabulka křivky**, pak musí být splněny tyto podmínky:

Ni	Jmenovatel faktoru lineární vazby musí být nastaven na 0. (Hodnota jmenovatele 0 by byla pro lineární vazby nepřípustná.) Nulový jmenovatel řídícímu systému signalizuje, že:
Zi	má být interpretováno jako číslo tabulky křivky ,která se má použít. Tabulka křivky se zadaným číslem musí být už v okamžiku aktivování vazby definována.
LAi	Zadání řídící osy odpovídá údaji řídící osy ve vazbě pomocí faktoru vazby (lineární vazba).

Pokud budete potřebovat další informace týkající se používání tabulek křivek a kaskádového řazení elektronických převodovek a jejich synchronizace, viz:
Další funkce:
Příručka k funkcím, Osy a vřetena

Chování elektronické převodovky při zapnutí (Power-On), resetu, změně provozního režimu, vyhledávání bloku

- Po zapnutí systému není aktivní **žádná** vazba.
- V případě resetu a změny provozního režimu zůstávají aktivní vazby zachovány.
- Při vyhledávání bloku nejsou příkazy pro aktivování, mazání a definice elektronických převodovek prováděny a nejsou ani shromažďovány, nýbrž jsou přeskakovány.

Systémové proměnné elektronických převodovek

Pomocí systémových proměnných elektronických převodovek mohou být ve výrobním programu zjišťovány aktuální stavy svazku os typu "Elektronická převodovka" a v případě nutnosti je možno na ně reagovat.

Systémové proměnné elektronických převodovek jsou označeny následujícím způsobem:

$AA_EG_ ...

nebo

$VA_EG_ ...

Další informace
Příručka k funkcím, Systémové proměnné

4.16.4.3 Vypnutí elektronické převodovky (EGOFS, EGOFC)

Pro vypínání aktivního svazku os typu "Elektronická převodovka" existují 3 varianty:

Programování

Varianta 1:

Syntaxe	Význam
`EGOFS(vlečná osa)`	Elektronická převodovka se vypne. Vlečná osa bude zabrzděna až do úplného klidu. Tento příkaz zastavuje předběžné zpracování.

Varianta 2:

Syntaxe	Význam
`EGOFS(vlečná osa,řídící osa1, ...,řídící osa5)`	Tyto programové příkazy umožňují **selektivně** odstraňovat vliv jednotlivých řídících os na pohyby vlečné osy.

Musí být uvedena nejméně jedna řídící osa. Vliv uvedených řídících os na vlečnou osu se cíleně vypne. Tento příkaz zastavuje předběžné zpracování. Pokud zůstávají ještě nějaké aktivní řídící osy, pohybuje se vlečná osa podle jejich působení i nadále. Jestliže jsou tímto způsobem zrušeny všechny vlivy řídících os, vlečná osa se zabrzdí do úplného klidu.

Varianta 3:

Syntaxe	Význam
`EGOFC(vlečná osa1)`	Elektronická převodovka se vypne. Vlečné vřeteno se pohybuje dál s otáčkami/rychlostí, které byly v platnosti v okamžiku vypnutí. Tento příkaz zastavuje předběžné zpracování.

Poznámka

Tato varianta je povolena jen pro vřetena.

4.16.4.4 Vymazání definice elektronické převodovky (EGDEL)

Svazek os typu "Elektronická převodovka" musí být vypnutý, jinak jeho definici není možné vymazat.

Programování

Syntaxe	Význam
`EGDEL(vlečná osa)`	Definice vazby svazku os se vymaže. Dokud není dosaženo maximálního počtu současně aktivních svazků os je možné pomocí příkazu `EGDEF` nově definovat další svazky os. Tento příkaz zastavuje předběžné zpracování.

4.16.4.5 Otáčkový posuv (G95) / elektronická převodovka (FPR)

Pomocí příkazu FPR může být i vlečná osa elektronické převodovky zadána jako osa, která určuje hodnotu otáčkového posuvu. V tomto případě se systém chová následujícím způsobem:

- Posuv je závislý na požadované hodnotě rychlosti vlečné osy elektronické převodovky.
- Požadovaná hodnota rychlosti se vypočítává z rychlostí řídícího vřetena a řídících os typu Modulo (které nejsou dráhovými osami) a z jim přiřazených faktorů vazby.
- Složky rychlosti z lineárních os, příp. z jiných os než typu Modulo, a superponované pohyby vlečné osy nejsou zohledňovány.

4.16.5 Synchronní vřeteno

V synchronním režimu tvoří řídící vřeteno (LS) a vlečné vřeteno (FS) tzv. **synchronní pár vřeten**. Když je aktivní vazba (synchronní režim), sleduje vlečné vřeteno pohyby řídícího vřetena v souladu s definovanou funkční závislostí.

Dvojice synchronních vřeten mohou být u každého stroje napevno konfigurovány pomocí specifických kanálových strojních parametrů, mohou být ale definovány také uživatelsky pomocí CNC výrobního programu. V jednom kanálu NC systému mohou být současně připraveny až 2 páry synchronních vřeten.

Vazba může být ve výrobním programu:

- definována, příp. upravována
- aktivována
- deaktivována
- vymazána

.

Kromě toho je možné v závislosti na stavu programového vybavení provádět tyto operace:

- Čekání na splnění podmínky pro synchronizaci
- Změna chování na přechodech mezi bloky

- Volba druhu vazby, tzn. buď vazba pomocí požadované nebo pomocí řídící hodnoty, nebo je možno zadat úhlové posunutí mezi řídícím a vlečným vřetenem
- Před aktivováním vazby lze převzít předcházející programové příkazy vlečného vřetena
- Lze provést korekci změřené nebo již známé odchylky synchronního režimu

.

4.16.5.1 Synchronní vřeteno: Programové příkazy (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC)

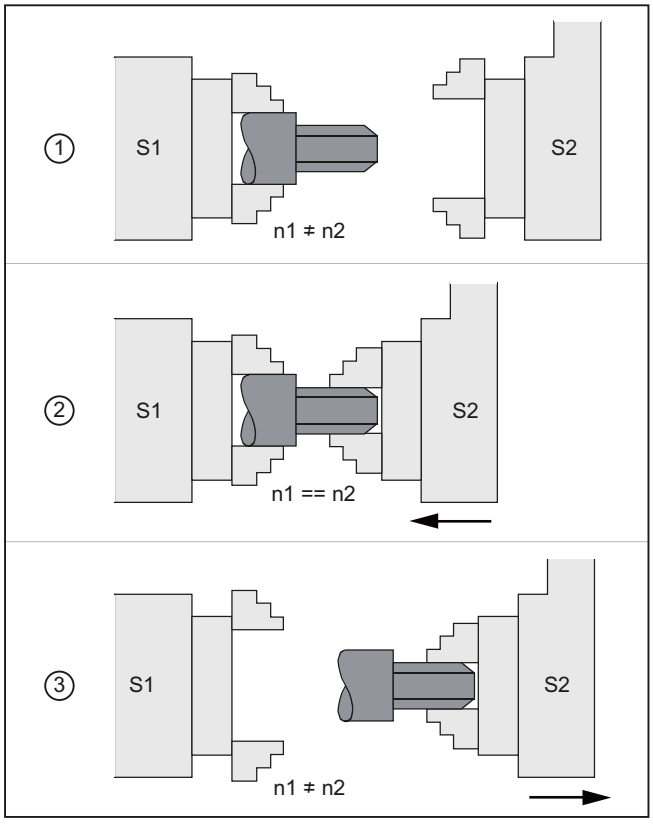
Funkce "Synchronní vřeteno" umožňuje ovládat pohyb vlečného vřetena (FS) podle řídícího vřetena (LS) se synchronizovanými otáčkami a s programovatelným převodovým poměrem.

Tato funkce nabízí následující režimy:

- Synchronizace otáček ($n_{FS} = n_{LS}$)
- Synchronizace polohy ($\phi_{FS} = \phi_{LS}$)
- Synchronizace polohy s úhlovým offsetem ($\phi_{FS} = \phi_{LS} + \Delta\phi$)

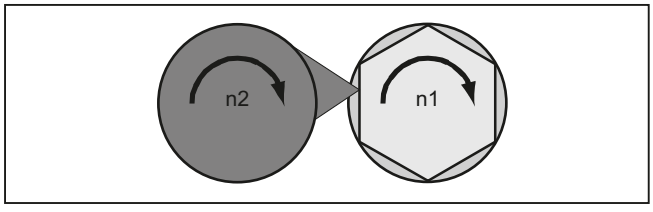
Příklady aplikace:

- Předávání obrobku za pohybu, např. kvůli opracování na zadní straně, převodový poměr: 1:1



① Synchronizace otáček
② Předání obrobku
③ Opracování na zadní straně

- Opracování mnohohranu (soustružení mnohoúhelníku), synchronizace otáček, převodový poměr: n_1:n_2



Syntaxe

```
COUPDEF(<FS>,<LS>,<ZFS>,<NLS>,<přechod na další blok>,<druh vazby>)
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>,<přechod na další blok>,<LS>,<přechod na další blok>)
```

Poznámka

Zkrácený způsob zápisu

V případě příkazů COUPOF, COUPOFS, COUPRES a COUPDEL je možný také zkrácený způsob zápisu bez udání řídícího vřetena.

Význam

COUPDEF:	Definice/editace specifické uživatelské vazby
COUPON:	Aktivování vazby. Vlečné vřeteno se synchronizuje podle řídícího vřetena, přičemž se vychází z momentálních otáček.
COUPONC:	Vazba při svém aktivování převezme předcházející naprogramované příkazy M3 S... nebo M4 S.... Rozdíl otáček vlečného vřetena se okamžitě převezme.
COUPOF:	Deaktivování vazby. • s okamžitým přechodem na další blok: COUPOF(<S2>,<S1>) • Přechod na další blok teprve po přejetí vypínací pozice <POSFS>, příp. <POSLS>: COUPOF(<S2>,<S1>,<POSFS>) COUPOF(<S2>,<S1>,<POSFS>,<POSLS>)
COUPOFS:	Vypnutí vazby se zastavením vlečného vřetena. Co možno nejrychlejší přechod na další blok. COUPOFS(<S2>,<S1>) Přechod na další blok teprve po přejetí vypínací pozice: COUPOFS(<S2>,<S1>,<POSFS>)
COUPRES:	Reset parametrů vazby na hodnoty nastavené v konfiguraci pomocí MD a SD
COUPDEL:	Vymazání uživatelem definované vazby
WAITC:	Čekání na splnění synchronizační podmínky (NOC se při přechodu na další blok uloží do IPO)
<FS>:	Popis vlečného vřetena
Volitelné parametry:	
<LS>:	Označení řídícího vřetena Zadání čísla vřetena: např. S2, S1
<ZFS>, <NLS>:	Převodový poměr mezi FS a LS. <ZFS> / <NLS> = čitatel / jmenovatel Předdefinované nastavení: <ZFS> / <NLS> = 1.0 ; uvedení jmenovatele je nepovinné

<přechod na další blok>:	Chování při přechodu na další blok Přechod na další blok se uskuteční:	
	"NOC"	okamžitě
	"FINE"	při dosažení "jemného chodu synchronizace"
	"COARSE"	při dosažení "hrubého chodu synchronizace"
	"IPOSTOP"	při dosažení zastavení interpolátoru (IPOSTOP), tzn. po dosažení synchronizace s požadovanou hodnotou (předdefinované nastavení)
	Chování při přechodu na další blok má modální působnost.	
<druh vazby>:	Druh vazby: Vazba mezi FS a LS.	
	"DV"	Vazba pomocí požadované hodnoty (předdefinované nastavení)
	"AV"	Vazba pomocí skutečné hodnoty
	"VV"	Vazba pomocí rychlosti
	Druh vazby má modální působnost.	
<POSFS>:	Úhlové posunutí mezi řídícím a vlečným vřetenem	
	Rozsah hodnot:	0°... 359,999°
<POSFS>,<POSLS>:	Vypínací pozice vlečného a řídícího vřetena "Přechod na další blok se odblokuje až po přejetí pozic POS_{FS}, POS_{LS}."	
	Rozsah hodnot:	0°... 359,999°

Příklady

Práce s řídícím a vlečným vřetenem

Programový kód	**Komentář**
	Řídící vřeteno = Hlavní vřeteno = Vřeteno 1
	Vlečné vřeteno = Vřeteno 2
N05 M3 S3000 M2=4 S2=500	Řídící vřeteno se otáčí s otáčkami 3000 ot/min, vlečné vřeteno se otáčí s otáčkami 500 ot/min.
N10 COUPDEF(S2,S1,1,1,"NOC","Dv")	Definice vazby (může být nastavena také v konfiguraci).
...	
N70 SPCON	Řídící vřeteno v režimu polohové regulace (vazba pomocí požadované hodnoty).
N75 SPCON(2)	Vlečné vřeteno v režimu polohové regulace.
N80 COUPON(S2,S1,45)	Aktivování vazby za pohybu s offsetovou pozicí = 45 stupňů.
...	
N200 FA[S2]=100	Rychlost polohování = 100 stupňů/min
N205 SPOS[2]=IC(-90)	Najetí na superponovanou pozici 90 stupňů v záporném směru.
N210 WAITC(S2,"Fine")	Čekání na "jemné chod synchronizace".
N212 G1 X... Y... F...	Opracování
...	
N215 SPOS[2]=IC(180)	Najetí na superponovanou pozici 180 stupňů v kladném směru.

Programový kód	Komentář
N220 G4 S50	Doba prodlevy = 50 otáček hlavního vřetena
N225 FA[S2]=0	Aktivování rychlosti nastavené v konfiguraci (MD).
N230 SPOS[2]=IC(-7200)	20 otáček. Pohyb s rychlosti nastavenou v konfiguraci v záporném směru.
...	
N350 COUPOF(S2,S1)	Zrušení vazby za pohybu, S=S2=3000
N355 SPOSA[2]=0	Zastavení FS v poloze nula stupňů.
N360 G0 X0 Y0	
N365 WAITS(2)	Čekání na vřeteno 2.
N370 M5	Zastavení FS.
N375 M30	

Programování rozdílných otáček

Programový kód	Komentář
	Řídící vřeteno = Hlavní vřeteno = Vřeteno 1
	Vlečné vřeteno = Vřeteno 2
N01 M3 S500	Řídící vřeteno se otáčí s otáčkami 500 ot/min.
N02 M2=3 S2=300	vlečné vřeteno se otáčí s otáčkami 300 ot/min.
...	
N10 G4 F1	Doba prodlevy pro hlavní vřeteno.
N15 COUPDEF (S2,S1,-1)	Faktor vazby s převodovým poměrem -1:1
N20 COUPON (S2,S1)	Aktivování vazby. Otáčky vlečného vřetena vyplývají z otáček řídícího vřetena a z faktoru vazby.
...	
N26 M2=3 S2=100	Programování rozdílných otáček.

Příklady převzetí pohybu pro rozdílné otáčky

1. Aktivování vazby pomocí příkazu COUPON, když byly předtím pro vlečné vřeteno naprogramovány nějaké příkazy

Programový kód	Komentář
	Řídící vřeteno = Hlavní vřeteno = Vřeteno 1
	Vlečné vřeteno = Vřeteno 2
N05 M3 S100 M2=3 S2=200	Řídící vřeteno se otáčí s otáčkami 100 ot/min, vlečné vřeteno s otáčkami 200 ot/min.
N10 G4 F5	Doba prodlevy = 5 sekund hlavního vřetena
N15 COUPDEF(S2,S1,1)	Převodový poměr FS ku LS je 1,0 (předdefinované nastavení)
N20 COUPON (S2,S1)	Aktivování vazby na řídící vřeteno za pohybu.
N10 G4 F5	vlečné vřeteno se otáčí s otáčkami 100 ot/min.

2. Aktivování vazby pomocí příkazu COUPONC, když byly předtím pro vlečné vřeteno naprogramovány nějaké příkazy

Programový kód	Komentář
	Řídící vřeteno = Hlavní vřeteno = Vřeteno 1
	Vlečné vřeteno = Vřeteno 2
N05 M3 S100 M2=3 S2=200	Řídící vřeteno se otáčí s otáčkami 100 ot/min, vlečné vřeteno s otáčkami 200 ot/min.
N10 G4 F5	Doba prodlevy = 5 sekund hlavního vřetena
N15 COUPDEF(S2,S1,1)	Převodový poměr FS ku LS je 1,0 (předdefinované nastavení)
N20 COUPONC(S2,S1)	Aktivování vazby na řídící vřeteno za pohybu a převzetí předcházejících otáček pro S2.
N10 G4 F5	S2 se otáčí s rychlostí 100 ot/min + 200 ot/min = 300 ot/min

3. Aktivování vazby pomocí příkazu COUPON, když vlečné vřeteno stojí

Programový kód	Komentář
	Řídící vřeteno = Hlavní vřeteno = Vřeteno 1
	Vlečné vřeteno = Vřeteno 2
N05 SPOS=10 SPOS[2]=20	Vlečné vřeteno S2 v polohovacím režimu.
N15 COUPDEF(S2,S1,1)	Převodový poměr FS ku LS je 1,0 (předdefinované nastavení)
N20 COUPON (S2,S1)	Aktivování vazby na řídící vřeteno za pohybu.
N10 G4 F1	Vazba je vytvořena, S2 zůstává stát v poloze 20 stupňů.

4. Aktivování vazby pomocí příkazu COUPONC, když vlečné vřeteno stojí

Poznámka

Polohovací nebo osový režim

Jestliže se vlečné vřeteno před zapojením do vazby nachází v polohovacím nebo v osovém režimu, potom se toto vlečné vřeteno chová v případě příkazů COUPON(<FS>,<LS>) a COUPONC(<FS>,<LS>) stejně.

Poznámka

Řídící vřeteno a osový režim

Jestliže se řídící vřeteno nachází před definicí vazby v režimu osy, po aktivování vazby se bude uplatňovat mezní hodnota rychlosti ze strojního parametru:

MD32000 $MA_MAX_AX_VELO (maximální rychlost osy)

Aby se tomuto chování zabránilo, musí být osa před definicí vazby přepnuta do režimu vřetena (M3 S... nebo M4 S...).

Další informace

Vazba nastavená v konfiguraci

V případě vazby nastavené v konfiguraci jsou řídící (LS) a vlečné (FS) vřeteno definovány pomocí strojního parametru. Vřetena nastavená v konfiguraci mohou být ve výrobním programu změněna. Parametry vazby je možno ve výrobním programu nastavovat pomocí příkazu `COUPDEF` (předpoklad: ochrana proti zápisu není aktivována).

Uživatelem definovaná vazba

Pomocí příkazu `COUPDEF` může být vazba ve výrobním programu nově definována nebo změněna. Jestliže už je nějaká vazba aktivní, musí být před definicí nové vazby napřed vymazána pomocí příkazu `COUPDEL`.

Vazba je úplně definována pomocí následujícího příkazu:

```
COUPDEF(<FS>,<LS>,<ÜFS>,<ÜLS>, chování na přechodu mezi bloky, druh vazby)
```

Vlečné vřeteno (FS) a řídící vřeteno (LS)

Pomocí názvů os pro vlečné (FS) a řídící (LS) vřeteno je vazba jednoznačně definována. Názvy os musí být naprogramovány u každého příkazu `COUPDEF`. Ostatní parametry vazby mají modální platnost a musí být naprogramovány jedině tehdy, pokud jsou měněny.

Příklad:

```
COUPDEF(S2,S1)
```

Převodový poměr

Převodový poměr se udává jako poměr otáček mezi vlečným (FS) a řídícím (LS) vřetenem:

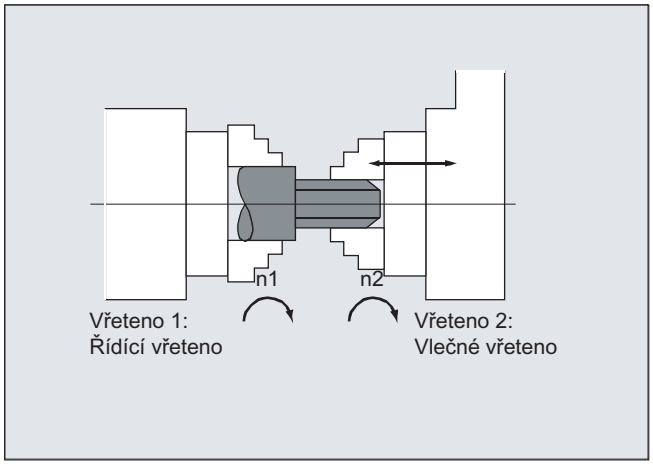
Vlečné vřeteno / řídící vřeteno = čitatel / jmenovatel

Čitatel musí být naprogramován. Jmenovatel nemusí být naprogramován. Pro jmenovatele se potom dosazuje předdefinovaná hodnota 1,0.

Příklad:

Vlečné vřeteno S2 a řídící vřeteno S1, převodový poměr = 1 / 1

```
COUPDEF(S2, S1, 1.0)
```



Poznámka

Převodový poměr může být upravován i v době, kdy je vazba aktivní a vřetena se otáčejí.

Chování na přechodech mezi bloky NOC, FINE, COARSE, IPOSTOP

Při programování chování na přechodech mezi bloky jsou možné následující zkrácené způsoby zápisu:

- `"NO"`: okamžitě (předdefinované nastavení)
- `"FI"`: při dosažení "jemného chodu synchronizace"
- `"CO"`: při dosažení "hrubého chodu synchronizace"
- `"IP"`: při dosažení zastavení interpolátoru (IPOSTOP), tzn. po dosažení synchronizace s požadovanou hodnotou

Druh vazby

Poznámka

Druh vazby se smí měnit jedině tehdy, když je vazba deaktivovaná.

Aktivování synchronizačního režimu, COUPON, <POSFS>

- Aktivování vazby s libovolným úhlovým offsetem mezi osami LS a FS:
  - `COUPON(S2,S1)`
  - `COUPON(S2)`
- Aktivování vazby s úhlovým offsetem `<POSFS>`
  Příkaz `<POSFS>` se vztahuje na polohu 0° řídícího vřetena v kladném směru otáčení
  Rozsah hodnot pro příkaz `<POSFS>`: 0°... 359,999°
  - `COUPON(S2,S1,30)`

Poznámka

Úhlový offset je možné změnit, i když je vazba aktivní.

Polohování vlečného vřetena

I když je aktivována vazba mezi synchronními vřeteny, je možno provádět nastavování polohy FS v rozsahu ±180°, a to nezávisle na LS.

- Nastavování polohy vlečného vřetena (FS) pomocí příkazu `SPOS`
  Příklad: `SPOS[2]=IC(-90)`
  Pokud budete potřebovat podrobnější informace týkající se příkazu `SPOS`, viz:
  Další informace
  Nastavování polohy vřetena (SPOS, SPOSA, M19, M70, WAITS) (Strana 118)

Rozdíl otáček

Rozdíl otáček v režimu regulace otáček a s aktivní vazbou mezi synchronními vřeteny vzniká prostřednictvím sečtení s příslušným znaménkem otáček FS na základě pohybu LS a otáček FS na základě naprogramovaných příkazů pro vřeteno:

- Vazba mezi synchronními vřeteny `COUPONC`
- `S<FS>=<otáčky> [ M<FS>=<směr otáčení>]`

Poznámka

Okrajové podmínky

- Při změně směru otáčení `M3` / `M4` musí být také znovu naprogramovány otáčky `S...`
- Sčítání otáček vřetena (`M<směr otáčení> S<FS>`) prostřednictvím pohybu LS v případě vazby synchronních vřeten `COUPONC` je v platnosti jen tehdy, pokud je superpozice tohoto druhu povolena.
- Dynamika řídícího vřetena musí být omezena natolik, aby při superpozici vlečného vřetena nemohlo dojít k překročení mezních hodnot jeho dynamiky.

Pokud budete potřebovat další informace týkající se rozdílu otáček, viz:
Další informace
Příručka k funkcím, Osy a vřetena

Rychlost, zrychlení: FA, ACC, OVRA, VELOLIMA

Rychlost a zrychlení osy vlečného vřetena mohou být naprogramovány pomocí těchto příkazů:

- `FA[SPI(S<n>)]`, příp. `FA[S<n>]` (axiální rychlost)
- `ACC[SPI(S<n>)]`, příp. `ACC[S<n>]` (axiální zrychlení)
- `OVRA[SPI(S<n>)]`, příp. `OVRA[S<n>]` (axiální korekce (override))
- `VELOLIMA[SPI(S<n>)]`, příp. `VELOLIMA[S<n>]` (axiální zvýšení, příp. snížení rychlosti)

Kde `<n>` = 1, 2, 3, ... (číslo vlečného vřetena)

Další informace
Regulace posuvu (Strana 106)

Poznámka

Zmenšení nebo zvětšení maximálního ryvu pro danou osu nemá u vřeten žádný význam.

Pokud budete potřebovat další informace týkající se dynamiky os, viz:
Další informace
Příručka k funkcím, Osy a vřetena

Programovatelné chování při přechodech mezi bloky, WAITC

Pomocí příkazu `WAITC` může být zadáno chování na přechodech mezi bloky, např. po změně parametrů vazby nebo při polohovacích operacích, s různými podmínkami chování synchronizace (hrubá, jemná, `IPOSTOP`). Jestliže není žádná podmínka chodu synchronizace uvedena, platí chování při přechodech mezi bloky zadané při definici v příkazu `COUPDEF`.

Příklady

- Čekání na dosažení podmínky chodu synchronizace `FINE` u vlečného vřetena S2 a `COARSE` u vlečného vřetena S4: `WAITC(S2,"FINE",S4,"COARSE")`
- Čekání na dosažení podmínky synchronizovaného chodu v souladu s příkazem `COUPDEF`: `WAITC( )`

Deaktivování vazby COUPOF

Pomocí příkazu `CUOPOF` je možné definovat chování při deaktivování vazby:

- Vypnutí vazby s okamžitým přechodem na další blok:
  - `COUPOF(S2,S1)` (s uvedením řídícího vřetena)
  - `COUPOF(S2)` (bez uvedení řídícího vřetena)
- Deaktivování vazby po přejetí vypínací pozice. Přechod na následující blok se provede po přejetí vypínací pozice.
  - `COUPOF(S2,S1,150)` (vypínací pozice FS: 150°)
  - `COUPOF(S2,S1,150,30)` (vypínací pozice FS: 150°, LS: 30°)

Deaktivování vazby se zastavením vlečného vřetena, COUPOFS

Pomocí příkazu `CUOPOFS` je možné definovat chování při deaktivování vazby se zastavením vlečného vřetena:

- Vypnutí vazby se zastavením vlečného vřetena a s okamžitým přechodem na další blok:
  - `COUPOFS(S2,S1)` (s uvedením řídícího vřetena)
  - `COUPOFS(S2)` (bez uvedení řídícího vřetena)
- Deaktivování vazby po přejetí vypínací pozice se zastavením vlečného vřetena. Přechod na následující blok se provede po přejetí vypínací pozice.
  - `COUPOFS(S2,S1,150)` (vypínací pozice FS: 150°)

Vymazání vazby, COUPDEL

Pomocí příkazu `COUPDEL` se vazba vymaže:

- `COUPDEL(S2,S1)` (s uvedením řídícího vřetena)
- `COUPDEL(S2)` (bez uvedení řídícího vřetena)

Obnovení původního nastavení parametrů vazby, COUPRES

Pomocí příkazu `COUPRES` je možno aktivovat hodnoty parametrů vazby, které jsou uloženy ve strojních a nastavovaných parametrech:

- `COUPRES(S2,S1)` (s uvedením řídícího vřetena)
- `COUPRES(S2)` (bez uvedení řídícího vřetena)

Systémové proměnné

- Aktuální stavové informace o vazbě vlečného vřetena
  Aktuální stavové informace o vazbě vlečného vřetena je možné načítat jako bitově kódované pomocí:
  `<hodnota> = $AA_COUP_ACT[<FS>]`

Bit	<hodnota>	Význam
-	0	žádná vazba není aktivní
2	4	synchronní vazba mezi vřeteny je aktivní
Poznámka • Všechny ostatní hodnoty se vztahují na režim osy • Jestliže je vřeteno vlečným vřetenem pro větší počet vazeb, bude systém poskytovat jako hodnotu stavu všech vazeb jejich součtový stav.		

- Aktuální úhlový offset
  Aktuální úhlový offset vlečného vřetena vůči řídícímu vřetenu je možné načítat pomocí:
  - `$AA_COUP_OFFS[<FS>]` (úhlový offset vztahující se na požadovanou hodnotu)
  - `$VA_COUP_OFFS[<FS>]` (úhlový offset vztahující se na skutečnou hodnotu)

  Příklad použití
  Korekce rozdílu úhlového offsetu v NC programu po deaktivování režimu vlečení:
  Rozdíl úhlového offsetu = naprogramovaný úhlový offset - systémová proměnná

Další informace

Pokud budete potřebovat podrobné informace týkající se systémových proměnných, viz:

Příručka pro sestavy systémových proměnných

4.16.6 Generická vazba (CP...)

"Generická vazba" je všeobecná funkce pro vytváření vazeb, ve které jsou zahrnuty všechny vlastnosti již existujících druhů vazby (vlečení, vazba řídící hodnotou, elektronická převodovka a synchronní vřeteno).

Tato funkce umožňuje flexibilní způsob programování:

- Pro své aplikace může uživatel volně vybírat vlastnosti vazeb, které jsou zapotřebí (modulární princip).
- Každá vlastnost vazby je samostatně programovatelná.
- Vlastnosti vazby, která již byla definována (např. faktor vazby), je možné změnit.
- Je možné také pozdější využití dalších vlastností vazby.
- Vztažný souřadný systém pro vlečnou osu (základní souřadný systém nebo souřadný systém stroje) může být naprogramován.
- Určité vlastnosti vazeb mohou být programovány i v synchronních akcích.
  Další informace: Příručka k funkcím, Synchronní akce

Poznámka

Až dosud bylo volání vazby pro vlečení (TRAIL*), vazbu řídící hodnotou (LEAD*), elektronickou převodovku (EG*) a synchronní vřeteno (COUP*) dále podporováno pomocí přizpůsobovacích cyklů.

Přehled všech klíčových slov a vlastností vazeb

Následující tabulka ukazuje přehled všech klíčových slov generické vazby a vlastností vazby, jež mohou být jimi naprogramovány:

Klíčové slovo	Vlastnost vazby / význam	Syntaxe		
CPDEF	Definice režimu vazby	CPDEF=(<FAx>)		
CPDEL	Vymazání režimu vazby	CPDEL=(<FAx>)		
CPLA	Definice řídící osy	CPLA[<FAx>]=(<LAx>)		
CPLDEF	Definice řídící osy a definice režimu osy (možné také pomocí příkazů CPDEF + CPLA)	CPLDEF[<FAx>]=(<LAx>) nebo CPDEF=(<FAx>) CPLA[<FAx>]=(<LAx>)		
CPLDEL	Vymazání řídící osy a režimu vazby (možné také pomocí příkazů CPDEL + CPLA)	CPLDEL[<FAx>]=(<LAx>) nebo CPDEL=(<FAx>) CPLA[<FAx>]=(<LAx>)		
CPON	Aktivování režimu vazby	CPON=(<FAx>)		
CPOF	Deaktivování režimu vazby	CPOF=(<FAx>)		
CPLON	Aktivování řídící osy režimu vazby	CPLON[<FAx>]=<LAx>		
CPLOF	Deaktivování řídící osy režimu vazby	CPLOF[<FAx>]=<LAx>		
CPLNUM	Čitatel faktoru vazby	CPLNUM[FAx,LAx]=<hodnota>		
CPLDEN	Jmenovatel faktoru vazby	CPLDEN[FAx,LAx]=<hodnota>		
CPLCTID	Číslo tabulky křivky	CPLCTID[FAx,LAx]=<hodnota>		
CPLSETVAL	Vztažný objekt vazby	CPLSETVAL[FAx,LAx]="<Vztažný objekt vazby>"		
		"<Vztažný objekt vazby>":	"CMDPOS"	Vazba pomocí požadované hodnoty
			"CMDVEL"	Vazba pomocí rychlosti
			"ACTPOS"	Vazba pomocí skutečné hodnoty
CPFRS	Vztažný souřadný systém	CPFRS[FAx]="<vztažný souřadný systém>"		
		"<vztažný souřadný systém>":	"BCS"	Základní souřadný systém
			"MCS"	Souřadný systém stroje

Klíčové slovo	Vlastnost vazby / význam	Syntaxe		
CPBC	Kritérium přechodu na další blok	CPBC[FAx]="<kritérium přechodu na další blok>"		
		"<Kritérium přechodu na další blok>":	"NOC"	Přechod na další blok se uskutečňuje nezávisle na stavu vazby.
			"IPOSTOP"	Přechod na další blok se uskutečňuje při dosažení synchronizace s požadovanou hodnotou.
			"COARSE"	Přechod na další blok se uskutečňuje při dosažení synchronizace s "hrubou" skutečnou hodnotou.
			"FINE"	Přechod na další blok se uskutečňuje při dosažení synchronizace s "jemnou" skutečnou hodnotou.
CPFPOS + CPON	Synchronní poloha vlečné osy při zapnutí	CPON=FAx CPFPOS[FAx]=<hodnota>		
CPLPOS + CPON	Synchronní poloha řídící osy při zapnutí	CPLPOS[FAx,LAx]=<hodnota>		

Klíčové slovo	Vlastnost vazby / význam	Syntaxe		
CPFMSON	Režim synchronní akce	CPFMSON[FAx]="<režim synchronní akce>"		
		"<režim synchronní akce>":	"CFAST"	Vazba se ukončí s časovou optimalizací.
			"CCOARSE"	Vazba se aktivuje až tehdy, když se poloha vlečné osy vyžadovaná podle předpisu, kterým se vazba řídí, nachází v oblasti aktuální polohy vlečné osy.
			"NTGT"	Na následující mezeru mezi zuby se najíždí s časovou optimalizací.
			"NTGP"	Na následující mezeru mezi zuby se najíždí s dráhovou optimalizací.
			"NRGT"	Na následující segment se najíždí s časovou optimalizací podle poměru počtu chodů ku počtu zubů.
			"NRGP"	Na následující segment se najíždí s dráhovou optimalizací podle poměru počtu chodů ku počtu zubů.
			"ACN"	Jen u kruhových os! Kruhová osa najíždí ve svém záporném směru na synchronizovanou polohu. Synchronizace se uskutečňuje okamžitě.
			"ACP"	Jen u kruhových os! Kruhová osa najíždí ve svém kladném směru na synchronizovanou polohu. Synchronizace se uskutečňuje okamžitě.
			"DCT"	Jen u kruhových os! Kruhová osa najíždí s časovou optimalizací na naprogramovanou synchronizovanou polohu. Synchronizace se uskutečňuje okamžitě.
			"DCP"	Jen u kruhových os! Kruhová osa najíždí s dráhovou optimalizací na naprogramovanou synchronizovanou polohu. Synchronizace se uskutečňuje okamžitě.

Klíčové slovo	Vlastnost vazby / význam	Syntaxe		
CPFMON	Chování vlečné vazby při aktivování	CPFMON[FAx]="<chování při aktivování>"		
		"<chování při aktivování>":	"STOP"	Jen u vřeten! Aktivní pohyb vlečného vřetena se před aktivováním zastaví.
			"CONT"	Jen u vřeten a os v hlavní zpracovávané větvi! Aktuální pohyb vlečné osy/vlečného vřetena se přebírá do vazby jako počáteční pohyb.
			"ADD"	Jen u vřeten! Pohybové složky vazby se uplatňují navíc k aktuálnímu superponovanému pohybu, tzn. aktuální pohyb vlečné osy/vlečného vřetena zůstává zachován jako superponovaný pohyb.
CPFMOF	Chování vlečné osy při úplném deaktivování	CPFMOF[FAx]="<chování při deaktivování>"		
		"<chování při deaktivování>":	"STOP"	Zastavení vlečné osy/vlečného vřetena Aktivní superponovaný pohyb bude zabrzděn až do klidu. Potom se vazba deaktivuje.
			"CONT"	Jen u vřeten a os v hlavní zpracovávané větvi! Vlečné vřeteno se pohybuje dál s otáčkami/rychlostí, které byly v platnosti v okamžiku vypnutí.
CPFPOS + CPOF	Vypínací pozice vlečné vazby při deaktivování	CPOF=(FAx) CPFPOS[FAx]=<hodnota>		

Klíčové slovo	Vlastnost vazby / význam	Syntaxe		
CPMRESET	Chování vazby při resetu	CPMRESET[FAx]="<chování při resetu>"		
		"<chování při resetu>":	"NONE"	Aktuální stav vazby zůstává zachován.
			"ON"	Pokud je definován odpovídající modul vazby, vazba se aktivuje. Aktivují se všechny definované vztahy s řídícími osami. To se uskuteční i tehdy, když již všechny nebo některé z těchto vztahů řídících os jsou již aktivovány, tzn. také v případě zcela aktivované vazby se uskuteční nová synchronizace.
			"OF"	Aktivní superponovaný pohyb bude zabržděn až do klidu. Potom se vazba deaktivuje. Jestliže byl definován odpovídající modul vazby bez explicitní definice (CPDEF), modul vazby se vymaže. Pokud tomu tak není, zůstává i nadále definován, tzn. je možno jej opět využívat.
			"OFC"	Může být použito jen u vřeten! Vlečné vřeteno se pohybuje dál s otáčkami/rychlostí, které byly v platnosti v okamžiku vypnutí. Vazba se deaktivuje. Jestliže byl definován odpovídající modul vazby bez explicitní definice (CPDEF), modul vazby se vymaže. Pokud tomu tak není, zůstává i nadále definován, tzn. je možno jej opět využívat.
			"DEL"	Aktivní superponovaný pohyb bude zabržděn až do klidu. Potom se vazba deaktivuje a následně se vymaže.
			"DELC"	Může být použito jen u vřeten! Vlečné vřeteno se pohybuje dál s otáčkami/rychlostí, které byly v platnosti v okamžiku vypnutí. Vazba se deaktivuje a následně se vymaže.

Klíčové slovo	Vlastnost vazby / význam	Syntaxe		
CPMSTART	Chování vazby při spuštění výrobního programu	CPMSTART[FAx]="<chování při spuštění>"		
		"<chování při spuštění>":	"NONE"	Aktuální stav vazby zůstává zachován.
			"ON"	Vazba se aktivuje. Aktivují se všechny definované vztahy s řídícími osami. To se uskuteční i tehdy, když již všechny nebo některé z těchto vztahů řídících os jsou již aktivovány, tzn. také v případě zcela aktivované vazby se uskuteční nová synchronizace.
			"OF"	Vazba se deaktivuje. Jestliže byl definován odpovídající modul vazby bez explicitní definice (CPDEF), modul vazby se vymaže. Pokud tomu tak není, zůstává i nadále definován, tzn. je možno jej opět využívat.
			"DEL"	Vazba se deaktivuje a následně se vymaže.
CPMPRT	Chování vazby při spuštění výrobního programu pomocí hledání bloku prostřednictvím zkušebního zpracování programu	CPMPRT[FAx]="<chování při spuštění>"		
		"<chování při spuštění>":	viz CPMSTART	
CPLINTR	Hodnota posunutí pro vstupní hodnotu řídící osy	CPLINTR[FAx,LAx]=<hodnota>		
CPLINSC	Faktor měřítka pro vstupní hodnotu řídící osy	CPLINSC[FAx,LAx]=<hodnota>		
CPLOUTTR	Hodnota posunutí pro výstupní hodnotu vazby	CPLOUTTR[FAx,LAx]=<hodnota>		
CPLOUTSC	Faktor měřítka pro výstupní hodnotu vazby	CPLOUTSC[FAx,LAx]=<hodnota>		
CPSYNCOP	Prahová hodnota pro "hrubé" synchronní najíždění na určitou pozici	CPSYNCOP[FAx]=<hodnota>		
CPSYNFIP	Prahová hodnota pro "jemné" synchronní najíždění na určitou pozici	CPSYNFIP[FAx]=<hodnota>		
CPSYNCOP2	Druhá mezní hodnota pro "hrubé" synchronní najíždění na určitou pozici	CPSYNCOP2[FAx]=<hodnota>		
CPSYNFIP2	Druhá mezní hodnota pro "jemné" synchronní najíždění na určitou pozici	CPSYNFIP2[FAx]=<hodnota>		

Klíčové slovo	Vlastnost vazby / význam	Syntaxe		
CPSYNCOV	Mezní hodnota pro "hrubé" synchronní najíždění na určitou rychlost	CPSYNCOV[FAx]=<hodnota>		
CPSYNFIV	Mezní hodnota pro "jemné" synchronní najíždění na určitou rychlost	CPSYNFIV[FAx]=<hodnota>		
CPMBRAKE	Chování vlečné osy při určitých signálech a příkazech pro zastavení	CPMBRAKE[FAx]=<bitově kódovaná hodnota>		
CPMVDI	Chování vlečné osy při určitých signálech rozhraní NC/PLC	CPMVDI[FAx]=<bitově kódovaná hodnota>		
CPMALARM	Potlačení výstupu speciálních alarmů vztahujících se k vazbě	CPMALARM[FAx]=<bitově kódovaná hodnota>		
CPSETTYPE	Typ vazby	CPSETTYPE[FAx]="<typ vazby>"		
		"<typ vazby>":	"CP"	Možnost volného programování
			"TRAIL"	Typ vazby "Vlečení"
			"LEAD"	Typ vazby "Vazba řídící hodnotou"
			"EG"	Typ vazby "Elektronická převodovka"
			"COUP"	Typ vazby "Synchronní vřeteno"

FAx: Vlečná osa/vlečné vřeteno

LAx: Řídící osa/řídící vřeteno

Poznámka

Vlastnostem vazby, které nejsou explicitně naprogramovány (ve výrobním programu nebo v synchronních akcích), jsou dosazena jejich standardní nastavení.

V závislosti na nastavení klíčového slova CPSETTYPE mohou být dosazeny místo standardních nastavení (CPSETTYPE="CP") také předem nastavené vlastnosti vazby.

Další informace

Pokud budete potřebovat podrobné informace týkající se generické vazby, viz:

- Příručka k funkcím, Osy a vřetena

4.16.7 Tangenciální řízení

4.16.7.1 Definice vazby (TANG)

Prostřednictvím předem definované procedury TANG(...) je definována tangenciální vazba mezi kruhovou osou, která je vlečnou osou, a dvěma geometrickými osami, které jsou řídícími osami. Vlečná osa se přitom kontinuálně nastavuje vzhledem k tečně dráhy řídících os.

Poznámka

Převodový faktor

Faktor vazby 1 nemusí být explicitně naprogramován.
Prostřednictvím faktoru vazby -1 se směr tangenciální osy obrátí.

Syntaxe

```
TANG(<vlečná osa>, <řídící osa_1>, <řídící osa_2>, <faktor vazby>,
<souřadný systém>, <optimalizace>)
```

Význam

`TANG(...):`	Definice tangenciální vazby		
`<vlečná osa>:`	Název vlečné osy (kruhová osa)		
	Datový typ:	AXIS	
	Rozsah hodnot:	Názvy kanálových os	
`<řídící osa_1>` `<řídící osa_2>:`	Názvy řídících os (geometrické osy) [1]		
	Datový typ:	AXIS	
	Rozsah hodnot:	Názvy geometrických os v kanálu	
`<faktor vazby>:`	Faktor n pro změnu úhlu vlečné osy při změně tečny dráhy řídících os: Změna hodnoty úhlu$_{\text{vlečná osa}}$ = Změna hodnoty úhlu$_{\text{tečna dráhy}}$ * n		
	Datový typ:	REAL	
	Předdefinovaná hodnota:	1,0	
`<Souřadný systém>:`	Souřadný systém, který je v platnosti [2]		
	Datový typ:	CHAR	
	Hodnota:	`"B"`:	Základní souřadný systém (předdefinovaná hodnota)
		`"W"`:	Souřadný systém obrobku (není k dispozici)

<optimalizace>:	Druh optimalizace		
	Datový typ:	CHAR	
	Hodnota:	"S":	Standardní (předdefinovaná hodnota) Dynamika kruhové osy nemá žádný zpětný vliv na řídící osy. Jestliže je dynamika kruhové osy vyšší, než je pro vlečení zapotřebí, je její pohyb dostatečně přesný. Pokud není dynamika kruhové osy dostatečně vysoká, aby stíhala náležitě reagovat na změnu tečny dráhy, odlišuje se nastavení kruhové osy podél nedefinovaných úseků zaoblení od požadovaného nasměrování.
		"P":	Dynamika kruhové osy je při plánování dráhy řídících os zohledňována. Za tím účelem musí být při aktivování tangenciální vazby pomocí příkazu TANGON() zadány dva doplňkové parametry: • Dráha zaoblení • Tolerance úhlu Viz kapitola "Aktivování vazby (TANGON) (Strana 957)". **Poznámka** Ve spojení s kinematickými transformacemi doporučujeme používat druh optimalizace "P".
Poznámka Předdefinované hodnoty nemusí být explicitně naprogramovány.			
[1] **Poznámka** Jako řídící osy tangenciální vazby se musí používat geometrické osy, které objíždějí naprogramovanou dráhu v souřadném systému stroje (MCS) (vztaženo na základní nastavení stroje). Jestliže je např. na frézce s naklápěcí hlavičkou použit cyklus naklápění CYCLE800, uskutečňuje se v závislosti na konfiguraci cyklu interpolace v systému WCS, např. s geometrickými osami X a Y. Tangenciální vazba ale musí být definována s geometrickými osami, jež jsou pak řídícími osami, které objíždějí naprogramovanou dráhu v systému MCS. Pro tento účel je jako řídící osy potřeba použít geometrické osy v **nenaklopeném** stavu stroje.			
[2] **Poznámka** Základní souřadný systém (BCS) nesmí být vůči MCS otočený. Jestliže je systém BCS otočený, např. pomocí příkazu ROT nebo prostřednictvím cyklu naklápění CYCLE800, nepracuje tangenciální řízení správně.			

4.16.7.2 Aktivování vytváření pomocných bloků (TLIFT)

Jestliže tangenciální změna vlečné osy na určitém místě naprogramované dráhy řídících os překročí mezní hodnotu nastavenou ve strojním parametru MD37400 $MA_EPS_TLIFT_TANG_STEP, bude další plánování dráhy záviset na nastaveném chování v rozích. Pokud není použita předem definovaná procedura TLIFT(...), bude pohyb probíhat v závislosti na chování na rohových zaobleních, které bylo naprogramováno v souvislosti s funkcemi TANG(...) (Strana 955) a TANGON(...) (Strana 957).

Aktivování vytváření pomocných bloků

Když je v návaznosti na funkci TANG(...) naprogramován příkaz TLIFT(...), pak když je během předběžného zpracování rozpoznán roh, na tomto místě dráhy řídící systém automaticky vloží vygenerovaný pomocný blok.

Při zpracování programu se pak po dosažení tohoto pomocného bloku řídící osy pozastaví. V pomocném bloku se vlečná osa se svou maximální dynamikou otočí ve směru tečny ke dráze v následujícím bloku. Potom se řídící osy dále pohybují po své naprogramované dráze.

Deaktivování vytváření pomocných bloků

Abyste vytváření pomocných bloků deaktivovali, musí být prostřednictvím příkazu TANG(...) tangenciální vazba nově definována, ale tentokrát bez navazujícího aktivování vkládání pomocných bloků pomocí příkazu TLIFT(...).

Syntaxe

```
TLIFT(<vlečná osa>)
```

Význam

`TLIFT(...)`:	Aktivování rozpoznávání rohů s výpočtem pomocných bloků	
`<vlečná osa>`:	Název vlečné osy (kruhová osa)	
	Datový typ:	AXIS
	Rozsah hodnot:	Názvy kanálových os

Rychlost otáčení vlečné osy

Dráhová osa

Pokud se vlečná osa již před aktivováním tangenciální vazby pohybovala jako osa dráhová, probíhá otáčivý pohyb v pomocném bloku v režimu dráhové osy.

Prostřednictvím zadání vztažného rádiusu pomocí příkazu `FGREF[<osa>]=0.001` bude otáčivý pohyb probíhat s maximální rychlostí osy nastavenou v parametru:

MD32000 $MA_MAX_AX_VELO[<vlečná osa>]

Polohovací osa

Pokud se vlečná osa před aktivováním tangenciální vazby ještě nepohybovala jako osa dráhová, probíhá otáčení v pomocném bloku v režimu polohovací osy.

Pro otáčivý pohyb se přitom použije rychlost polohovací osy nastavená v parametru:

MD32060 $MA_POS_AX_VELO[<vlečná osa>]

4.16.7.3 Aktivování vazby (TANGON)

Prostřednictvím předem definované procedury TANGON(...) se aktivuje tangenciální vazba, která byla předtím definována pomocí příkazu TANG(...) (Strana 955). Vlečná osa se potom při následujících pohybech řídících os kontinuálně nastavuje vzhledem k tečně jejich dráhy.

Úhel vlečné osy

Úhel, který vlečná osa zaujímá vzhledem k tečně k dráze, závisí na přenosovém poměru zadaném předem v příkazu TANG(...), na úhlu offsetu stanoveném ve strojním parametru MD37402 $MA_TANG_OFFSET a na přičítaném úhlu offsetu zadaném v příkazu TANGON(...).

Optimalizace "P"

Pokud byla při definici tangenciální vazby (TANG...)) zadána jako optimalizační parametr hodnota „P“, musí být při aktivování vazby zadány parametry "Dráha zaoblení" a případně také parametr "Tolerance úhlu" (nepovinné).

Jestliže je jako tolerance úhlu zadána hodnota 0, uplatňuje se pouze parametr "Dráha zaoblení".

Pokud je jako úhlová tolerance zadána hodnota větší než 0, odpovídá používaná dráha zaoblení minimu z dráhy zaoblení nastavené v parametrech a z dráhy zaoblení vyplývající ze zadané úhlové tolerance.

Jestliže dynamika vlečné osy není postačující, takže podmínky definované v parametrech nelze splnit, rychlost pohybu řídících os po dráze se odpovídajícím způsobem sníží.

Syntaxe

```
TANGON(<vlečná osa>, <offset úhlu>, <dráha zaoblení>, <tolerance úhlu>)
```

Význam

`TANGON(...)`:	Aktivování tangenciální vazby	
`<vlečná osa>`:	Název vlečné osy (kruhová osa)	
	Datový typ:	AXIS
	Rozsah hodnot:	Názvy kanálových os
`<offset úhlu>`:	Úhlový offset vlečné osy vzhledem k tečně dráhy Vztažným bodem je nula (počátek) kruhové osy.	
	Datový typ:	REAL
`<dráha zaoblení>`:	Maximální přípustná dráha zaoblení Pokud by byla dráha zaoblení v důsledku dynamických podmínek větší, rychlost pohybu řídících os po dráze se sníží.	
	Datový typ:	REAL
`<úhlová tolerance>`:	Maximální přípustná tolerance vztahující se na zadaný úhel mezi nulovou polohou vlečné osy a tečnou ke dráze	
	Datový typ:	REAL

4.16.7.4 Deaktivování vazby (TANGOF)

Prostřednictvím předem definované procedury TANGOF(...) se deaktivuje tangenciální vazba, která byla předtím definována pomocí příkazu TANG(...) (Strana 955) a aktivována pomocí příkazu TANGON(...) (Strana 957). Vlečná osa se potom přestane nastavovat vzhledem k tečně dráhy řídící osy. Vazba vlečné osy vzhledem k řídícím osám zůstává ale i po svém deaktivování definována, což např. blokuje následující funkce:

- Změna roviny
- Přepínání geometrických os
- Definice nové tangenciální vazby pro danou vlečnou osu

K úplnému zrušení vazby mezi vlečnou osou a řídícími osami dojde teprve až po vymazání vazby pomocí příkazu TANGDEL(...) (Strana 959).

Programování

```
TANGOF(<vlečná osa>)
```

Význam

`TANGOF(...)`:	Deaktivování tangenciální vazby	
`<vlečná osa>`:	Název vlečné osy (kruhová osa)	
	Datový typ:	AXIS
	Rozsah hodnot:	Názvy kanálových os

4.16.7.5 Vymazání vazby (TANGDEL)

Když je tangenciální vazba definovaná pomocí příkazu TANG(...) (Strana 955) deaktivována pomocí příkazu TANGOF(...) (Strana 959), nepřestává existovat. Existující tangenciální vazba potom blokuje např. následující funkce:

- Změna roviny
- Přepínání geometrických os
- Definice nové tangenciální vazby pro danou vlečnou osu

Pomocí předem definované procedury TANGDEL(...) se existující tangenciální vazba po svém deaktivování pomocí příkazu TANGOF(...) vymaže.

Syntaxe

```
TANGDEL(<vlečná osa>)
```

Význam

`TANGDEL(...)`:	Vymazání tangenciální vazby definované pomocí příkazu TANG()	
	Platnost:	Bloková

<vlečná osa>:	Název vlečné osy, jejíž tangenciální vazba má být vymazána	
	Datový typ:	AXIS
	Rozsah hodnot:	Názvy kanálových os

Příklady

Změna řídící osy

Předtím, než je možno pro danou vlečnou osu definovat novou tangenciální vazbu s jinou řídící osou, musí být napřed již existující tangenciální vazba vymazána.

```
Programový kód                  Komentář
N10 TANG(A, X, Y, 1)            ; definice tangenciální vazby pro vlečnou osu A: A
                                vzhledem k osám X a Y
N20 TANGON(A)                   ; aktivování tangenciální vazby pro vlečnou osu A
N30 X10 Y20
...
N80 TANGOF(A)                   ; deaktivování tangenciální vazby pro vlečnou osu A
N90 TANGDEL(A)                  ; vymazání tangenciální vazby pro vlečnou osu A
...
N120 TANG(A, X, Z)              ; definice nové tangenciální vazby pro vlečnou osu A
N130 TANGON(A)                  ; aktivování nové tangenciální vazby pro vlečnou osu A
...
```

Přepínání geometrických os

Předtím, než je možno pro již existující vazbu uskutečnit přepnutí geometrických os, musí být tato vazba napřed vymazána.

```
Programový kód                  Komentář
N10 GEOAX(2, Y1)                ; 2. geometrická osa = strojní osa Y1
N20 TANG(A, X, Y)               ; definice tangenciální vazby pro vlečnou osu A
N30 TANGON(A, 90)               ; aktivování tangenciální vazby pro vlečnou osu A
N40 G2 F8000 X0 Y0 I0 J50       ; pohybový blok
N50 TANGOF(A)                   ; deaktivování tangenciální vazby pro vlečnou osu A
N60 TANGDEL(A)                  ; vymazání tangenciální vazby pro vlečnou osu A
N70 GEOAX(2, Y2)                ; 2. geometrická osa = strojní osa Y2
N80 TANG(A, X, Y)               ; definice nové tangenciální vazby pro vlečnou osu A
N90 TANGON(A, 90)               ; aktivování nové tangenciální vazby pro vlečnou osu
                                A
...
```

4.16.8 Vazba Master/Slave (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS):

"Vazba Master/Slave" umožňuje následující:

- Aktivování vazby osy typu Slave s osou typu Master, když jsou obě tyto osy v klidu.
- Aktivování a deaktivování vazby **otáčejících se** vřeten v režimu regulace otáček.
- Dynamická konfigurace.

Poznámka

Režim polohování

U os a vřeten v režimu polohování se vazba aktivuje a rozpojuje, jen když jsou osy v klidu.

Syntaxe

```
MASLON(<Slave_1>,<Slave_2>,...)
MASLOF(<Slave_1>,<Slave_2>,...)
MASLOFS(<Slave_1>,<Slave_2>,...)
```

Dynamická konfigurace:

```
MASLDEF(<Slave_1>,<Slave_2>,...,<Master>)
MASLDEL(<Slave_1>,<Slave_2>,...)
```

Význam

MASLON:	Aktivování dočasné vazby typu Master/Slave	
	<Slave_x>,...:	Osa typu Slave 1 ... n
MASLOF:	Deaktivování aktivní vazby typu Master/Slave	
	<Slave_1>,...:	Osa typu Slave 1 ... n
MASLOFS:	Deaktivování vazby typu Master/Slave a automatické zabrždění vřeten typu Slave (viz upozornění "Chování vazby u vřeten v režimu regulace otáček"!)	
	<Slave_1>,...:	Osa typu Slave 1 ... n
MASLDEF:	Definice/změna spojení typu Master/Slave pomocí výrobního programu	
	<Slave_1>,...:	Osa typu Slave 1 ... n
	<Master>:	Osa typu Master
MASLDEL:	Zrušení vazby Master/Slave a vymazání definice tohoto spojení	
	<Slave_1>,...:	Osa typu Slave 1 ... n
	Upozornění: Definice spojení Master/Slave konfigurovaná ve strojních parametrech zůstává zachována.	

Poznámka

Chování vazby u vřeten v režimu regulace otáček

U vřeten v režimu regulace otáček je chování vazby v případě příkazů MASLON, MASLOF, MASLOFS a MASLDEL explicitně definováno pomocí následujícího strojního parametru:

MD37263 $MA_MS_SPIND_COUPLING_MODE

Při standardním nastavení MD37263 = 0 se uskutečňuje aktivování a rozpojování vazby s osami typu Slave jen tehdy, jsou-li všechny podílející se osy v klidu. Příkaz MASLOFS odpovídá příkazu MASLOF.

Jestliže je nastaveno MD37263 = 1 příkazy pro ovládání vazby se uskuteční okamžitě, tedy během pohybu. V případě příkazu MASLON se vazba okamžitě zapojí a v případě příkazů MASLOFS nebo MASLOF se ihned rozpojí. Když je v platnosti příkaz MASLOF, vřetena typu Slave, které se v tomto okamžiku otáčejí, si své otáčky zachovají, dokud nebudou naprogramovány nové. V případě příkazu MASLOFS jsou oproti tomu automaticky zabrzděna.

Poznámka

V případě příkazů MASLOF/MASLOFS odpadá implicitní zastavení předběžného zpracování. Protože se neuskutečňuje zastavení předběžného zpracování, neposkytují systémové proměnné $P... pro osy typu Slave až do okamžiku opětovného naprogramování žádné aktualizované hodnoty.

Poznámka

Pro osu Slave může být pomocí příkazu PRESETON skutečná hodnota synchronizována na stejnou hodnotu, jakou má osa Master. Za tím účelem musí být trvalá vazba Master/Slave na krátkou chvíli deaktivována, aby bylo možné ose Slave, která po zapnutí systému (Power-on) nenajela na referenční bod, dosadit hodnotu osy Master. Potom se trvalá vazba znovu obnoví.

Trvalá vazba typu Master/Slave se aktivuje pomocí následujícího nastavení strojního parametru:
MD37262 $MA_MS_COUPLING_ALWAYS_ACTIVE = 1
Toto nastavení nemá žádný vliv na příkazy NC jazyka pro dočasnou osu.

Příklady

Příklad 1: Dosazení skutečné hodnoty ose typu Slave ve vazbě Master-slave

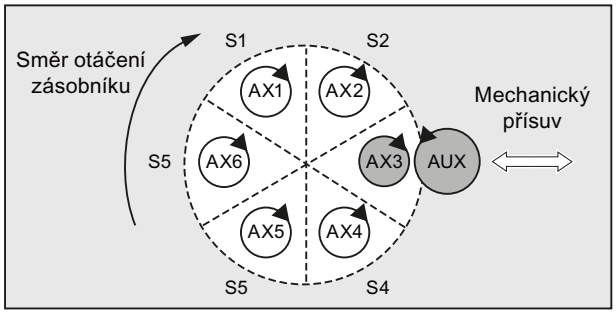
U permanentní vazby typu Master-slave je možno pomocí příkazu PRESETON nastavit skutečnou hodnotu osy typu slave na hodnotu osy typu master.

Programový kód	Komentář
$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=0	; deaktivování permanentní vazby osy typu Slave
NEWCONF	; aktivování změny strojního parametru
STOPRE	
MASLOF(Y1)	; deaktivování dočasného spojení
PRESETON(AX2,$VA_IM(M_AX))	; skutečná hodnota osy typu Slave = skutečná hodnota osy typu Master

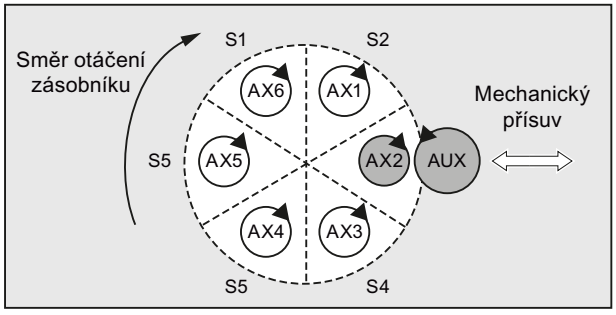
Programový kód	Komentář
`$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=1`	`; aktivování permanentní vazby osy typu Slave`
`NEWCONF`	`; aktivování změny strojního parametru`

Příklad 2: Dynamická konfigurace spojení os typu Master/Slave

Aby po otočení osového zásobníku mohla být vazba připojena na jiné vřeteno, musí být napřed stará vazba rozpojena, její konfigurace vymazána a zadána nová konfigurace vazby.



Obrázek 4-15 Před otáčením osového zásobníku.



Obrázek 4-16 Nové otočení osového zásobníku o jeden slot

Programový kód	Komentář
`MASLDEF(AUX,S3)`	`; AUX: Slave, S3: Master = AX3`
`MASLON(AUX)`	`; aktivování vazby`
`M3=3 S3=4000`	`; otáčení osy typu Master`
`MASLDEL(AUX)`	`; deaktivování a vymazání vazby`
`AXCTSWE(CT1)`	`; uvolnění otáčení osového zásobníku`
`MASLDEF(AUX,S3)`	`; AUX: Slave, S3: Master = AX2`

Další informace

- Příručka k funkcím, Technologie
- Příručka k funkcím, Základní funkce

4.17 Synchronní akce

4.17.1 Definice synchronní akce

Synchronní akce je definována v jednom bloku výrobního programu. V rámci tohoto bloku nesmějí být naprogramovány žádné další příkazy, které nejsou součástí synchronní akce.

Synchronní akce se skládají z následujících komponent:

Platnost, ident. č. (volitelný)	Podmínková část (volitelný)			Akční část v případě splněné podmínky			Akční část v případě nesplněné podmínky (volitelný)		
	Četnost	G-příkaz (volitelný)	Podmínka	Klíčové slovo	G-příkaz (volitelný)	Akce	Klíčové slovo	G-příkaz (volitelný)	Akce
--- [1] ID=<č.> IDS=<č.>	--- [1] WHENEVER FROM WHEN EVERY	G...	Logický Výraz	DO	G...	Akce 1 ... Akce n	ELSE	G...	Akce 1 ... Akce n

[1] není naprogramováno

Syntaxe

```
DO <Akce 1> ... <Akce n>
<Četnost> [<G-Funce>] <Podmínka> DO <Akce 1> ... <Akce n>
ID=<Č.> <Četnost> [<G-Funkce>] <Podmínka> DO <Akce 1> ... <Akce n>
IDS=<Č.> <Četnost> [<G-Funkce>] <Podmínka> DO <Akce 1> ... <Akce n>
IDS=<č.> <Četnost> [<G-funkce>] <Podmínka> DO <Akce 1...n> ELSE
<Akce 1...n>
```

Další informace

Příručka k funkcím, Synchronní akce

4.18 Pohyb tam a zpět

4.18.1 Asynchronní oscilační pohyby (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB)

Osa vykonávající oscilační pohyb se pohybuje mezi dvěma body obratu 1 a 2 se zadanou hodnotou posuvu, dokud se pohyb tam a zpět nevypne.

Ostatní osy mohou být v průběhu oscilačního pohybu libovolně interpolovány. Prostřednictvím pohybu po dráze nebo s polohovací osou je možné dosáhnout spojitého přísuvu. Přitom však neexistuje **žádná souvislost** mezi oscilačním pohybem a přísuvným pohybem.

Vlastnosti asynchronního oscilačního pohybu

- Asynchronní oscilační pohyb je vztažen na specifickou osu a je v platnosti i přes hranice bloků.
- Blokově synchronní zapínání oscilačního pohybu je zajišťováno pomocí výrobního programu.
- Společná interpolace většího počtu os a superpozice úseku pro oscilační pohyb nejsou možné.

Programování

Zapínání a ovlivňování asynchronního oscilačního pohybu u výrobního programu, které odpovídá zpracování NC programu, je možné prostřednictvím následujících příkazů.

Naprogramované hodnoty se synchronně s blokem v rámci zpracování hlavního programu zadávají do odpovídajících nastavovaných parametrů a zůstávají v platnosti, dokud nejsou změněny.

Syntaxe

```
OSP1[<osa>]=<hodnota> OSP2[<osa>]=<hodnota>
OST1[<osa>]=<hodnota> OST2[<osa>]=<hodnota>
FA[<osa>]=<hodnota>
OSCTRL[<osa>]=(<volba pro aktivování>,<volba pro deaktivování>)
OSNSC[<osa>]=<hodnota>
OSE[<osa>]=<hodnota>
OSB[<osa>]=<hodnota>
OS[<osa>]=1
OS[<osa>]=0
```

Význam

`<osa>`:	Posuv oscilační osy		
`OS`:	Zapínání a vypínání oscilačního pohybu		
	Hodnota:	1	**Aktivování** oscilačního pohybu
		0	**Deaktivování** oscilačního pohybu
`OSP1`:	Definice polohy bodu obratu 1		
`OSP2`:	Definice polohy bodu obratu 2 **Upozornění:** Jestliže je aktivní inkrementální zadávání rozměrů, vypočítá se poloha inkrementálně vůči odpovídajícímu bodu obratu naposled naprogramovanému v NC programu.		
`OST1`:	Definice doby pozastavení v bodě obratu 1 v [s].		
`OST2`:	Definice doby pozastavení v bodě obratu 2 v [s].		

	<hodnota>:	-2	Interpolace bude pokračovat bez čekání na přesné najetí
		-1	Čekání na hrubé přesné najetí
		0	Čekání na jemné přesné najetí
		>0	Čekání na jemné přesné najetí a následné setrvání po zadanou dobu pozastavení **Upozornění:** Jednotky pro dobu pozastavení jsou identické s dobou pozastavení naprogramovanou pomocí příkazu G4.
FA:	Definice rychlosti posuvu Jako rychlost posuvu platí definovaná rychlost posuvu polohovací osy. Jestliže rychlost posuvu není definována, platí hodnota uložená ve strojním parametru.		
OSCTRL:	Zadání možností pro aktivování a deaktivování Hodnoty voleb 0 - 3 určují chování v bodech obratu při vypínání. Mohou být zvoleny varianty 0 - 3. Zbývající nastavení mohou být podle potřeby kombinovány se zvolenou variantou. Větší počet voleb může být uveden postupně pomocí znamének plus (+).		
	<hodnota>:	0	Při vypnutí oscilačního pohybu zastavit v následujícím bodě obratu (předdefinované nastavení). **Upozornění:** Hodnoty 1 a 2 jsou možné jen prostřednictvím resetu.
		1	Při vypnutí oscilačního pohybu zastavit v bodě obratu 1.
		2	Při vypnutí oscilačního pohybu zastavit v bodě obratu 2.
		3	Při vypnutí oscilačního pohybu nenajíždět do žádného bodu obratu, jestliže nejsou naprogramovány žádné vyjiskřovací zdvihy.
		4	Po vyjiskření najet na koncovou pozici.
		8	Jestliže je oscilační pohyb přerušen prostřednictvím vymazání zbytkové dráhy, mají se potom zpracovat vyjiskřovací zdvihy a v případě potřeby se má najet na koncovou pozici.
		16	Jestliže je oscilační pohyb přerušen prostřednictvím vymazání zbytkové dráhy, má se po vypnutí najet do odpovídajícího bodu obratu.
		32	Změněná hodnota posuvu se uplatní teprve až od následujícího bodu obratu.
		64	FA se rovná 0, FA = 0: Superpozice dráhy je aktivní. FA se nerovná 0, FA <> 0: Superpozice rychlosti je aktivní.
		128	V případě kruhové osy DC (nejkratší dráha)
		256	Vyjiskřovací zdvih se uskuteční jako dvojitý zdvih (standardní) 1 = vyjiskřovací zdvih se uskuteční jako jeden zdvih.
		512	Najíždění napřed na počáteční pozici
OSNSC:	Stanovení počtu vyjiskřovacích zdvihů		

OSE:	Definice koncové pozice (ve WCS), na kterou se má po vypnutí oscilačního pohybu najet. **Poznámka:** Jestliže byl naprogramován příkaz „OSE“, pro příkaz „OSCTRL“ systém automaticky aktivuje volbu 4.
OSB:	Definice počáteční pozice (ve WCS), na kterou se má před zapnutím oscilačního pohybu najet. Na počáteční pozici se najíždí před bodem obratu 1. Jestliže se počáteční pozice kryje s polohou bodu obratu 1, jako následující se najíždí na bod obratu 2. Při dosažení počáteční pozice se neuplatňuje žádný doba prodlevy, a to i když se počáteční pozice kryje s bodem obratu 1. Místo toho se čeká na jemné přesné najetí. Nastavená podmínka přesného najetí je dodržena. **Upozornění:** Aby se najelo na počáteční pozici, musí být v nastavovaném parametru SD43770 $SA_OSCILL_CTRL_MASK nastaven bit 9.

Příklady

Příklad 1: Osa vykonávající oscilační pohyb se má pohybovat mezi dvěma body obratu

Osa Z se má pohybovat tam a zpět mezi polohami 10 a 100. Na bod obratu 1 se má najíždět s jemným přesným najetím, na bod obratu 2 se má najíždět s hrubým přesným najetím. Posuv pro oscilační osu má činit 250. Na konci opracování se mají uskutečnit 3 vyjiskřovací zdvihy a potom má oscilační osa odjet na koncovou pozici 200. Posuv pro přísuvnou osu má být 1, konec přísuvu ve směru osy X má být dosažen v poloze 15.

Programový kód	Komentář
WAITP(X,Y,Z)	; výchozí nastavení
G0 X100 Y100 Z100	; přepnutí do režimu polohování osy
WAITP(X,Z)	
OSP1[Z]=10 OSP2[Z]=100	; bod obratu 1, bod obratu 2
OSE[Z]=200	; koncová pozice
OST1[Z]=0 OST2[Z]=-1	; doba pozastavení v bodě U1: Jemné přesné najetí
	; doba pozastavení v bodě U2: Hrubé přesné najetí
FA[Z]=250 FA[X]=1	; posuv oscilační osy, posuv přísuvné osy
OSCTRL[Z]=(4,0)	; možnosti nastavení
OSNSC[Z]=3	; 3 vyjiskřovací zdvihy
OS[Z]=1	; spuštění oscilačního pohybu
WHEN $A_IN[3]==TRUE DO DELDTG(X)	; vymazání zbytkové dráhy
POS[X]=15	; počáteční poloha osy X
POS[X]=50	; koncová polohy osy X
OS[Z]=0	; zastavení oscilačního pohybu
M30	

Poznámka

Posloupnost příkazů „OSP1[Z]=...“ až „OSNCS[Z]=...“ může být naprogramována také v jednom bloku.

Příklad 2: Oscilační pohyb s on-line změnou bodu obratu

Nastavované parametry, které jsou zapotřebí pro asynchronní oscilační pohyby, mohou být nastavovány ve výrobním programu.

Jestliže se do nastavovaných parametrů zapisují hodnoty přímo ve výrobním programu, je změna v platnosti už v okamžiku předběžného zpracování. Synchronního chování lze dosáhnout pomocí zastavení předběžného zpracování (STOPRE).

Programový kód	**Komentář**
$SA_OSCILL_REVERSE_POS1[Z]=-10	
$SA_OSCILL_REVERSE_POS2[Z]=10	
G0 X0 Z0	
WAITP(Z)	
ID=1 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS1[Z] DO $AA_OVR[X]=0	; jestliže dojde k překročení skutečné hodnoty oscilační osy v bodě obratu, přísuvná osa se zastaví
ID=2 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[X]=0	
OS[Z]=1 FA[X]=1000 POS[X]=40	; zapnutí oscilačního pohybu
OS[Z]=0	; vypnutí oscilačního pohybu
M30	

Další informace

Osa vykonávající pohyb tam a zpět

Pro osu vykonávající pohyb tam a zpět platí:

- Pro oscilační pohyb je možné využít kteroukoli osu.
- Současně může být aktivních i více os vykonávajících pohyb tam a zpět (maximum: celkový počet polohovacích os).
- Pro oscilační osu je vždy aktivní lineární interpolace G1, a to nezávisle na G-příkazu, který momentálně platí v programu.

Oscilační osa může:

- být vstupní osou pro dynamickou transformaci
- být řídící osou pro osy gantry a vlečné osy
- se pohybovat:
  - bez omezení ryvu „BRISK“
    nebo
  - s omezením ryvu „SOFT“
    nebo
  - s nespojitou charakteristikou zrychlení (jako u polohovacích os)

Body obratu oscilačního pohybu

Při určování poloh pro pohyb tam a zpět je zapotřebí mít na paměti momentálně platná posunutí:

- Zadání absolutních hodnot
  „OSP1[Z]=<hodnota>"
  Poloha bodu obratu = součet posunutí + naprogramovaná hodnota
- Zadání relativních hodnot
  „OSP1[Z]=IC(<hodnota>)"
  Poloha bodu obratu = bod obratu 1 + naprogramovaná hodnota

Příklad:

Programový kód

```
N10 OSP1[Z] = 100 OSP2[Z] = 110
...
N40 OSP1[Z] = IC(3)
```

WAITP

Jestliže má oscilační pohyb vykonávat geometrická osa, musí být pro účely tohoto pohyb uvolněna příkazem "WAITP".

Po ukončení oscilačního pohybu se pomocí příkazu "WAITP" osa vykonávající oscilační pohyb znovu zaregistruje jako polohovací osa a je možné ji znovu normálně používat.

Oscilační pohyby s pohybovými synchronními akcemi a doby pozastavení

Po uplynutí nastavené doby pozastavení se v případě oscilačního pohybu uskuteční interní přechod na další blok (což lze pozorovat podle nové zbytkové dráhy os). Při přechodu na další blok se kontroluje vypínací funkce. Přitom je podle určeného nastavení řídícího systému pro průběh pohybu (OSCTRL) stanovena vypínací funkce. *Toto časové chování může být ovlivněno pomocí korekce (override) posuvu.*

Podle okolností se potom uskuteční ještě jeden oscilační zdvih, než se spustí vyjiskřovací zdvihy nebo než se najede do koncové pozice. *Přitom vzniká dojem, jakoby se změnilo chování při vypínání. Ale není tomu tak.*

4.18.2 Oscilační pohyby řízené prostřednictvím synchronních akcí (OSCILL)

U tohoto druhu oscilačních pohybů je přísuvný pohyb povolen jedině v bodech obratu, příp. v rámci definované oblasti okolo bodu obratu.

V závislosti na požadavcích mohou být oscilační pohyby v průběhu přísuvu

- uskutečňovány nebo
- pozastaveny, dokud není přísuv zcela dokončen.

Syntaxe

1. **Definice parametrů pro oscilační pohyby**
2. **Definice pohybových synchronních akcí**
3. **Přiřazení os, definice přísuvu**

Význam

`OSP1[<oscilační osa>]=`	Poloha bodu obratu 1
`OSP2[<oscilační osa>]=`	Poloha bodu obratu 2
`OST1[<oscilační osa>]=`	Doba prodlevy v bodě obratu 1 v sekundách
`OST2[<oscilační osa>]=`	Doba prodlevy v bodě obratu 2 v sekundách
`FA[<oscilační osa>]=`	Posuv oscilační osy
`OSCTRL[<oscilační osa>]=`	Možnosti aktivování/deaktivování
`OSNSC[<oscilační osa>]=`	Počet vyjiskřovacích zdvihů
`OSE[<oscilační osa>]=`	Koncová pozice
`WAITP(<oscilační osa>)`	Uvolnění osy pro oscilační pohyby

Přiřazení os, přísuv

```
OSCILL[<oscilační osa>]=(<přísuvná osa 1>,<přísuvná osa 2>,<přísuvná osa 3>)
POSP[<přísuvná osa>]=(<koncová pozice>,<dílčí délka>,<režim>)
```

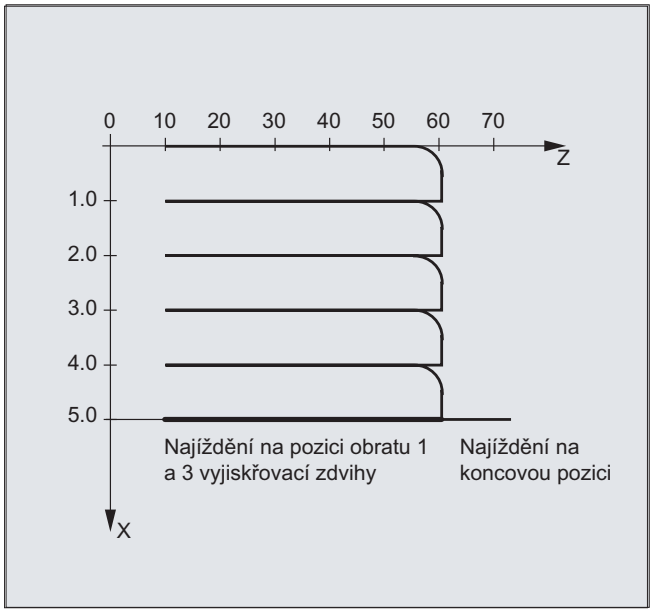
`OSCILL`:	Přiřazení přísuvných os k oscilační ose
`POSP`:	Definice celkového a dílčího přísuvu (viz kapitola správa souborů a programů)
`koncová pozice`:	Koncová pozice pro přísuvnou osu poté, co byly uskutečněny všechny dílčí přísuvy.
`dílčí délka`:	Velikost dílčího přísuvu v bodě obratu/oblasti obratu
`režim`:	Rozdělení celkového přísuvu do dílčích přísuvů = dva stejně velké zbytkové přísuvy (standardní nastavení), = všechny dílčí přísuvy jsou stejně veliké

Pohybové synchronní akce

`WHEN… … DO`	jestliže..., potom...
`WHENEVER … DO`	vždycky když..., potom...

Příklad:

V bodě obratu 1 se nemá provádět žádný přísuv. U bodu obratu 2 se má uskutečnit přísuv již ve vzdálenosti ii2 před bodem obratu 2 a oscilační osa v bodě obratu nemá čekat na ukončení dílčího přísuvu. Oscilační osou je osa Z a přísuvnou osou je osa X.



1. Parametry pro oscilační pohyby

Programový kód	Komentář
DEF INT ii2	; definice proměnné pro oblast obratu 2
OSP1[Z]=10 OSP2[Z]=60	; definice bodu obratu 1 a 2
OST1[Z]=0 OST2[Z]=0	; bod obratu 1: Jemné přesné najetí
	Bod obratu 2: Jemné přesné najetí
FA[Z]=150 FA[X]=0.5	; posuv oscilační osy Z, posuv přísuvné osy X
OSCTRL[Z]=(2+8+16,1)	; oscilační pohyb vypnout v bodě obratu 2, po vymazání zbytkové dráhy vyjiskření a najíždění do koncové polohy, po vymazání zbytkové dráhy najíždění do odpovídajícího bodu obratu
OSNC[Z]=3	; vyjiskřovací zdvihy
OSE[Z]=70	; koncová pozice = 70
ii2=2	; nastavení oblasti obratu
WAITP(Z)	; povolení oscilačního pohybu pro osu Z

2. Pohybové synchronní akce

Programový kód	Komentář
WHENEVER $AA_IM[Z]<$SA_OSCILL_REVERSE_POS2[Z] DO -> $AA_OVR[X]=0 $AC_MARKER[0]=0	; vždy když je aktuální pozice oscilační osy Z v MCS menší, než je začátek oblasti obratu 2, nastavit axiální korekci (override) přísuvné osy X na 0% a značku s indexem 0 na hodnotu 0

Programový kód	Komentář
`WHENEVER $AA_IM[Z]>=$SA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[Z]=0`	`; vždy když je aktuální pozice oscilační osy Z v MCS větší nebo rovny poloze bodu obratu 2, nastaví se axiální korekce (override) oscilační osy Z na 0%`
`WHENEVER $AA_DTEPW[X]==0 DO $AC_MARKER[0]=1`	`; vždy když je zbytková dráha dílčího přísuvu rovna 0, nastavit značku s indexem 0 na hodnotu 1`
`WHENEVER $AC_MARKER[0]==1 DO $AA_OVR[X]=0 $AA_OVR[Z]=100`	`; vždy když je značka s indexem 0 rovna 1, nastavit axiální korekci (override) přísuvné osy X na 0% Tím se zabrání příliš brzkému přísuvu (oscilační osa Z ještě oblast obratu 2 znovu neopustila, přísuvná osa X je ale připravena pro nový přísuv). Nastavení axiální korekce (override) oscilační osy Z z 0% (akce 2. synchronní akce) znovu na 100% pro další posuv.`

-> musí být naprogramováno v jednom bloku

3. Spuštění oscilačního pohybu

Programový kód	Komentář

`OSCILL[Z]=(X) POSP[X]=(5,1,1)`	`; spuštění os`
	`Oscilační osa Z je přiřazena ose X jako přísuvné ose.`
	`Osa X má najíždět až do koncové pozice 5 v krocích po 1.`

`M30`	`; konec programu`

Další informace

1. **Definice parametrů oscilačního pohybu**
   Před pohybovým blokem, který obsahuje přiřazení přísuvné a oscilační osy, jakož i stanovení přísuvu, je zapotřebí definovat parametry pro oscilační pohyby (viz "Asynchronní oscilační pohyby").
2. **Definice pohybových synchronních akcí**
   Prostřednictvím podmínek synchronních akcí se uskutečňuje:
   Potlačení přísuvu, dokud se oscilační osa nachází uvnitř oblasti obratu (ii1, ii2) nebo v bodě obratu (U1, U2).
   Pozastavení oscilačního pohybu v průběhu přísuvu v bodě obratu.
   Spuštění oscilačního pohybu po skončení dílčího přísuvu.
   Stanovení **spuštění následujícího dílčího přísuvu**.
3. **Přiřazení oscilační a přísuvné osy** a definice **celkového a dílčího přísuvu**.

Definice parametrů oscilačního pohybu

Přiřazení oscilační a přísuvné osy: OSCILL

```
OSCILL[<oscilační osa>]=(<přísuvná osa 1>,<přísuvná osa 2>,<přísuvná
osa 3>)
```

Pomocí příkazu „OSCILL“ se přiřazují osy a spouští se oscilační pohyb.

Maximálně mohou být oscilační ose přiřazeny 3 přísuvné osy.

Poznámka

Před spuštěním oscilačního pohybu musí být definovány synchronizačním podmínky pro chování os.

Definice přiřazení: POSP

```
POSP[<přísuvná osa>]=(<koncová pozice>,<dílčí délka>,<režim>)
```

Pomocí příkazu „POSP“ je řídícímu systému sdělováno:

- Celkový přísuv (přes koncovou pozici)
- Velikost jednotlivých dílčích přísuvů v bodě obratu, příp. v oblasti obratu.
- Chování dílčího přísuvu při dosažení koncové pozice (pomocí parametru "režim").

Režim = 0	Posledním dvěma dílčím přísuvům je přiřazena taková délka, aby zbývající dráha do cílového bodu byla rozdělena na 2 stejně velké úseky (předdefinované nastavení).
Režim = 1	Všechny dílčí přísuvy jsou stejně veliké. Vypočítají se z celkového přísuvu.

Definice pohybových synchronních akcí

Pohybové synchronní akce popisované v následujících odstavcích se používají zcela obecně pro oscilační pohyby.

Představují vzorové postupy pro řešení jednotlivých požadavků, takže Vám mohou posloužit jako stavební kameny pro sestavování specifických uživatelských oscilačních pohybů.

Poznámka

V jednotlivých případech mohou být synchronizační podmínky naprogramovány i jinak.

Klíčová slova

`WHEN … DO …`	jestliže..., potom...
`WHENEVER … DO`	vždycky když..., potom...

Funkce

Pomocí následujících podrobně popisovaných prostředků NC jazyka můžete realizovat následující:

1. Přísuv v bodě obratu.
2. Přísuv v oblasti obratu.
3. Přísuv v obou bodech obratu.
4. Pozastavení oscilačního pohybu v bodě obratu.
5. Opětovné spuštění oscilačního pohybu.
6. Dílčí přísuv nespouštět příliš brzy.

Pro všechny synchronní akce uváděné v těchto příkladech platí následující předpoklady:

- Bod obratu 1 < bod obratu 2
- Z = oscilační osa
- X = přísuvná osa

Poznámka

Pokud budete potřebovat podrobnější vysvětlení, viz kapitola "Pohybové synchronní akce".

Přiřazení oscilační a přísuvné osy a definice celkového a dílčího přísuvu

Přísuv v oblasti obratu

Přísuvný pohyb by měl začínat v rámci oblasti obratu, předtím než je dosaženo bodu obratu.

Tyto synchronní akce blokují přísuvný pohyb, dokud se oscilační osa nachází v oblasti obratu.

Za uvedených předpokladů (viz výše) z toho vyplývají následující příkazy:

Oblast obratu 1:

```
WHENEVER $AA_IM[Z]>$SA_OSCILL_RESERVE_POS1[Z]+ii1 DO $AA_OVR[X]=0
```

Vždy když je aktuální pozice oscilační osy v MCS větší, než je začátek oblasti obratu 1, nastaví se axiální korekce (override) přísuvné osy na 0%.

Oblast obratu 2:

```
WHENEVER $AA_IM[Z]<$SA_OSCILL_RESERVE_POS2[Z]+ii2 DO $AA_OVR[X]=0
```

Vždy když je aktuální pozice oscilační osy v MCS menší, než je začátek oblasti obratu 2, nastaví se axiální korekce (override) přísuvné osy na 0%.

Přísuv v bodě obratu

Dokud oscilační osa nedosáhla bodu obratu, neuskutečňuje se žádný pohyb v přísuvné ose.

Za uvedených předpokladů (viz výše) z toho vyplývají následující příkazy:

Oblast obratu 1:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Vždy když je aktuální pozice oscilační osy Z v MCS větší nebo menší, než je poloha bodu obratu 1, potom se nastaví axiální korekce (override) přísuvné osy X na 0% a axiální korekce (override) oscilační osy Z na hodnotu 100%.

Oblast obratu 2:

Pro bod obratu 2:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Vždy když je aktuální pozice oscilační osy Z v MCS větší nebo menší, než je poloha bodu obratu 2, potom se nastaví axiální korekce (override) přísuvné osy X na 0% a axiální korekce (override) oscilační osy Z na hodnotu 100%.

Pozastavení oscilačního pohybu v bodě obratu

Oscilační osa se v bodě obratu pozastaví a současně se spustí přísuvný pohyb. Oscilační pohyb bude pokračovat, až když je přísuvný pohyb zcela dokončen.

Současně se může tyto synchronní akce používat k tomu, aby se přísuvný pohyb spouštěl, jestliže byl zastaven předcházející synchronní akcí, která je stále ještě v platnosti.

Za uvedených předpokladů (viz výše) z toho vyplývají následující příkazy:

Oblast obratu 1:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Vždy když je aktuální pozice oscilační osy v MCS rovna poloze bodu obratu 1, nastaví se axiální korekce (override) oscilační osy na 0% a axiální korekce (override) přísuvné osy na 100%.

Oblast obratu 2:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Vždy když je aktuální pozice oscilační osy Z v MCS rovna poloze bodu obratu 2, nastaví se axiální korekce (override) oscilační osy X na 0% a axiální korekce (override) přísuvné osy na 100%.

On-line vyhodnocování bodu obratu

Jestliže se na pravé straně porovnávaného výrazu nachází proměnná hlavního zpracování označená znaky `$$`, jsou obě proměnné průběžně vyhodnocovány během taktu IPO a vzájemně porovnávány.

Poznámka

Pokud budete potřebovat další související informace, viz kapitola "Pohybové synchronní akce".

Opětovné spuštění oscilačního pohybu

Tyto synchronní akce se použije tehdy, když potřebujete, aby pohyb oscilační osy pokračoval, jakmile je dílčí přísuvný pohyb dokončen.

Za uvedených předpokladů (viz výše) z toho vyplývají následující příkazy:

```
WHENEVER $AA_DTEPW[X]==0 DO $AA_OVR[Z]=100
```

Vždy když je zbytková dráha pro dílčí pohyb přísuvné osy X v MCS rovna nule, potom se nastaví axiální korekce (override) oscilační osy na 100%.

Následující dílčí přísuv

Po dokončení přísuvu je nutno zajistit, aby se následující dílčí přísuv nespustil příliš brzy.

Pro tento účel se používá kanálová značka (`$AC_MARKER[Index]`), která se na konci dílčího přísuvu (dílčí zbytková dráha ≡ 0) nastavuje a která se při opuštění oblasti obratu opět vymaže. Potom se pomocí synchronní akce zablokuje následující přísuvný pohyb.

Za uvedených předpokladů (viz výše) z toho vyplývají např. pro bod obratu 1 následující příkazy:

1. Nastavení značky:

```
WHENEVER $AA_DTEPW[X]==0 DO $AC_MARKER[1]=1
```

Vždy když je zbytková dráha pro dílčí pohyb přísuvné osy X v MCS rovna nule, potom se značka s indexem 1 nastaví na 1.

2. Vymazání značky

```
WHENEVER $AA_IM[Z]<> $SA_OSCILL_RESERVE_POS1[Z] DO $AC_MARKER[1]=0
```

Vždy když je aktuální pozice oscilační osy Z v MCS větší nebo menší, než je poloha bodu obratu 1, nastaví se značka 1 na 0.

3. Blokování přísuvu

```
WHENEVER $AC_MARKER[1]==1 DO $AA_OVR[X]=0
```

Vždy když je značka 1 rovna, nastaví se axiální korekce (override) přísuvné osy X na 0%.

4.19 Lisování a prostřihování

4.19.1 Aktivování/deaktivování

4.19.1.1 Aktivování/deaktivování lisování a prostřihování (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC)

Aktivování/deaktivování lisování, příp. prostřihování

Lisování, příp. prostřihování se aktivují pomocí příkazů `PON` a `SON`. Příkazem `SPOF` se všechny specifické funkce pro lisování a prostřihování ukončí. Příkazy `PON` a `SON` s modální platností se vzájemně vylučují, tzn. příkaz `PON` deaktivuje příkaz `SON` a naopak.

Lisování/prostřihování s hlavičkou

Funkce pro lisování, příp. prostřihování se spouštějí také pomocí příkazů `SONS` a `PONS`.

Oproti příkazům SON/PON, u kterých se řízení zdvihu uskutečňuje na úrovni interpolace, probíhá u těchto funkcí řízení spouštění zdvihu na úrovni technických signálů servomechanismů. Díky tomu lze pracovat s vyšší frekvencí zdvihů a tedy i s vyšším výkonem při lisování.

V průběhu vyhodnocování signálů v hlavičce jsou blokovány všechny funkce, které mají za následek změnu polohy lisovacích nebo prostřihovacích os (např. manuální posuvy ručním kolečkem, změny framu pomocí PLC, měřicí funkce).

Lisování se zpožděním

Funkce PDELAYON způsobuje zpožděné spuštění lisovacího zdvihu. Tento příkaz s modální platností má přípravnou funkci a zpravidla se nachází před příkazem PON. Po příkazu PDELAYOF probíhá normální další lisování.

Poznámka

Doba prodlevy se nastavuje pomocí nastavovaného parametru SD42400 $SC_PUNCH_DWELLTIME.

Zrychlení závisející na dráze

Pomocí příkazu PUNCHACC může být definována charakteristika zrychlení, která definuje různá zrychlení v závislosti na vzdálenosti díry.

Druhé rozhraní pro lisování

Stroje, které mají střídavě využívat druhé stanoviště s rozhraním (druhá lisovací jednotka nebo srovnatelné médium) mohou být přepínány na druhou dvojici rychlých digitálních vstupů a výstupů řídícího systému (V/V-pár). Obě stanoviště s rozhraním mohou využívat plné spektrum funkcí pro lisování/prostřihování. Pro přepínání mezi prvním a druhým stanovištěm s rozhraním slouží příkazy SPIF1 a SPIF2.

Poznámka

Předpoklad: Prostřednictvím strojních parametrů musí být druhý pár V/V definován pro lisovací funkce (--> viz informace od výrobce stroje!).

Syntaxe

```
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
```

Význam

PON:	Aktivování lisování	
SON:	Aktivování prostřihování	
PONS:	Aktivování lisování s hlavičkou	
SONS:	Aktivování prostřihování s hlavičkou	
SPOF:	Deaktivování lisování/prostřihování	
PDELAYON:	Aktivování lisování se zpožděním	
PDELAYOF:	Deaktivování lisování se zpožděním	
PUNCHACC:	Aktivování zrychlení závisejícího na dráze Parametr:	
	<Smin>	Nejmenší vzdálenost od díry
	<Amin>	Počáteční zrychlení Parametr <Amin> může být větší než <Amax>.
	<Smax>	Největší vzdálenost od díry
	<Amax>	Koncové zrychlení Parametr <Amax> může být větší než <Amin>.
SPIF1:	Aktivování **prvního** rozhraní pro lisování Řízení zdvihu se uskutečňuje prostřednictvím první dvojice rychlých V/V.	
SPIF2:	Aktivování **druhého** rozhraní pro lisování Řízení zdvihu se uskutečňuje prostřednictvím druhé dvojice rychlých V/V.	
	Upozornění: Po resetu nebo po náběhu řídícího systému je aktivní vždy první rozhraní. Jestliže se pro lisování používá jen jedno rozhraní, nemusí být naprogramováno.	

Příklady

Příklad 1: Aktivování prostřihování

Programový kód	Komentář
...	
N70 X50 SPOF	; najíždění na polohu bez spuštění lisování
N80 X100 SON	; aktivování prostřihování, spouštění zdvihu před pohybem (X=50) a na konci naprogramovaného pohybu (X=100)
...	

Příklad 2: Lisování se zpožděním

Programový kód	Komentář
...	
N170 PDELAYON X100 SPOF	; najíždění na polohu bez spuštění lisování, aktivování zpožděného spuštění lisování
N180 X800 PON	; aktivování lisování Po dosažení koncové pozice se uskuteční zpožděný lisovací zdvih.
N190 PDELAYOF X700	; deaktivování lisování se zpožděním, normální spouštění lisování na konci naprogramovaného pohybu

Programový kód	Komentář
...	

Příklad 3: Lisování se dvěma stanovišti s rozhraním

Programový kód	Komentář
...	
N170 SPIF1 X100 PON	; na konci bloku se uskutečňuje spuštění zdvihu na prvním rychlém výstupu Na prvním rychlém vstupu je monitorován signál "Zdvih aktivní".
N180 X800 SPIF2	; druhý zdvih se spouští na druhém rychlém výstupu Na druhém rychlém vstupu je monitorován signál "Zdvih aktivní".
N190 SPIF1 X700	; řízení zdvihu se pro všechny další zdvihy uskutečňuje na prvním rozhraní
...	

Další informace

Lisování a prostřihování s hlavičkou (PONS/SONS)

Lisování a prostřihování s hlavičkou není možné provádět ve více kanálech současně. Příkazy PONS, příp. SONS mohou být aktivovány vždy jen v jednom kanálu.

Aktivování zrychlení závisejícího na dráze (PUNCHACC)

Příklad:
PUNCHACC(2,50,10,100)

Vzdálenost od díry menší než 2 mm:

Pohyb bude probíhat se zrychlením 50% maximálního zrychlení.

Vzdálenost od díry 2 mm až 10 mm:

Zrychlení se bude přímo úměrně vzdálenosti zvyšovat až na 100%.

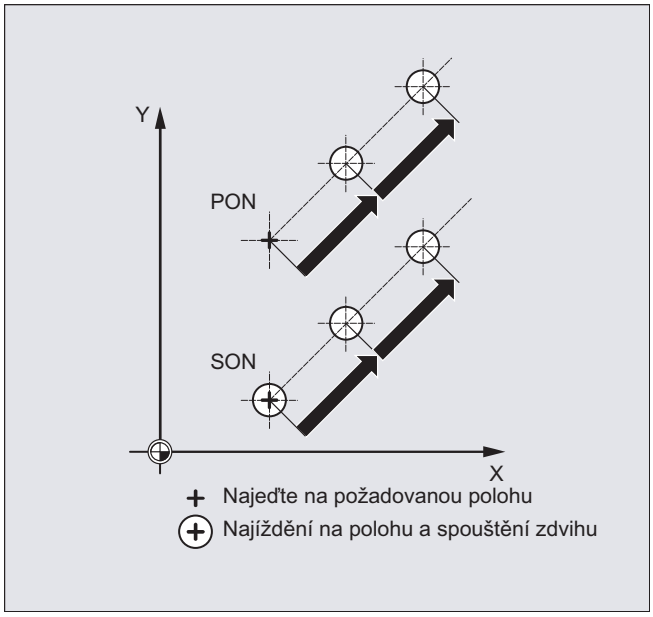
Vzdálenost od díry větší než 10 mm:

Pohyb se rychlením 100%.

Spouštění prvního zdvihu

Spouštění prvního zdvihu po aktivování funkce se uskutečňuje při prostřihování časově odlišně než při lisování.

- PON/PONS:
  - Všechny zdvihy - i zdvih prvního bloku po aktivování - probíhají na konci bloku.
- SON/SONS:
  - První zdvih po aktivování prostřihování se uskuteční už na začátku bloku.
  - Všechny ostatní zdvihy jsou však spouštěny až na konci bloku.



Lisování a prostřihování na místě

Ke spuštění zdvihu dojde jen tehdy, pokud blok obsahuje informace o pohybu lisovacích nebo prostřihovacích os (osy aktivní roviny).

Aby se přesto spustil zdvih na stejném místě, naprogramují se lisovací/prostřihovací osy s délkou dráhy 0.

Práce s otočnými nástroji

Poznámka

Aby byly otočné nástroje nastavovány tangenciálně vůči naprogramované dráze, používejte tangenciální řízení.

Použití příkazů M-funkcí

Pomocí techniky maker je stejně jako dříve možné používat speciální M-funkce namísto příkazů NC jazyka (kompatibilita). Vzhledem ke starším systémům přitom platí následující konvence:

```
M20, M23   ≙  SPOF
M22        ≙  SON
M25        ≙  PON
M26        ≙  PDELAYON
```

Příklad souboru makra:

Programový kód	Komentář
DEFINE M25 AS PON	; zapnutí lisování
DEFINE M125 AS PONS	; aktivování lisování s hlavičkou
DEFINE M22 AS SON	; aktivování prostřihování

Programový kód	Komentář
DEFINE M122 AS SONS	; aktivování prostřihování s hlavičkou
DEFINE M26 AS PDELAYON	; aktivování lisování se zpožděním
DEFINE M20 AS SPOF	; lisování, prostřihování deaktivovat
DEFINE M23 AS SPOF	; lisování, prostřihování deaktivovat

Příklad programování:

Programový kód	Komentář
...	
N100 X100 M20	; najíždění na polohu bez spuštění lisování
N110 X120 M22	; aktivování prostřihování, spouštění zdvihu před a po pohybu
N120 X150 Y150 M25	; aktivování lisování, spouštění zdvihu na konci pohybu
...	

4.19.2 Automatické rozdělení dráhy

Rozčlenění na dílčí úseky

Když je aktivní lisování, příp. prostřihování, zajišťují příkazy SPP a SPN rozčlenění celkové dráhy posuvu, která byla naprogramována pro dráhové osy, na daný počet stejně dlouhých dílčích úseků (ekvidistantní rozdělení dráhy). Interně odpovídá každému dílčímu úseku jeden blok.

Počet zdvihů

Při lisování se první zdvih uskuteční na konci prvního dílčího úseku, při prostřihování oproti tomu v počátečním bodě prvního dílčího úseku. V důsledku toho na celkové dráze platí následující počty:

Lisování: Počet zdvihů = počet dílčích úseků

Prostřihování: Počet zdvihů = počet dílčích úseků + 1

Pomocné funkce

Pomocné funkce jsou zpracovávány v prvním vygenerovaném úseku.

Syntaxe

SPP=

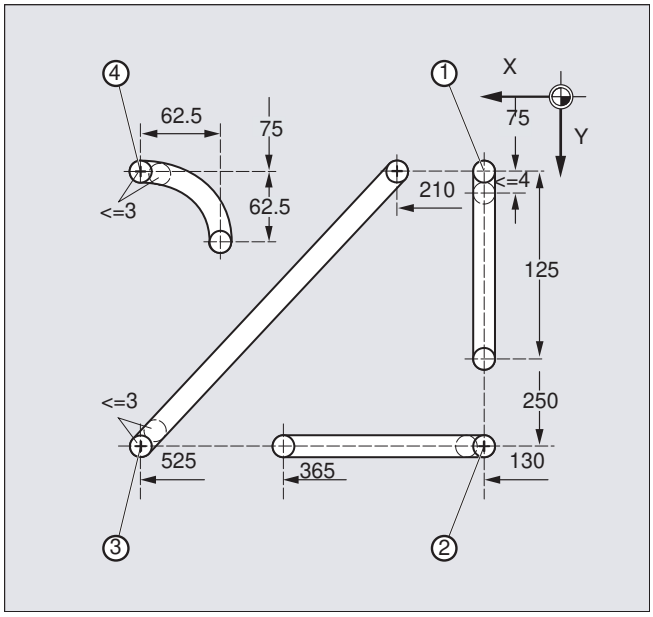
SPN=

Význam

SPP:	Velikost dílčího úseku (maximální vzdálenost zdvihů), modální platnost
SPN:	Počet úseků na blok, bloková platnost

Příklad 1

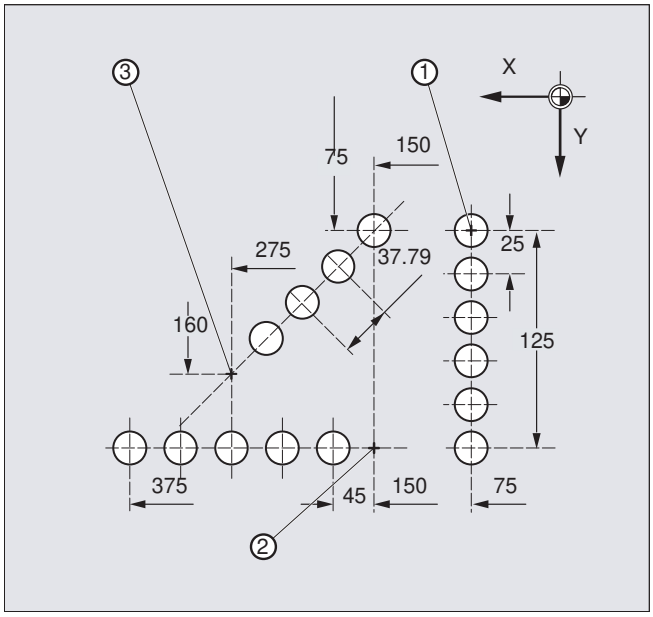
Naprogramované úseky pro prostřihování mají být automaticky rozděleny na stejně veliké úseky.



Programový kód	**Komentář**
N100 G90 X130 Y75 F60 SPOF	; najíždění na polohu počátečního bodu 1
N110 G91 Y125 SPP=4 SON	; aktivování prostřihování, maximální délka dílčího úseku pro automatické rozdělení dráhy: 4 mm
N120 G90 Y250 SPOF	; vypnutí prostřihování, najíždění na pozici počátečního bodu 2
N130 X365 SON	; aktivování prostřihování, maximální délka dílčího úseku pro automatické rozdělení dráhy: 4 mm
N140 X525 SPOF	; vypnutí prostřihování, najíždění na pozici počátečního bodu 3
N150 X210 Y75 SPP=3 SON	; aktivování prostřihování, maximální délka dílčího úseku pro automatické rozdělení dráhy: 3 mm
N160 X525 SPOF	; vypnutí prostřihování, najíždění na pozici počátečního bodu 4
N170 G02 X-62.5 Y62.5 I J62.5 SPP=3 SON	; aktivování prostřihování, maximální délka dílčího úseku pro automatické rozdělení dráhy: 3 mm
N180 G00 G90 Y300 SPOF	; vypnutí prostřihování

Příklad 2

Pro jednotlivé řady děr má být provedeno automatické rozdělení dráhy na úseky. Pro rozdělení se vždy zadává maximální délka úseku dráhy (hodnota SPP).



Programový kód	**Komentář**
N100 G90 X75 Y75 F60 PON	; najíždění na počáteční bod 1, lisování jednotlivé díry
N110 G91 Y125 SPP=25	; maximální délka dílčího úseku pro automatické rozdělení dráhy: 25 mm
N120 G90 X150 SPOF	; vypnutí lisování, najíždění na pozici počátečního bodu 2
N130 X375 SPP=45 PON	; aktivování lisování, maximální délka dílčího úseku pro automatické rozdělení dráhy: 45 mm
N140 X275 Y160 SPOF	; vypnutí lisování, najíždění na pozici počátečního bodu 3
N150 X150 Y75 SPP=40 PON	; aktivování lisování, namísto naprogramované délky dílčího úseku 40 mm se ;použije vypočítaná délka díl- čího úseku 37,79 mm
N160 G00 Y300 SPOF	; vypnutí lisování, najíždění na danou pozici

4.19.2.1 Rozdělení cesty u dráhových os

Délka dílčího úseku SPP

Pomocí příkazu `SPP` zadáváte maximální vzdálenost zdvihů a v důsledku toho maximální délku dílčího úseku, na které má být celková dráha rozdělena. Pro deaktivování této funkce slouží příkazy `SPOF` nebo `SPP=0`.

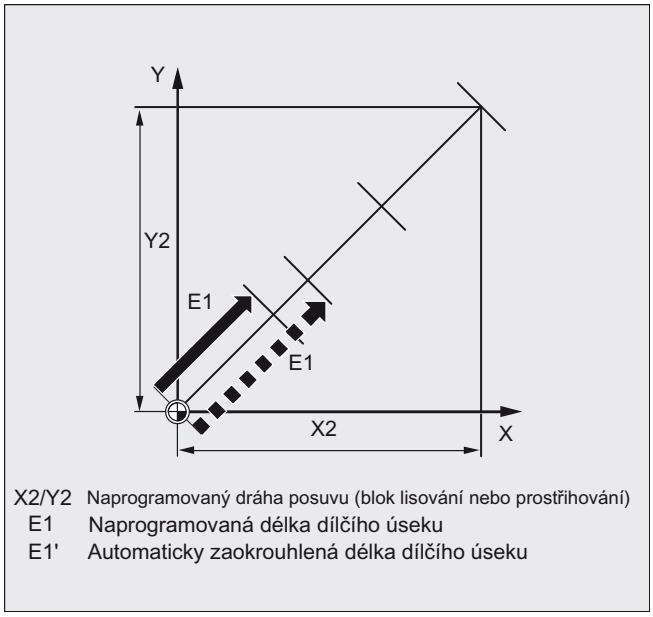
Příklad:

```
N10 SON X0 Y0
N20 SPP=2 X10
```

Celková dráha 10 mm se rozdělí na 5 dílčích úseků každý o délce 2 mm (SPP=2).

Poznámka

Rozdělení dráhy pomocí příkazu SPP je vždy ekvidistantní: všechny dílčí úseky jsou stejně dlouhé. To znamená, že naprogramovaná velikost dílčího úseku (hodnota SPP) platí jen tehdy, pokud je podíl celkové délky dráhy a hodnoty SPP celé číslo. Jestliže tomu tak není, interně se velikost dílčího úseku zmenší natolik, aby byl podíl celé číslo.



Příklad:

```
N10 G1 G91 SON X10 Y10
N20 SPP=3.5 X15 Y15
```

V případě délky celé dráhy 15 mm a délky dílčích úseků 3,5 mm vyplývá hodnota podílu, která není celočíselná (4,28). V důsledku toho systém hodnotu SPP zmenší, aby vyšel nejbližší možný celočíselný podíl. V tomto případě vyjde délka dílčího úseku 3 mm.

Počet dílčích úseků SPN

Pomocí příkazu SPN definujete počet dílčích úseků, které mají být z celkové délky dráhy vytvořeny. Délka dílčího úseku se vypočítá automaticky. Protože příkaz SPN má blokovou platnost, musí být předtím pomocí příkazů PON nebo SON aktivováno lisování nebo prostřihování.

4.19.2.2 Rozdělení cesty u jednotlivých os

Jestliže jsou vedle dráhových os také samostatné osy definovány jako osy pro lisování/ prostřihování, je možné i na tyto osy aplikovat automatické rozdělení dráhy.

Chování samostatné osy v případě SPP

Naprogramovaná délka dílčího úseku (SPP) se v zásadě vztahuje na dráhové osy. Proto je v bloku, ve kterém kromě pohybu samostatnou osou a hodnoty příkazu SPP není naprogramována žádná dráhová osa, je hodnota SPP ignorována.

Jestliže jsou v bloku naprogramovány jak samostatné osy, tak i dráhové osy, řídí se chování samostatné osy podle nastavení odpovídajícího strojního parametru.

1. Standardní nastavení
   Dráha samostatné osy je prostřednictvím pomocných bloků vygenerovaných příkazem SPP rozdělena na stejné úseky.

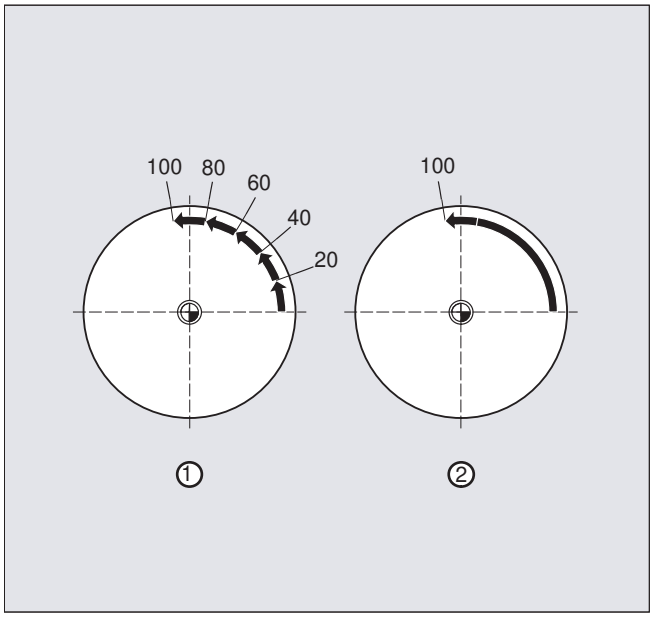
Příklad:

```
N10 G1 SON X10 A0

N20 SPP=3 X25 A100
```

Na základě zdvihové vzdálenosti 3 mm se v případě celkové délky posuvu osy X (dráhová osa) 15 mm vytvoří 5 bloků.

Osa A se tedy v každém bloku otočí o 20°.



1. Samostatná osa bez rozdělení dráhy
   Samostatná osa urazí svou celkovou dráhu v rámci prvního z vytvořených bloků.
2. Odlišné rozdělení dráhy
   Chování samostatné osy je závislé na interpolaci dráhových os.

- Kruhová interpolace: Rozdělení dráhy
- Lineární interpolace: Žádné rozdělení dráhy

Chování v případě SPN

Naprogramovaný počet dílčích úseků platí také tehdy, pokud dráhová osa není naprogramována současně.

Předpoklad: Samostatná osa je definována jako osa pro prostřihování/lisování.

4.20 Broušení

4.20.1 Aktivování/deaktivování speciálního monitorování pro brusné nástroje (TMON, TMOF)

Prostřednictvím předem definovaných procedur TMON(...) a TMOF(...) je možno aktivovat, příp. deaktivovat specifické monitorování pro brusné nástroje (monitorování geometrie a otáček).

Předpoklady

Musí být nastaveny parametry specifického nástroje $TC_TPG1 až $TC_TPG9.

Syntaxe

```
TMON(<TNr>)
...
TMOF(<TNr>)
```

Význam

`TMON(...)`:	Aktivování specifického monitorování pro brusné nástroje Příkaz musí být naprogramován v kanálu, v němž má být specifické monitorování pro brusní nástroje aktivováno.
`TMOF(...)`:	Deaktivování specifického monitorování pro brusné nástroje Příkaz musí být naprogramován v kanálu, v němž má být specifické monitorování pro brusní nástroje deaktivováno.
`<TNr>`:	T-číslo **Poznámka:** Je zapotřebí jen tehdy, pokud má být monitorování aktivováno, příp. deaktivováno pro neaktivní brusný kotouč a nikoli na aktivní nástroj, který je momentálně používán.
`TMOF(0)`:	Deaktivování monitorování pro všechny nástroje

4.21 Rozšířené zastavování a odjíždění (ESR)

Funkce "Rozšířené zastavování a odjíždění", v dalším textu označovaná zkratkou ESR, nabízí možnost flexibilní reakce v chybových situacích v závislosti na procesu.

- **Rozšířené zastavování**
  Pokud to specifická chybová situace umožňuje, jsou všechny osy, které byly pro rozšířené zastavování uvolněny, řízeným způsobem zastaveny.
- **Odjíždění**
  Nástroj, který se momentálně nachází v záběru, je co možno nejrychleji stažen pryč od obrobku.
- **Generátorický režim (funkce pohonu SINAMICS "Regulace stejnosměrného napětí")**
  Jestliže napětí meziobvodu poklesne pod hodnotu nastavitelnou v parametrech, např. při výpadku síťového napětí, elektrická energie potřebná pro odjížděcí pohyb se vytvoří prostřednictvím napájení z druhé strany z brzdicí energie pohonu, který toto umožňuje (generátorický režim).

Zdroje pro spuštění

Obecné zdroje (externí mimo NC systém/globální nebo BAG/kanálové)

- Digitální vstupy (např. na modulu NCU), zpětně čitelné zobrazení digitálních výstupů existující uvnitř řídícího systému ($A_IN, $A_OUT)
- Stav kanálu ($AC_STAT)
- Signály VDI ($A_DBB)
- Součtová hlášení určitého počtu alarmů ($AC_ALARM_STAT)

Osové zdroje

- Mezní hodnota pro nouzový zpětný pohyb vlečné osy (synchronní chod elektronické spojky, $VC_EG_SYNCDIFF[<vlečná osa>])
- Pohon: Mezní hodnota výstrahy meziobvodu (hrozící příliš nízké napětí), $AA_ESR_STAT[<Osa>]
- Pohon: Spodní mezní hodnota otáček generátoru (již není k dispozici žádná další rotační energie, která by se mohla použít pro napájení z druhé strany), $AA_ESR_STAT[<Osa>].

Spojovací logika statických synchronních akcí: Vazba zdroj/reakce

Aby se na základě zdrojových událostí relativně brzy spouštěly určité reakce, používají se flexibilní možnosti vazeb ve statických synchronních akcích.

Zajištění spojení všech důležitých zdrojů pomocí statických synchronních vazeb má plně v rukou uživatel. Systémové proměnné zdrojů může vyhodnocovat buď jako celek nebo také selektivně pomocí bitových masek a k tomu připojovat požadované reakce. Statické synchronní akce se uplatňují ve všech provozních režimech.

Aktivování

Odblokování funkcí

Pro odblokování funkcí generátorického režimu, zastavování a odjíždění se používá nastavení odpovídajících řídících signálů $AA_ESR_ENABLE. Tyto řídící signály mohou být ze synchronních akcí měněny.

Spouštění funkcí

ESR se spouští společně pro všechny odblokované osy nastavením systémové proměnné $AC_ESR_TRIGGER.

Generátorický režim se aktivuje v pohonu "automaticky", kdykoli je rozpoznán hrozící pokles napětí meziobvodu.

Zastavování a/nebo odjíždění nezávislé na pohonu se aktivuje při rozpoznání výpadku komunikace (mezi NC systémem a pohonem), ale také v případě zjištění poklesu napětí v meziobvodu v pohonu (za předpokladu příslušné konfigurace a odblokování funkcí).

Zastavení a/nebo odjíždění nezávislé na pohonu může být kromě toho spuštěno ještě i na straně NC systému prostřednictvím aktivování odpovídajících řídících signálů $AN_ESR_TRIGGER (příkaz přenášení do všech pohonů).

Další informace

Příručka k funkcím, Osy a vřetena

4.21.1 ESR řízené NC systémem

4.21.1.1 Odjíždění řízené NC systémem (POLF, POLFA, POLFMASK, POLFMLIN)

Pro odjíždění řízené NC systémem musí být splněny určité výchozí podmínky (viz "Odjíždění řízené NC systémem (POLF, POLFA, POLFMASK, POLFMLIN) (Strana 988)"). Jsou-li tyto výchozí předpoklady splněny, potom se pro osy provádějící odjíždění, které jsou nastavené v konfiguraci kanálu, aktivuje nastavením systémových proměnných $AC_ESR_TRIGGER (příp. $AA_ESR_TRIGGER pro samostatné osy) operace rychlého pozvednutí (LIFTFAST).

Syntaxe

```
POLF(<Osa>)=<Poloha>
POLFA(<Osa>,<Typ>,<Poloha>)
POLFMASK(<Osa_1>,<Osa_2>,...)
POLFMLIN(<Osa_1>,<Osa_2>,...)
```

U příkazu `POLFA` jsou přípustné následující zkrácené způsoby zápisu:

```
POLFA(<Osa>,<Typ>) ; zkrácená forma pro odjíždění samostatné osy
POLFA(Osa,0/1/2) ; rychlé deaktivování nebo aktivování
POLFA(Osa,0,$AA_POLFA[Osa]) ; způsobí zastavení předběžného
zpracování
POLFA(Osa,0) ; nezpůsobí žádné zastavení předběžného zpracování
```

Význam

POLF:	Adresa pro zadání cílové pozice osy, která má odjíždět Příkaz POLF má modální působnost..		
	<osa>:	Název geometrické nebo kanálové/strojní osy, která má provádět odjíždění	
	<pozice>:	Pozice pro odjíždění	
		Typ:	REAL
		Pro geometrické osy platí WCS, jinak MCS. V případě stejných identifikátorů pro geometrické a kanálové/strojní osy se odjíždění provádí ve WCS.	
POLFA:	Volání předem definovaného podprogramu pro zadání pozice pro zpětný pohyb pro jednotlivé osy		
	<osa>:	Identifikátor kanálové osy	
	<typ>:	Režim zadávání polohy	
		Typ:	INT
		Hodnota:	0: Označení hodnoty polohy jako neplatné
			1: Hodnota polohy je absolutní
			2: Hodnota polohy je inkrementální (vzdálenost)
		Upozornění: Pokud osa není samostatnou osou nebo pokud chybí typ, příp. Typ = 0, aktivuje se odpovídající alarm.	
	<pozice>:	Pozice pro zpětný pohyb (viz výše)	
		Upozornění: Hodnota pozice se přebírá, i když je nastaveno Typ = 0. Tato hodnota je jen potom označena jako neplatná, a aby se zpětný pohyb uskutečnil, musí být naprogramována znovu.	
POLFMASK:	Volání předdefinovaného programu pro volbu os, u kterých se má po spuštění rychlého pozvednutí uskutečňovat zpětný pohyb **nezávisle na sobě**.		
	<Osa_1>,...:	Názvy os, které mají při rychlém pozvednutí najíždět na pozice definované příkazem POLF. Všechny uvedené osy se musejí nacházet ve stejném souřadném systému.	
	Příkaz POLFMASK() bez udání osy deaktivuje operaci rychlého pozvednutí pro všechny osy, které mají uskutečňovat zpětný pohyb nezávisle na sobě.		
POLFMLIN:	Volání předdefinovaného programu pro volbu os, u kterých se má po spuštění rychlého pozvednutí uskutečňovat zpětný pohyb **v lineárním vztahu**.		
	<Osa_1>,...:	viz výše	
	Příkaz POLFMLIN() bez udání osy deaktivuje operaci rychlého pozvednutí pro všechny osy, které mají uskutečňovat zpětný pohyb v lineárním vztahu.		

Poznámka

Předtím, než může být rychlé pozvednutí na pevně danou pozici pomocí příkazů POLFMASK nebo POLFMLIN uvolněno, musí být pro vybrané osy naprogramována pomocí příkazu POLF požadovaná pozice.

Poznámka

Jestliže jsou osy uvolněny postupně pomocí příkazů POLFMASK, POLFMLIN nebo POLFMLIN, POLFMASK, platí pro příslušnou osu vždy poslední definice.

Poznámka

Polohy naprogramované pomocí příkazu POLF a aktivování prostřednictvím funkce POLFMASK nebo POLFMLIN jsou při spuštění výrobního programu vymazány. To znamená, že uživatel musí v každém výrobním programu znovu naprogramovat hodnoty pro příkaz POLF a osy vybrané v příkazech POLFMASK, příp. POLFMLIN

Poznámka

Pokud se při použití zkrácených forem příkazu POLFA změní pouze parametr Typ, pak musí uživatel zajistit, že parametry pro polohu zpětného pohybu nebo pro odjížděcí dráhu obsahují smysluplnou hodnotu. Polohu pro zpětný pohyb a odjížděcí dráhu je zapotřebí znovu nastavit zejména po spuštění systému (Power On).

Příklad:

Zpětný pohyb samostatné osy:

Programový kód	**Komentář**
`MD37500 $MA_ESR_REACTION[AX1]=21`	`; Odjíždění řízené NC systémem.`
`...`	
`$AA_ESR_ENABLE[AX1]=1`	
`POLFA(AX1,1,20.0)`	`; Ose AX1 je přiřazena axiální poloha zpětného pohybu 20,0 (absolutně).`
`$AA_ESR_TRIGGER[AX1]=1`	`; Od tohoto místa začíná zpětný pohyb.`

Další informace

Předpoklady pro odjíždění řízené NC systémem

- V konfiguraci kanálu je nastavena osa určená pro odjíždění řízené NC systémem:
  MD37500 $MA_ESR_REACTION = 21
- Pro tuto osu nesmí být ESR blokováno:
  $AA_ESR_ENABLE = 1
- Jsou definována časová zpoždění:
  MD21380 $MC_ESR_DELAY_TIME1
  MD21381 $MC_ESR_DELAY_TIME2
- Ve výrobním programu jsou pro příslušné osy pomocí příkazu POLF naprogramovány pozice pro zpětný pohyb.
- Osy jsou pomocí příkazů POLFMASK/POLFMLIN vybrány pro odjíždění řízené NC systémem.
- Pro zpětný pohyb musí být nastaveny signály zajišťující odblokování a tyto signály musí v tomto stavu zůstat.

Uvolnění a spouštění odjíždění řízeného NC systémem

Jestliže je nastavena systémová proměnná$AC_ESR_TRIGGER = 1 a jestliže je v konfiguraci tohoto kanálu nastavena osa pro zpětný pohyb (tzn. MD37500 $MA_ESR_REACTION = 21) a pro tuto osu je nastavena proměnná $AA_ESR_ENABLE = 1, pak se v tomto kanálu aktivuje rychlé pozvednutí (LIFTFAST).

Pozvedávací pohyb, jehož konfigurace je nastavena pomocí funkcí `POLF` (příp. `LFPOS`), os vybraných pomocí příkazů `POLFMASK` nebo `POLFMLIN` nahrazuje pro tyto osy pohyb po dráze definovaný ve výrobním programu.

Pro odjetí je k dispozici maximálně součet časů MD21380 $MC_ESR_DELAY_TIME1 a MD21381 $MC_ESR_DELAY_TIME2. Po uplynutí tohoto řasového intervalu se i pro osy provádějící zpětný pohyb spustí rychlé zabrždění s následným monitorováním.

Poznámka

Rozšířený zpětný pohyb (tzn. pomocí proměnné $AC_ESR_TRIGGER spuštěná funkce LIFTFAST/ LFPOS) **nemůže být přerušen** a může být předčasně ukončen pouze nouzovým zastavením.

Poznámka

Odjíždění spouštěné pomocí proměnné $AC_ESR_TRIGGER je blokováno vůči několikanásobnému zpětnému pohybu.

Odjíždění samostatnou osou

Při odjíždění samostatnou osou musí být pomocí příkazu `POLFA` naprogramována pozice samostatné osy pro zpětný pohyb a musí být splněny následující podmínky:

- $AA_ESR_ENABLE = 1
- <Osa> musí být v okamžiku spuštění ($A**A**_ESR_TRIGGER = 1) samostatnou osou.
- Parametr <Typ> musí mít hodnotu 1 nebo 2.

Směr odjíždění při rychlém pozvednutí

V okamžiku aktivování rychlého pozvednutí se bere ohled na momentálně platný frame.

Poznámka

Framy s otočením ovlivňují pomocí příkazu `POLF` také směr, ve kterém se pozvednutí nástroje uskutečňuje.

Výměna osy

Osy provádějící zpětný pohyb jsou vždy přiřazeny právě jednomu kanálu a nesmí být vyměňovány mezi kanály. Dojde-li k pokusu vyměnit osu provádějící odjíždění do jiného kanálu, aktivuje se alarm. Teprve poté, co byla osa pomocí proměnné $AA_ESR_ENABLE[AX] = 0 opět deaktivována, může být vyměněna do nového kanálu. Po dokončení výměny mohou být osy znovu pomocí nastavení proměnné $AA_ESR_ENABLE[AX] = 1 ovládány.

Neutrální osy

Neutrální osy nejsou schopny ESR řízené NC systémem uskutečňovat.

4.21.1.2 Zastavení NC systémem

Pro osy provádějící zastavení, které jsou nastavené v konfiguraci kanálu, aktivuje nastavením systémových proměnných $AC_ESR_TRIGGER (příp. $AA_ESR_TRIGGER pro samostatné osy) operace zastavení řízeného NC systémem.

Předpoklady

- V konfiguraci kanálu je nastavena osa určená pro zastavení řízené NC systémem:
  MD37500 $MA_ESR_REACTION = 22
- Pro tuto osu nesmí být ESR blokováno:
  $AA_ESR_ENABLE = 1
- Jsou definována časová zpoždění:
  MD21380 $MC_ESR_DELAY_TIME1 (časové zpoždění pro ESR os)
  MD21381 $MC_ESR_DELAY_TIME2 (čas ESR pro interpolační brždění)

Postup

Po dobu časového intervalu v parametru MD21380 provádí osa nerušený interpolační pohyb, jak bylo naprogramováno. Po uplynutí časového intervalu daného parametrem MD21380 se spustí brždění řízené pomocí interpolace. Pro brždění řízené pomocí interpolace je potom k dispozici maximálně časové interval daný parametrem MD21381 . Po uplynutí tohoto časového intervalu se spouští rychlé brždění s následným monitorováním.

Příklad:

Zastavení samostatné osy:

Programový kód	**Komentář**
`MD37500 $MC_ESR_REACTION[AX1]=22`	`; zastavení řízené NC systémem`
`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`	`; od tohoto místa začíná zastavování`

4.21.2 ESR nezávislé na pohonu

4.21.2.1 Konfigurace zastavování nezávislého na pohonu (ESRS)

Pomocí funkce `ESRS(...)` se nastavuje konfigurace parametrů pohonu pro "zastavování" v rámci funkce ESR nezávislé na pohonu.

Syntaxe

```
ESRS(<osa_1>,<doba zastavení_1>[,...,<osa_n>,<doba zastavení_n>])
```

Význam

`ESRS(...)`:	Funkce pro zápis parametrů pohonu pro funkci ESR "zastavení" Funkce: • musí se nacházet v samostatném bloku • spouští předběžné zpracování • nemůže se používat v synchronních akcích	
`<osa_1>`, `...`, `<osa_n>`:	Osa, pro kterou má být konfigurováno zastavování nezávislé na pohonu. V pohonu se pro tuto osu zapisuje parametr pohonu p0888 (konfigurace): p0888 = 1	
	Typ:	AXIS
	Rozsah hodnot:	Identifikátor kanálové osy
`<doba zastavení_1>`, `...`, `<doba zastavení_n>`:	Časový interval po který se pohon poté, co se vyskytla chyba, pohybuje konstantně s momentální požadovanou hodnotou otáček. V pohonu se pro uvedenou osu zapisuje parametr pohonu p0892 (časový interval): p0892 = <doba zastavení>	
	Jednotka:	s
	Typ:	REAL
	Rozsah hodnot:	0.00 - 20.00
V jednom volání funkce může být naprogramováno maximálně 5 os; n = 5		

4.21.2.2 Konfigurace odjíždění nezávislého na pohonu (ESRR)

Pomocí funkce `ESRR(...)` se nastavuje konfigurace parametrů pohonu pro "odjíždění" v rámci funkce ESR nezávislé na pohonu.

Syntaxe

```
ESRR(<osa_1>,<návratová dráha_1>,<návratová
rychlost_1>[,...,<osa_n>,<návratová dráha_n>,<návratová
rychlost_n>])
```

Význam

`ESRR(...):`	Funkce pro zápis parametrů pohonu pro funkci ESR "odjíždění" Funkce: • musí se nacházet v samostatném bloku • spouští předběžné zpracování • nemůže se používat v synchronních akcích	

`<osa_1>,`
`...,`
`<osa_n>:`
	Osa, pro kterou má být konfigurováno odjíždění nezávislé na pohonu
	V pohonu se pro tuto osu zapisuje parametr pohonu p0888 (konfigurace):
	p0888 = 2	

	Typ:	AXIS
	Rozsah hodnot:	Identifikátor kanálové osy

`<návratová dráha_1>,`
`...,`
`<návratová dráha_n>:`
	Návratová dráha se pro pohon přepočítává na návratové otáčky. Hodnota se pro uvedenou osu zapisuje do parametru pohonu p0893 (otáčky):
	p0893 = (<návratová dráha_n> přepočítaná na návratové otáčky)	

	Jednotka:	mm/min, palce/min, stupně/min (v závislosti na jednotkách nastavených pro osu)
	Typ:	REAL
	Rozsah hodnot:	MIN - MAX

`<návratová rychlost_1>,`
`...,`
`<návratová rychlost_n>:`
	Návratová rychlost se pro pohon přepočítává na časový interval. Hodnota se pro uvedenou osu zapisuje do parametru pohonu p0892 (časový interval) [s]:
	p0892 = <návratová dráha_n> / <návratová rychlost_n>	

	Jednotka:	mm/min, palce/min, stupně/min (v závislosti na jednotkách nastavených pro osu)
	Typ:	REAL
	Rozsah hodnot:	0.00 - MAX
V jednom volání funkce může být naprogramováno maximálně 5 os; n = 5		

4.22 Doba zpracování programu / počítadlo obrobků

Za účelem podpory pracovníků obsluhy obráběcích strojů jsou k dispozici informace týkající doby zpracovávání programu a počtu kusů.

Tyto informace mohou být zpracovávány jako systémové proměnné v NC programu a/nebo v programu PLC. Současně jsou tyto informace k dispozici pro zobrazování na uživatelském rozhraní.

4.22.1 Doba zpracování programu

Funkce "Doba zpracování programu" dává k dispozici interní časovač NC systému pro monitorování technologických procesů, jehož hodnotu je možné načítat ve výrobním programu a v synchronních akcích pomocí systémových proměnných kanálu a NC systému.

Spouštění měření doby zpracování ($AC_PROG_NET_TIME_TRIGGER) je jediná systémová proměnná této funkce, do které lze zapisovat, a slouží pro selektivní měření úseků programu, tzn. zapsáním hodnoty pro spuštění může být v NC programu měření času zastaveno a znovu spuštěno.

Systémové proměnné	Význam	Aktivita
V NC systému		
$AN_SETUP_TIME	Čas od posledního náběhu řídícího systému se standardními hodnotami ("studený start") v minutách Při každém náběhu řídícího systému se standardními hodnotami se automaticky nastaví na "0".	• vždy aktivní
$AN_POWERON_TIME	Čas od posledního normálního náběhu řídícího systému ("teplý start") v minutách Při každém normálním náběhu řídícího systému se automaticky nastaví na "0".	
V kanálu		
$AC_OPERATING_TIME	Celková doba, po kterou jsou zpracovávány NC-programy v automatickém provozním režimu v sekundách Hodnota se s každým náběhem řídícího systému se automaticky nastaví na "0".	• Aktivování prostřednictvím MD27860 • v provozním režimu "AUTO"
$AC_CYCLE_TIME	Doba zpracování zvoleného NC programu v sekundách Hodnota se s každým spuštěním nového NC programu automaticky nastaví na "0".	
$AC_CUTTING_TIME	Doba opracovávání v sekundách Měří se doba, po kterou se pohybují dráhové osy (minimálně jedna je aktivní) bez aktivního rychlého posuvu ve všech NC programech od stisknutí tlačítka NC-Start do konce programu/resetu NC systému. Měření této doby se přeruší také tehdy, když je aktivní doba prodlevy. Hodnota se při každém náběhu řídícího systému se standardními hodnotami se automaticky nastaví na "0".	
$AC_ACT_PROG_NET_TIME	Aktuální čistá doba zpracování momentálně vybraného NC programu v sekundách S každým spuštěním nového NC programu automaticky nastaví na "0".	• vždy aktivní • v provozním režimu "AUTO"
$AC_OLD_PROG_NET_TIME	Čistá doba zpracování právě správně a pomocí příkazu M30 ukončeného programu v sekundách	
$AC_OLD_PROG_NET_TIME_COUNT	Změny parametru $AC_OLD_PROG_NET_TIME Po náběhu systému je proměnná $AC_OLD_PROG_NET_TIME_COUNT nastavena na "0". Proměnná $AC_OLD_PROG_NET_TIME_COUNT se zvýší vždy, když řídící systém zapíše do proměnné $AC_OLD_PROG_NET_TIME novou hodnotu.	

Systémové proměnné	Význam		Aktivita
$AC_PROG_NET_TIME_TRIGGER	Spouštění pro měření doby zpracování:		• v provozním režimu "AUTO"
	0	Neutrální stav Spouštění není aktivní.	
	1	Ukončit Ukončí měření a zkopíruje hodnotu z proměnné $AC_ACT_PROG_NET_TIME do $AC_OLD_PROG_NET_TIME. $AC_ACT_PROG_NET_TIME se znovu nastaví na "0" a potom se zpustí od začátku.	
	2	Start Spustí se měření a proměnná $AC_ACT_PROG_NET_TIME se přitom nastaví na "0“. Proměnná $AC_OLD_PROG_NET_TIME zůstane nezměněna.	
	3	Zastavit Zastaví měření: Proměnná $AC_OLD_PROG_NET_TIME se nemění a obsahuje hodnotu proměnné $AC_ACT_PROG_NET_TIME konstantní, dokud se nespustí další pokračování.	
	4	Pokračovat Pokračování měření, tzn. dříve zastavené měření bude pokračovat dál. $AC_ACT_PROG_NET_TIME bude běžet dále. $AC_OLD_PROG_NET_TIME zůstane nezměněna.	
Při náběhu systému (Power-On) jsou všechny systémové proměnné nastaveny na "0"!			

Poznámka

Výrobce stroje

Spuštění aktivovatelného časovače se uskutečňuje pomocí strojního parametru MD27860 $MC_PROCESSTIMER_MODE.

Chování aktivního měření času v případě určitých funkcí (např. GOTOS, korekce (override) = 0%, aktivní posuv při zkušebním zpracování, zkouška programu, ASUP, PROG_EVENT, ...) se nastavuje v konfiguraci pomocí strojních parametrů MD27850 $MC_PROG_NET_TIMER_MODE a MD27860 $MC_PROCESSTIMER_MODE.

Další informace: Příručka k funkcím, Základní funkce

Poznámka

Zbývající čas pro obrobek

Jestliže mají být jeden po druhém vyráběny stejné obrobky, je možno pomocí následujících časových hodnot vypočítat zbývající čas pro daný obrobek:

- dobu opracovávání pro naposled vyrobený obrobek (viz $AC_OLD_PROG_NET_TIME)
- aktuální dobu opracovávání (viz $AC_ACT_PROG_NET_TIME)

Zbývající čas se vedle aktuální doby opracovávání vypisuje na uživatelském rozhraní.

Poznámka

Použití příkazu STOPRE

Systémové proměnné $AC_OLD_PROG_NET_TIME a $AC_OLD_PROG_NET_TIME_COUNT nezpůsobují žádné implicitní zastavení předběžného zpracování. Při použití ve výrobním programu nelze poznat, kdy hodnota systémové proměnné pochází z předchozího zpracování programu. Jestliže se ale příkazy pro spuštění měření doby zpracování ($AC_PROG_NET_TIME_TRIGGER) zapisují s velmi vysokou frekvencí a hodnota v proměnné $AC_OLD_PROG_NET_TIME se v důsledku toho mění velmi často, pak by měl být ve výrobním programu použit explicitní příkaz STOPRE.

Okrajové podmínky

- **Vyhledávání bloku**
  Při vyhledávání bloku se žádná doba zpracovávání programu nezjišťuje.
- **REPOS**
  Doba trvání operace REPOS se do aktuální doby zpracovávání programu ($AC_ACT_PROG_NET_TIME) započítává.

Příklady

Příklad 1: Měření doby zpracovávání programu "mySubProgrammA"

Programový kód

```
...
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
```

Poté, co byl zpracován programový řádek N80, se v proměnné $AC_OLD_PROG_NET_TIME nachází čistá doba zpracování programu "mySubProgrammA".

Hodnota proměnné $AC_OLD_PROG_NET_TIME:

- zůstává zachována i za příkazem M30.
- po každém průchodu smyčkou se aktualizuje.

Příklad 2: Měření doby zpracovávání programů "mySubProgrammA" a "mySubProgrammC"

Programový kód

```
...
N10 DO $AC_PROG_NET_TIME_TRIGGER=2
N20 mySubProgrammA
N30 DO $AC_PROG_NET_TIME_TRIGGER=3
```

Programový kód

```
N40 mySubProgrammB
N50 DO $AC_PROG_NET_TIME_TRIGGER=4
N60 mySubProgrammC
N70 DO $AC_PROG_NET_TIME_TRIGGER=1
N80 mySubProgrammD
N90 M30
```

4.22.2 Počítadlo obrobků

Funkce "Počítadlo obrobků" poskytuje k dispozici různé čítače, které lze použít zejména pro počítání obrobků v rámci řídícího systému.

Počítadla existují jako specifické kanálové proměnné s oprávněními pro čtení a pro zápis a s rozsahem hodnot od 0 do 999 999 999.

Systémové proměnné	Význam
$AC_REQUIRED_PARTS	Počet obrobků, které mají být vyrobeny (požadovaná hodnota obrobků) V tomto čítači může být definován počet obrobků, při jehož dosažení se počet aktuálních obrobků v proměnné ($AC_ACTUAL_PARTS) nastaví zpět na "0".
$AC_TOTAL_PARTS	Celkový počet vyrobených obrobků (celková skutečná hodnota) Tento čítač udává počet všech obrobků vyrobených od okamžiku spuštění. Hodnota se automaticky nastaví na "0" jen při náběhu řídícího systému se standardními hodnotami.
$AC_ACTUAL_PARTS	Počet vyrobených obrobků (skutečná hodnota počtu obrobků) V tomto čítači je zaznamenáván počet všech obrobků vyrobených od okamžiku spuštění. Jestliže je dosaženo požadované hodnoty počtu kusů ($AC_REQUIRED_PARTS), čítač se automaticky nastaví zpět na "0" (předpokladem je, že $AC_REQUIRED_PARTS > 0).
$AC_SPECIAL_PARTS	Počet uživatelem počítaných obrobků Tento čítač umožňuje uživateli počítání obrobků podle jeho vlastní definice. Je možno také definovat aktivování alarmu, jakmile je dosaženo požadovaného počtu obrobků ($AC_REQUIRED_PARTS). Vynulování čítače musí provádět sám uživatel.

Poznámka

Všechna počítadla obrobků se při náběhu řídícího systému se standardními hodnotami nastavují na "0" a mohou být nezávisle na svém aktivování čteny nebo lze do nich zapisovat.

Poznámka

Aktivování čítačů, okamžik vynulování a algoritmus počítání je možno ovlivňovat pomocí kanálových strojních parametrů.

Poznámka

Počítání obrobků s příkazem M-funkce definovaným uživatelem

Pomocí strojního parametru může být nastaveno, že počítané impulzy pro různá počítadla obrobků mohou být spouštěny nikoli pomocí příkazu pro konec programu `M2`/`M30`, ale pomocí uživatelem definované M-funkce.

4.23 Simulace programu

4.23.1 Funkce

Při simulaci programu řídící systém kompletně vypočítává aktuální výrobní program a výsledky zobrazuje v grafické podobě na uživatelském rozhraní. Aniž by se osy stroje fyzicky pohybovaly, je možné takto zkontrolovat výsledek programování. Špatně naprogramované kroky opracování jsou tak zavčasu rozpoznány, takže se předejde chybnému opracování obrobku.

Simulační funkce v NC systému

Simulace využívá svůj vlastní objekt v rámci NC systému (simulační funkce). Z tohoto důvodu musí být před zahájením simulace uskutečněna kalibrace reálného NC systému vůči jeho simulační funkci. V rámci této kalibrace se mimo jiné z NC systému načítají všechny aktivní strojní parametry a ukládají se do simulační funkce NC systému. V rámci aktivních strojních parametrů se nacházejí strojní parametry NC systému a cyklů.

4.23.2 Simulace programu ve spojení s cykly překladače

Při simulaci programu jsou podporovány i některé cykly překladače (CC). Strojní parametry podporovaných cyklů překladače jsou po náběhu řídícího systému jedenkrát porovnány, zda se shodují. Toto porovnání se při operaci „spuštění simulace“ už neuskutečňuje!

Poznámka

Specifické příkazy jazyka a strojní parametry nepodporovaných cyklů překladače se ve výrobních programech **nesmí** používat (viz také odstavec „Příkazy cyklů překladače ve výrobním programu“).

Speciální pohyby vyvolávané podporovanými cykly překladače (transformace OEM) se za určitých okolností zobrazují nesprávně.

Příkazy cyklů překladače ve výrobním programu

Příkazy jazyka ve výrobním programu, které spadají do nepodporovaných cyklů překladače (`OMA1` ... `OMA5`, `OEMIPO1/2`, `G810` ... `G829`, některé procedury a funkce), pokud nejsou individuálně ošetřeny, mají za následek alarmové hlášení a přerušení simulace.

Řešení:

Pro chybějící specifické prvky jazyka týkající se cyklů překladače ve výrobním programu zajistěte individuální zpracování (zjišťování stavu $P_SIM).

Příklad:

Programový kód	**Komentář**
N1 G01 X200 F500	
IF (1==$P_SIM)	
N5 X300	; Při simulaci CC nejsou aktivní.
ELSE	
N5 X300 OMA1=10	
ENDIF	

4.24 Další funkce

4.24.1 Aktivování strojních parametrů (NEWCONF)

Pomocí příkazu NEWCONF jsou všechny strojní parametry nastaveny jako platné. Tato funkce může být aktivována také v uživatelském rozhraní HMI pomocí programového tlačítka "Aktivovat MD".

V rámci zpracovávání funkce "NEWCONF" se provádí implicitní zastavení předběžného zpracování, tzn. pohyb po dráze se přeruší.

Syntaxe

```
NEWCONF
```

Význam

NEWCONF:	Příkaz pro aktivování všech strojních parametrů do stavu platnosti "NEW_CONFIG".

Vyvolání příkazu NEWCONF z výrobního programu přes hranice kanálu

Jestliže jsou strojní parametry v důsledku zpracování výrobního programu změněny a potom jsou příkazem NEWCONF aktivovány, nastaví příkaz NEWCONF do aktivního stavu jen ty strojní parametry, které způsobují změny pro kanál výrobního programu.

Poznámka

Aby se všechny změny bezpečně aktivovaly do daného stavu platnosti, musí být příkaz NEWCONF uskutečněn v každém kanálu, ve kterém jsou momentálně vypočítávány také osy nebo funkce, které souvisejí se změněnými strojními parametry.

Příkazem NEWCONF nejsou aktivovány žádné osové strojní parametry.

V případě os ovládaných pomocí PLC musí být spuštěna funkce "reset osy".

Příklad:

Obrábění frézováním: Opracování pozic pro vrtání pomocí různých technologií

Programový kód	Komentář
N10 $MA_CONTOUR_TOL[AX]=1.0	; změna strojního parametru
N20 NEWCONF	; aktivování strojních parametrů
...	

4.24.2 Kontrola existujícího rozsahu NC jazyka (STRINGIS)

Pomocí funkce "STRINGIS(...)" je možné zkontrolovat, zda je zadaný řetězec k dispozici jako prvek aktuálního rozsahu programovacího jazyka NC systému.

Definice

```
INT STRINGIS(STRING <název>)
```

Syntaxe

```
STRINGIS(<název>)
```

Význam

STRINGIS:	Funkce s výstupní hodnotou
<název>:	Název zkoumaného prvku programovacího jazyka NC systému
Výsledná hodnota:	Výsledná hodnota má formát yxx (desetinný formát).

Prvky programovacího jazyka NC systému

Mohou být kontrolovány následující prvky programovacího jazyka NC systému:

- Příkazy G-funkcí všech existujících skupin G-funkcí, např. „G0“, „INVCW“, „POLY“, „ROT“, „KONT“, „SOFT“, „CUT2D“, „CDON“, „RMBBL“, „SPATH“
- DIN- nebo NC-adresy, jako např. „ADIS“, „RNDM“, „SPN“, „SR“, „MEAS“
- Funkce, jako např. „TANG(...)“ nebo „GETMDACT“
- Procedury, např. „SBLOF“.
- Klíčová slova, např. „ACN“, „DEFINE“ nebo „SETMS“
- Systémová data, např. strojní parametry $M... , nastavované parametry $S... nebo data volitelných doplňků $O...
- Systémové proměnné $A... , $V... , $P...
- Početní parametry R...
- Názvy aktivovaných cyklů
- Proměnné GUD a LUD
- Názvy maker
- Názvy návěští

Výsledná hodnota

Pro výslednou hodnotu jsou důležité pouze první tři desetinná místa. Výsledná hodnota má formát yxx, kde y = základní informace a xx = podrobné informace.

Výsledná hodnota	**Význam**	
000	Řetězec "název" není v předkládaném systému znám [1]	
100	Řetězec "název" je prvek programovacího jazyka NC systému, momentálně ale nemůže být naprogramován (volitelný doplněk/funkce jsou neaktivní)	
2xx	Řetězec "název" je naprogramovatelný prvek programovacího jazyka NC systému (volitelný doplněk/funkce jsou aktivní) Podrobné informace xx obsahují další údaje o druhu prvku:	
	xx	**Význam**
	01	Adresa podle normy DIN nebo NC adresa [2]
	02	Příkaz G-funkce (např. G04, INVCW)
	03	Funkce s výstupní hodnotou
	04	Funkce bez výstupní hodnoty
	05	Klíčové slovo (např. DEFINE)
	06	Strojní parametry ($M...), nastavované parametry ($S...) nebo parametry volitelných doplňků ($O...)
	07	Systémové parametry, např. systémové proměnné ($...) nebo početní parametry (R...)
	08	Cyklus (cyklus musí být načten v NC systému a program cyklu musí být aktivní [3])
	09	Proměnná GUD (proměnné GUD musí být definovány v definičním souboru GUD a proměnné GUD musí být aktivovány)
	10	Název makra (makro musí být definováno v definičním souboru makra a makra musí být aktivována) [4]
	11	Proměnná LUD aktuálního výrobního programu
	12	Příkaz G-funkce podle normy ISO (režim jazyka ISO musí být aktivní)

Výsledná hodnota	Význam
400	Řetězec "název" je NC-adresa, která nebyla rozpoznána jako xx = 01 nebo xx = 10 a není typu G nebo R [2)]
y00	Není možné žádné specifické přiřazení

1) V závislosti na řídícím systému je za určitých okolností známá pouze určitá podmnožina příkazů programovacího jazyka NC systému, např. v případě systému SINUMERIK 802D sl. U těchto řídících systémů je pro řetězce, které jsou v principu příkazy NC jazyka firmy Siemens, výsledkem hodnota 0. Toto chování může být pomocí strojního parametru MD10711 $MN_NC_LANGUAGE_CONFIGURATION změněno. Je-li nastaveno MD10711 = 1, potom bude pro příkazy NC jazyka firmy Siemens vždy výsledkem hodnota 100.

2) NC adresami jsou následující písmena: A, B, C, E, I, J, K, Q, U, V, W, X, Y, Z. Tyto NC adresy mohou být naprogramovány také s rozšířením adresy. Rozšíření adresy může být zadáno při kontrole pomocí funkce STRINGIS. Příklad: 201 == STRINGIS("A1"). Písmena: D, F, H, L, M, N, O, P, S, T jsou NC adresy nebo pomocné funkce, které jsou používány podle uživatelských definic. V jejich případě je výsledkem vždy hodnota 400. Příklad: 400 == STRINGIS("D"). Tyto NC adresy mohou být zadány při kontrole pomocí funkce STRINGIS, ale bez rozšíření adresy.
Příklad: 000 == STRINGIS("M02"), ale 400 == STRINGIS("M").

3) Názvy parametrů cyklů nemohou být pomocí funkce STRINGIS kontrolovány.
4) Adresa definovaná jako makro, např.. G, H, M, L, je jako makro identifikována.

Příklady

V následujících příkladech se předpokládá, že řetězec zadaný jako prvek jazyka NC systému, pokud není uvedeno jinak, může být v principu v NC systému naprogramován.

1. Řetězec "T" je definován jako pomocná funkce:
```
400 == STRINGIS("T")
000 == STRINGIS ("T3")
```
2. Řetězec "X" je definován jako osa:
```
201 == STRINGIS("X")
201 == STRINGIS("X1")
```
3. Řetězec "A2" je definován jako NC adresa s rozšířením:
```
201 == STRINGIS("A")
201 == STRINGIS("A2")
```
4. Řetězec "INVCW" je definován jako specifický příkaz G-funkce:
```
202 == STRINGIS("INVCW")
```
5. Řetězec "$MC_GCODE_RESET_VALUES" je definován jako strojní parametr:
```
206 == STRINGIS("$MC_GCODE_RESET_VALUES")
```
6. Řetězec "GETMDACT" je funkce NC-jazyka:
```
203 == STRINGIS("GETMDACT ")
```
7. Řetězec "DEFINE" je klíčové slovo:
```
205 == STRINGIS("DEFINE")
```
8. Řetězec "$TC_DP3" je systémový parametr (složka délky nástroje):
```
207 == STRINGIS("$TC_DP3")
```
9. Řetězec "$TC_TP4" je systémový parametr (velikost nástroje):
```
207 == STRINGIS("$TC_TP4")
```

10. Řetězec "$TC_MPP4" je systémový parametr (stav místa v zásobníku):
    - Správa zásobníku nástrojů je aktivní. `207 == STRINGIS("$TC_MPP4") ;`
    - Správa zásobníku nástrojů není aktivní. `000 == STRINGIS("$TC_MPP4")`

    Viz také kapitola: Správa zásobníku nástrojů.
11. Řetězec "MACHINERY_NAME" je definován jako proměnná typu GUD.
    `209 == STRINGIS("MACHINERY_NAME")`
12. Řetězec "LONGMACRO" je definován jako makro:
    `210 == STRINGIS("LONGMACRO")`
13. Řetězec "MYVAR" je definován jako proměnná typu LUD.
    `211 == STRINGIS("MYVAR")`
14. Řetězec "XYZ" není žádný příkaz, který by byl v NC systému známý, není ani názvem proměnné GUD, makra nebo cyklu:
    `000 == STRINGIS("XYZ")`

Správa zásobníku nástrojů

Jestliže funkce "Správa zásobníku nástrojů" není aktivní, poskytuje funkce STRINGIS pro systémové parametry správy zásobníku nástrojů, nezávisle na strojním parametru, následující výsledky:

- MD10711 $MN_NC_LANGUAGE_CONFIGURATION

Vždy hodnota 000.

Režim ISO

Pokud je funkce "Režim ISO" aktivní:

- MD18800 $MN_MM_EXTERN_LANGUAGE (aktivování externích BC jazyků)
- MD10880 $MN_ MM_EXTERN_CNC_SYSTEM (řídící systémy, které se mají přizpůsobit)

kontroluje funkce STRINGIS zadaný řetězec napřed jako příkaz G-funkce systému SINUMERIK. Pokud řetězec není příkazem G-funkce systému SINUMERIK, zkontroluje se potom, zda se nejedná o příkaz G-funkce podle normy ISO.

Naprogramovaná přepnutí (`G290` (režim SINUMERIK), `G291` (režim ISO)) nemají na funkci STRINGIS žádný vliv.

Příklad:

Strojní parametry, které jsou pro funkci STRINGIS(...) důležité, mají následující hodnoty:

- MD10711 $MN_NC_LANGUAGE_CONFIGURATION = 2 (jako známé jsou rozpoznány jen ty příkazy NC-jazyka, jejichž volitelné doplňky jsou aktivovány)
- MD19410 $ON_TRAFO_TYPE_MASK = 'H0' (volitelný doplněk: transformace)
- MD10700 $MN_PREPROCESSING_LEVEL='H43' (předběžné zpracování pro cykly je aktivní)

Program uvedený v následujícím příkladu se zpracuje bez chybového hlášení:

Programový kód	**Komentář**
N1 R1=STRINGIS("TRACYL")	R1 == 0, protože příkaz TRACYL je kvůli chybějícímu volitelnému doplňku "Transformace" rozpoznán jako "neznámý".
N2 IF STRINGIS("TRACYL") == 204	
N3 TRACYL(1,2,3)	; N3 se bude přeskakovat
N4 ELSE	
N5 G00	; a místo něj se bude zpracovávat N5
N6 ENDIF	
N7 M30	

4.24.3 Interaktivní vyvolávání oken z výrobního programu (MMC)

Prostřednictvím předem definovaného podprogramu MMC(...) mohou být z NC programu zobrazovány na uživatelském rozhraní specifická uživatelská dialogová okna.

Dialogová okna mohou být navrhována pro následující druhy těchto objektů:

- Run MyScreens
- Easy XML
- Uživatelské XML

Další informace:

- Run MyScreens, příručka programátora
- Easy XML, příručka programátora

Syntaxe

```
MMC("<ADDRESS>,<COMMAND>,<FILE>,<DIALOG>","<QUIT>")
```

Význam

`MMC(...)`:	Identifikátor podprogramu Parametry mají kódovaná místa a čárkou zadávají se rozdělené do dvou řetězců, což jsou příkazový řetězec a potvrzovací řetězec.
Parametry v rámci příkazového řetězce:	

<ADDRESS>:	Systémová oblast, ve které se v konfiguraci nastavený dialog s uživatelem uskutečňuje.	
	Funkce	**Systémové oblasti**
	Uživatelské dialogové okno „Run MyScreens“	CYCLES
	Uživatelské dialogové okno „Easy XML“	CYCLES
	Uživatelské XML	XML
	Okno v popředí „Run MyScreens“	POPUPDLG
	Okno v popředí „Easy XML“	POPUPDLG
<COMMAND>:	Příkaz, který má být zpracován	
	Funkce	**Příkazy**
	Uživatelské dialogové okno „Run MyScreens“	PICTURE_ON, PICTURE_OFF
	Uživatelské dialogové okno „Easy XML“	PICTURE_ON, PICTURE_OFF
	Uživatelské XML	XML_ON, XML_OFF
	Okno v popředí „Run MyScreens“	PICTURE_ON, PICTURE_OFF
	Okno v popředí „Easy XML“	PICTURE_ON, PICTURE_OFF
<FILE>:	Název souboru, ve kterém je naprogramováno dialogové okno, které se má zobrazit	
	Funkce	**Soubory**
	Uživatelské dialogové okno „Run MyScreens“	<name>.com
	Uživatelské dialogové okno „Easy XML“	<name>.xml
	Uživatelské XML	<name>.xml
	Okno v popředí „Run MyScreens“	<name>.com
	Okno v popředí „Easy XML“	<name>.xml
	Okno v popředí „Easy XML“ sestavené přímo v NC programu (viz příklad 2)	xmldial_emb.xml
<DIALOG>:	Název dialogového okna, které se má zobrazovat	
	Funkce	**Název dialogového okna**
	Všechny funkce kromě okna v popředí „Easy XML“ sestavené přímo v NC programu	Název dialogového okna zadaný v souboru <FILE>
	Okno v popředí „Easy XML“ sestavené přímo v NC programu (viz příklad 3)	main
Parametr v rámci potvrzovacího řetězce:		

<QUIT>:	Typ potvrzování	
	N:	Žádné potvrzování. Zpracování programu se uskutečňuje po odeslání příkazu. Jestliže příkaz nebylo možné úspěšně zpracovat, nejste o tom nijak informováni. **Poznámka** Typ potvrzování „N" je potřeba použít, jestliže je v NC programu naprogramována doba zobrazování (doba prodlevy) (viz příklad 2 níže)
	A:	Asynchronní potvrzování Zpracování programu se uskutečňuje po odeslání příkazu. Výstupní hodnota se ukládá do specifické uživatelské potvrzovací proměnné (proměnné GUD) definované v rámci vytváření dialogového okna a je možno ji v NC programu načíst.

Okrajové podmínky

- Definiční soubory *.com dialogových oken musí být uloženy ve složce „proj".
- Definiční soubory Easy XML *.xml dialogových oken musí být uloženy ve složce „appl". Pokud jsou definiční soubory uloženy v nějakém jiném adresáři, musí být nepřímo zadán údaj cesty, který vychází z adresáře „appl".
- Uživatelem definovaná dialogová okna nemohou být zobrazována současně z různých kanálů.
- Při simulaci nejsou funkce MMC podporovány.

Příklady

Příklad 1

Zobrazení dialogového okna a reakce na uživatelskou podmínku v NC programu.

```
; potvrzovací proměnná QUIT již byla deklarována jako globální uživatelská proměnná (GUD)
; typu STRING při vytváření dialogového okna:
; DEF NCK STRING[20] QUIT
QUIT = "XXX"                                ; inicializace potvrzovací proměnné
G4 F5
MMC("CYCLES,PICTURE_ON,test.com,test1","A")  ; zobrazení dialogového okna
                                            ; - systémová oblast: CYCLES
                                            ; - stav obrazovky: PICTURE_ON (zobrazovat)
                                            ; - soubor obrazovky s dialogového okna: test.com
                                            ; - obrazovka s dialogovým oknem test1
INPUT:                                      ; čekání na uživatelskou podmínku
   STOPRE                                   ; - zastavení předběžného zpracování
```

Programový kód	**Komentář**
IF MATCH (QUIT,"RUN") >= 0 GOTOF WORK	; - programové tlačítko „RUN“
IF MATCH (QUIT,"CHK") >= 0 GOTOF CHECK	; - programové tlačítko „CHK“
GOTOB INPUT	; => čekání
WORK:	; - programové tlačítko „RUN“ stisknuto
MSG(„Pokračovat ve zpracování -> NC-Start")	; výstup hlášení
MMC("CYCLES,PICTURE_OFF","N")	; zavření dialogového okna
M0	; Čekání na tlačítko NC-Start
GOTOF END	; => konec programu
CHECK:	; - programové tlačítko „CHK“ stisknuto
MSG(„Najíždění na pozici -> NC-Start")	; výstup hlášení
MMC("CYCLES,PICTURE_OFF","N")	; zavření dialogového okna
M0	; Čekání na tlačítko NC-Start
GOTOF END	; => konec programu
END:	
...	

Příklad 2

Doba zobrazování dialogového okna se v NC programu určuje např. dobou prodlevy.

Programový kód	**Komentář**
F1000 G94	
...	
MMC("POPUPDLG,PICTURE_**ON**,xmldial_emb.xml,main","N")	; zobrazení dialogového okna
X200	
Z40	
MMC("POPUPDLG,PICTURE_**OFF**","N")	; zavření dialogového okna

Příklad 3

Vložení skriptu okna v popředí do NC programu a jeho použití.

Programový kód

```
PROC POPUP_TEST
; ----------------------------- Skript -----------------------------
; <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>
```

Programový kód

```
;          <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>
;             <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>
; ========================= Úsek programu ==========================
```

Programový kód

```
...
G94 F100
MMC("POPUPDLG,PICTURE_ON,xmldial_emb.xml,main","N")
G4 F4
X200
MMC("POPUPDLG,PICTURE_OFF","N")
G4 F2
X0
...
```

4.24.4 Process DataShare - Výstup do externího zařízení/souboru (EXTOPEN, WRITE, EXTCLOSE)

Zápis dat z výrobního programu do externího zařízení/souboru probíhá ve třech krocích:

1. Otevření externího zařízení/souboru
   Pomocí příkazu EXTOPEN se pro kanál otevře externí zařízení/soubor, které lze použít pro zápis.
2. Zápis dat
   Výstupní data mohou být připravena pomocí funkcí NC jazyka pro práci s řetězci, např. SPRINT. Samotný zápis se uskutečňuje pomocí příkazu WRITE.
3. Zavření externího zařízení/souboru
   Pomocí příkazu EXTCLOSE nebo při dosažení konce programu (M30), jakož i při resetu kanálu, že v kanálu obsazený externí zařízení/soubor bude opět uvolněn.

Syntaxe

```
DEF INT <výsledek>
DEF STRING[<n>] <výstup>
…
EXTOPEN(<výsledek>,<ExtDev>,<SyncMode>,<AccessMode>,<WriteMode>)
…
<výstup>="výstup dat"
WRITE(<výsledek>,<ExtDev>,<Output>)
…
EXTCLOSE(<výsledek>,<ExtDev>)
```

Význam

EXTOPEN:	Předem definovaná procedura pro otevření externího zařízení/souboru		
<výsledek>:	**Parametr 1:** Proměnná pro výsledek Na základě hodnoty proměnné pro výsledek je možné v programu vyhodnocovat, jak operace skončila, a na základě toho potom odpovídajícím způsobem postupovat.		
	Typ:	INT	
	Hodnoty:	0	žádná chyba
		1	externí zařízení nebylo možné otevřít
		2	externí zařízení není konfigurováno
		3	pro externí zařízení je v konfiguraci zadána nesprávná cesta
		4	pro externí zařízení nemáte přístupová oprávnění
		5	Režim využívání: externímu zařízení je přiřazen "exkluzivní" přístup
		6	Režim využívání: externímu zařízení je přiřazen "sdílený" přístup
		7	Délka souboru je větší než kolik udává parametr LOCAL_DRIVE_MAX_FILESIZE
		8	je překročen maximální počet externích zařízení
		9	funkce pro LOCAL_DRIVE není aktivována
		11	rezervováno
		12	Režim zápisu: zadání je v rozporu se souborem extdev.ini
		16	naprogramována neplatná externí cesta
		22	externí zařízení není připojeno

<ExtDev>:	**Parametr 2:** Symbolický identifikátor pro otevírané externí zařízení/soubor		
	Typ:	STRING	
	Symbolický identifikátor se skládá z těchto částí: 1. název logického zařízení 2. v případě potřeby následovaný cestou k danému souboru (připojuje se pomocí znaku "/") Jsou definovány následující **názvy logických zařízení**:		
	"LOCAL_DRIVE":	Lokální kompaktní Flash-karta (předdefinováno)	
	"CYC_DRIVE":	Rezervované udání jednotky používané v cyklech firmy SIEMENS (předdefinováno)	
	"/dev/ext/1", ... "/dev/ext/9":	Síťové jednotky, které jsou k dispozici **Upozornění:** Je zapotřebí nastavit konfiguraci v souboru extdev.ini!	
	"/dev/cyc/1", "/dev/cyc/2":	Rezervované udání jednotky používané v cyklech firmy SIEMENS **Upozornění:** Je zapotřebí nastavit konfiguraci v souboru extdev.ini!	
	Cesta k souboru: • Pro "LOCAL_DRIVE" a "CYC_DRIVE" musí být zadána cesta k souboru, např.: "LOCAL_DRIVE/my_dir/my_file.txt" • Názvy logických zařízení "/dev/ext/1...9" a "/dev/cyc/1...2" mohou prostřednictvím konfigurace odkazovat: – na určitý soubor, potom smí být uváděny pouze názvy logických zařízení, např.: "/dev/ext/4" – nebo na adresář, potom ale musí být cesta k souboru zadána, např.: "/dev/ext/5/my_dir/my_file.txt"		
	Poznámka: Pro názvy logických zařízení "/dev/ext/1...9", "/dev/v24" a "/dev/cyc/1...2" nejsou velká a malá písmena rozlišována, u zadání cesty k souboru jsou velká a malá písmena rozlišována. Pro "LOCAL_DRIVE" a "CYC_DRIVE" jsou přípustná jen velká písmena.		
<SyncMode>:	**Parametr 3:** Režim zpracování pro příkazy WRITE k tomuto zařízení/souboru		
	Typ:	STRING	
	Hodnoty:	"SYN":	Synchronní zápis Zpracovávání programu je pozastaveno, dokud není operace zápisu dokončena. Úspěšné ukončení synchronního zápisu může být zkontrolováno pomocí vyhodnocování chybové proměnné příkazu WRITE.
		"ASYN":	Asynchronní zápis Zpracovávání programu není příkazem WRITE přerušeno. **Poznámka:** Proměnná pro výsledek příkazu WRITE nemá v tomto režimu žádnou vypovídací hodnotu a má vždy hodnotu 0 (žádná chyba). V tomto režimu si nemůžete být jisti, že byl příkaz WRITE úspěšně zpracován.

<AccessMode>:	**Parametr 4:** Režim využívání pro toto zařízení/tento soubor		
	Typ:	STRING	
	Hodnoty:	"SHARED":	Zařízení/soubor jsou přiřazovány ve sdíleném režimu. Zařízení může být používáno také jinými kanály, tzn. otevřenými rovněž v tomto režimu.
		"EXCL":	Zařízení/soubor jsou používány výlučně v tomto kanálu a žádný další kanál nemůže toto zařízení používat také.
<WriteMode>:	**Parametr 5:** Režim zápisu pro příkazy WRITE k tomuto zařízení/souboru (volitelné)		
	Typ:	STRING	
	Hodnoty:	"APP":	Připojit Obsah souboru zůstává zachován, zapisovaná data se vkládají na jeho konec.
		"OVR":	Přepisování Obsah souboru se vymaže a následující operací zápisu se znovu vytvoří.
	Poznámka: Režim zápisu nastavený v konfiguraci v souboru extdev.ini není možné pomocí tohoto parametru přepsat. V případě konfliktu je volání funkce EXTOPEN potvrzeno s chybou.		

WRITE:	Předem definovaná procedura pro zápis výstupních dat

EXTCLOSE	Předem definovaná procedura pro zavření otevřeného externího zařízení/souboru		
<výsledek>:	**Parametr 1:** Proměnná pro výsledek		
	Typ:	INT	
	Hodnoty:	0	žádná chyba
		16	naprogramována neplatná externí cesta
		21	chyba při zavírání externího zařízení
<ExtDev>:	**Parametr 2:** Symbolický identifikátor pro zavírané externí zařízení/soubor; popis naleznete v části věnované příkazu EXTOPEN! **Poznámka:** Tento identifikátor musí být identický s identifikátorem použitým ve volání pomocí příkazu EXTOPEN!		

Příklad:

Programový kód

```
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
```

Programový kód
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
…

Viz také

Operace s řetězci (Strana 448)

Zápis do souboru (WRITE) (Strana 570)

4.24.5 Alarmy (SETAL)

V NC programu mohou být aktivovány alarmy, které se pak vypisují na uživatelském rozhraní ve zvláštním poli. S alarmem je vždy spojena nějaká reakce řídícího systému závisející na kategorii alarmu.

Syntaxe

```
SETAL(<Č.>[,<řetězec>])
```

Význam

SETAL(...):	Předdefinovaná procedura pro aktivování alarmu Příkaz SETAL musí být naprogramován v samostatném NC-bloku.		
<č.>:	Číslo alarmu		
	Datový typ:	INT	
	Rozsah hodnot:	60000 ... 64999 (rezervováno)	Alarmy pro cykly SIEMENS
		65000 ... 69999	Alarmy pro uživatelské cykly

<řetězec>:	Řetězec znaků (nepovinný údaj)	
	Datový typ:	STRING
	Při programování alarmů uživatelských cyklů může být zadán ještě řetězec znaků obsahující až 4 parametry. V těchto parametrech mohou být definovány proměnné uživatelské texty. Jsou k dispozici také předdefinované parametry:	
	%1	Číslo kanálu
	%2	Číslo bloku, návěští
	%3	Textový index pro alarmy cyklů
	%4	Pomocný parametr alarmu

Poznámka

Texty alarmu musí být konfigurovány v uživatelském rozhraní.

Poznámka

Jestliže se má alarm vypisovat v aktivním jazyce uživatelského rozhraní, potřebuje uživatel informace o tom, který jazyk je momentálně na HMI nastaven. Tento údaj je možné ve výrobním programu a v synchronních akcích zjistit pomocí systémové proměnné $AN_LANGUAGE_ON_HMI (Strana 1320).

Příklad

```
Programový kód                          Komentář
...
N100 SETAL (65000)                      ; nastavení alarmu č. 65000
...
```

Další informace

Reakce na alarm a potvrzování alarmů

Alarmy uživatelských cyklů jsou přiřazeny intervalům čísel, které se liší, pokud jde o reakci na alarm a potvrzování alarmů.

Interval čísel	Reakce na alarm	Potvrzování alarmů
65000 - 65499	Zobrazení, zablokování funkce NC-Start	Reset
65500 - 65999	Zobrazení, zablokování funkce NC-Start (nikoli pro příkazy ASUP, když je MD20194 aktivní)	Reset
66000 - 66999	Zobrazení, zablokování funkce NC-Start, zastavení pohybů po zpracování předem dekódovaných bloků	Reset
67000 - 67999	Zobrazení	Cancel

Interval čísel	Reakce na alarm	Potvrzování alarmů
68000 - 68999	Zobrazení, zablokování funkce NC-Start, okamžité zastavení interpolátoru	Reset
69000 - 69999	Zobrazení, zablokování funkce NC-Start, zastavení na konci následujícího bloku	Reset

4.24.6 Definice surového obrobku (WORKPIECE)

Řídící systém musí znát tvar a velikost surového obrobku, aby jej mohl zobrazovat v rámci grafické simulace. Uživatel má proto možnost definovat surový obrobek prostřednictvím uživatelského rozhraní nebo přímo v rámci NC programu. Definice surového obrobku zůstává zachována i po resetu (konec programu/reset kanálu nebo BAG). V průběhu následujícího rozběhu řídícího systému se automaticky vymaže.

Syntaxe

```
WORKPIECE("<WP>", "<RefP>", "<ZeroOffset>", "<Type>", <Par5>,
<Par6>, ..., <Par12>)
```

Význam

`WORKPIECE(...)`:		Předdefinovaná procedura pro definici surového obrobku		
		Zastavení předběžného zpracování:	ano	
		Samostatný blok:	ano	
Parametry:				
1	`"<WP>"`:	Název obrobku (nepovinné)		
		Datový typ:	STRING	
		Tento údaj je zapotřebí jen tehdy, jestliže v jednom kanálu může být zadáno více obrobků. Pokud není parametr zadán, automaticky se přiřadí „WORKP<n>“, kde <n> je číslo deklarovaného kanálu.		
2	`"<RefP>"`:	Upnutí (nepovinné; jen v případě frézek)		
		Datový typ:	STRING	
		Rozsah hodnot:	"Tisch"	Upnutí na pevném stole
			"A"	Upnutí na kruhové ose A
			"B"	Upnutí na kruhové ose B
			"C"	Upnutí na kruhové ose C
		Předpoklad: Stůl, příp. kruhová osa musí být prostřednictvím odpovídajícího strojního parametru pro upínání obrobků odblokovány (viz Příručka pro uvedení do provozu, SINUMERIK Operate).		

3	"<ZeroOffset>":	Nastavitelné posunutí počátku pro umístění surového obrobku (nelze naprogramovat) Volba nastavitelného posunutí počátku pro umístění surového obrobku je nabízena pouze při zadávání surového obrobku prostřednictvím uživatelského rozhraní. Při přímé definici surového obrobku v rámci výrobního programu se oproti tomu surový obrobek vždy vztahuje na momentálně platné posunutí počátku.		
4	"<Type>":	Tvar surového obrobku		
		Datový typ:	STRING	
		Rozsah hodnot:	"CYLINDER":	Válec
			"PIPE":	Trubka
			"RECTANGLE":	Středově upnutý kvádr
			"BOX":	Kvádr
			"N_CORNER":	N-hran
5 ... 12	<Par5> ... <Par12>:	Parametry pro popis tvaru surového obrobku		
		Datový typ:	REAL	
		Počet potřebných parametrů a jejich význam jsou závislé na příslušném tvaru surového obrobku a na hodnotě bitových parametrů. Viz: • Tabulka „Parametry pro popis tvaru surového obrobku“ • Tabulka „Bitové parametry“		
WORPIECE():		Voláním příkazu WORKPIECE bez parametrů se vymažou všechny definice surového obrobku.		
WORPIECE(<WP>):		Voláním příkazu WORKPIECE s názvem obrobku se vymaže jen tato definice surového obrobku.		

Tabulka 4-6 Parametry pro popis tvaru surového obrobku

Tvar surového obrobku	**Parametry**							
	<Par5>	**<Par6>**	**<Par7>**	**<Par8>**	**<Par9>**	**<Par10>**	**<Par11>**	**<Par12>**
Válec	Bitový parametr Reálná hodnota, která je interpretována jako bitově kódované číslo typu Integer. Tyto bity definují význam následujících parametrů (viz tabulka „Bitové parametry").	Vztažný bod Z_0	Délka Z_1	Rozměr pro opracování Z_B	Vnější průměr d_0	-	Otočení okolo kruhové osy	-
Trubka		Vztažný bod Z_0	Délka Z_1	Rozměr pro opracování Z_B	Vnější průměr d_0	Tloušťka stěny (ink) / vnější průměr d_1 (abs)	Otočení okolo kruhové osy	-
Středově upnutý kvádr		Vztažný bod Z_0	Délka Z_1	Rozměr pro opracování Z_B	Šířka W	Délka L	Otočení okolo kruhové osy	-
Kvádr		Vztažný bod Z_0	Délka Z_1	Rozměr pro opracování Z_B	X_0	Y_0	X_1	Y_1
N-hran		Vztažný bod Z_0	Délka Z_1	Rozměr pro opracování Z_B	Počet hran	Šířka klíče	Otočení okolo kruhové osy	-

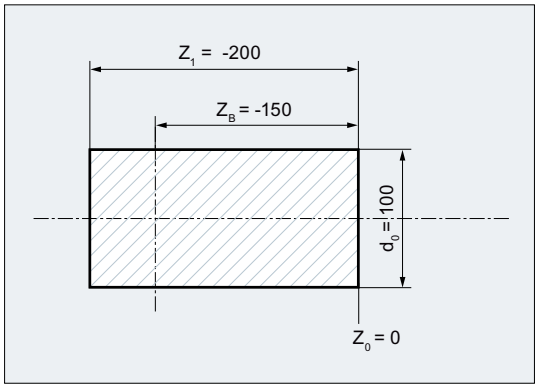
Tabulka 4-7 Bitový parametr

Bit	**Význam**	
4 (0x0010)	Kvádr: X_1	
	= 0	ink
	= 1	abs
5 (0x0020)	Kvádr: Y_1	
	= 0	ink
	= 1	abs
6 (0x0040)	Délka Z_1 (konečný rozměr)	
	= 0	ink
	= 1	abs
Bit 7 (0x0080)	Rozměr pro opracování Z_B	
	= 0	ink
	= 1	abs
Bit 8 (0x0100)	Trubka: Tloušťka stěny/vnitřní průměr	
	= 0	ink
	= 1	abs
9 (0x0200)	N-hran	
	= 0	Šířka klíče
	= 1	Délka hrany

Bit	Význam	
12 (0x1000)	Upnutí pro soustruhy	
	= 0	Hlavní vřeteno
	= 1	Protivřeteno
13 (0x2000)	Protivřeteno	
	= 0	se zrcadlovým převrácením
	= 1	bez zrcadlového převrácení

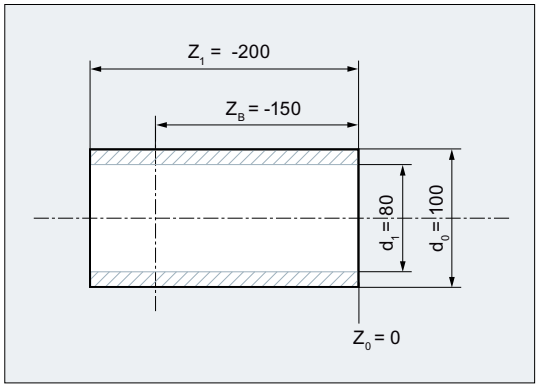
Příklady

Příklad 1: Válcový surový obrobek na soustruhu



Programový kód	Komentář
...	
WORKPIECE(,,,"CYLINDER",0,0,-200,-150,100)	; **definice surového obrobku**
	; tvar surového obrobku: Válec
	; bitový parametr=0 (není nastaven žádný bit) → hodnoty pro délku a rozměr pro opracování jsou inkrementální, surový obrobek v hlavním vřetenu
	; vztažný bod (Z0)=0
	; délka e(Z1)=-200
	; rozměr pro opracování (ZB)=-150
	; vnější průměr (d0)=100
...	

Příklad 2: Trubkový surový obrobek na soustruhu



Programový kód	Komentář
...	
WORKPIECE(,,,"PIPE",256,0,-200,-150,100,80)	; **definice surového obrobku** ; tvar surového obrobku: Trubka ; bitový parametr=256(Bit8=1) → vnitřní průměr je absolutní, délka a rozměr pro opracování jsou inkrementální, surový obrobek v hlavním vřetenu ; vztažný bod (Z0)=0 ; délka e(Z1)=-200 ; rozměr pro opracování (ZB)=-150 ; vnější průměr (d0)=100 ; vnitřní průměr (d1)=80
...	

4.24.7 Přepnutí režimu jazyka (G290, G291)

Řídící systém nabízí možnost načítat a zpracovávat výrobní programy z externích CNC systémů. Předpokladem je, že při uvádění systému do provozu byl definován odpovídající režim jazyka NC systému (dialekt ISO).

Režim dialektu ISO může být aktivován pro každý kanál odděleně. Například kanál 1 může pracovat v režimu dialektu ISO, zatímco v kanálu 2 je aktivní režim systému SINUMERIK.

Přepnutí mezi režimem systému SINUMERIK a režimem dialektu ISO se uskutečňuje v NC programu pomocí příkazů G-funkcí ze skupiny 47. Aktivní nástroj, korekční parametry nástroje a posunutí počátku (nuly) nejsou přepnutím nijak ovlivněny.

Syntaxe

```
G291
...
G290
```

Význam

G290:	Aktivování režimu jazyka systému SINUMERIK	
	Samostatný blok:	ano
	Platnost:	modální
G291:	Aktivování režimu jazyka ISO	
	Samostatný blok:	ano
	Platnost:	modální

Podmínky

Režim systému SINUMERIK

- Předdefinované nastavení příkazů G-funkcí může být pro každý kanál definováno pomocí strojního parametru.
- V režimu systému SINUMERIK nemohou být naprogramovány žádné příkazy jazyka z dialektu ISO.

Režim dialektu ISO

- Režim dialektu ISO jako základní nastavení řídícího systému může být nastaven pomocí strojních parametrů. Řídící systém se potom standardně spouští v režimu dialektu ISO.
- Mohou být programovány pouze příkazy G-funkcí z dialektu ISO. Programování funkcí systému SINUMERIK není v režimu dialektu ISO možné.
- Směšování dialektu ISO a jazyka systému SINUMERIK v jednom NC bloku není možné.
- Přepínání prostřednictvím příkazů G-funkce mezi dialektem ISO M (frézování) a dialektem ISO T (soustružení) není možné.
- Mohou být vyvolávány podprogramy, které jsou naprogramovány v režimu systému SINUMERIK.
- Jestliže se mají používat funkce systému SINUMERIK, je nutné napřed přepnout do režimu systému SINUMERIK (viz příklad).

Příklad

Komprese lineárních bloků v režimu dialektu ISO

Programový kód	**Komentář**
N5 G290	; aktivování režimu jazyka systému SINUMERIK
N10 COMPCAD	; COMPCAD je příkaz jazyka firmy Siemens, který aktivuje funkci kompresoru, která po sobě následující lineární bloky nahrazuje bloky polynomů s co možno nejdelšími úseky dráhy.
N15 G291	; aktivování režimu jazyka ISO

Programový kód	**Komentář**
N20 G01 X100 Y100 F1000	; Protože byl příkaz COMPCAD aktivován v režimu systému SINUMERIK, mohou být pomocí této funkce komprimovány také lineární bloky v režimu dialektu ISO.
...	

Další informace

Příručka k funkcím, Dialekty ISO

4.25 Vlastní programy pro oddělování třísky

4.25.1 Podporované funkce pro oddělování třísky

Pro oddělování třísky jsou Vám nabízeny již hotové obráběcí cykly. Kromě toho máte možnost pomocí funkcí uvedených v následujících odstavcích sestavovat své vlastní programy pro obrábění:

- Sestavování kontury (CONTPRON)
- Sestavování kódované tabulky kontury (CONTPRON)
- Ukončení přípravy kontury (EXECUTE)
- Zjištění průsečíku mezi dvěma konturovými prvky (INTERSEC)
  (jen pro tabulky, které byly vytvořeny pomocí funkce CONTPRON.)
- Zpracovávání konturových prvků v tabulce blok po bloku (EXECTAB)
  (jen pro tabulky, které byly vytvořeny pomocí funkce CONTPRON.)
- Výpočet parametrů kruhu (CALCDAT)

Poznámka

Tyto funkce můžete používat nejen pro oddělování třísky, jsou použitelné univerzálně.

Předpoklady

Před vyvoláváním funkcí CONTPRON nebo CONTDCON musíte:

- Najet na počáteční bod, který umožňuje bezkolizní obrábění.
- Deaktivovat korekci rádiusu nástroje pomocí příkazu `G40`.

4.25.2 Sestavování kontury (CONTPRON)

Pomocí příkazu `CONTPRON` aktivujete funkci pro přípravu kontury. Následně vyvolávané NC-bloky nejsou zpracovávány, ale jsou rozděleny do jednotlivých pohybů a uloženy do tabulky kontury. Každému konturovému prvku odpovídá řádek ve dvourozměrném poli tabulky kontury. Systém uvádí počet zjištěných podříznutí.

Syntaxe

Aktivování přípravy kontury:
`CONTPRON(<tabulka kontury>,<druh opracování>,<podříznutí>, <směr opracování>)`

Deaktivování přípravy kontury a návrat zpět do normálního režimu zpracování:
`EXECUTE(<FEHLER>)`

Viz "Ukončení přípravy kontury (EXECUTE) (Strana 1036)".

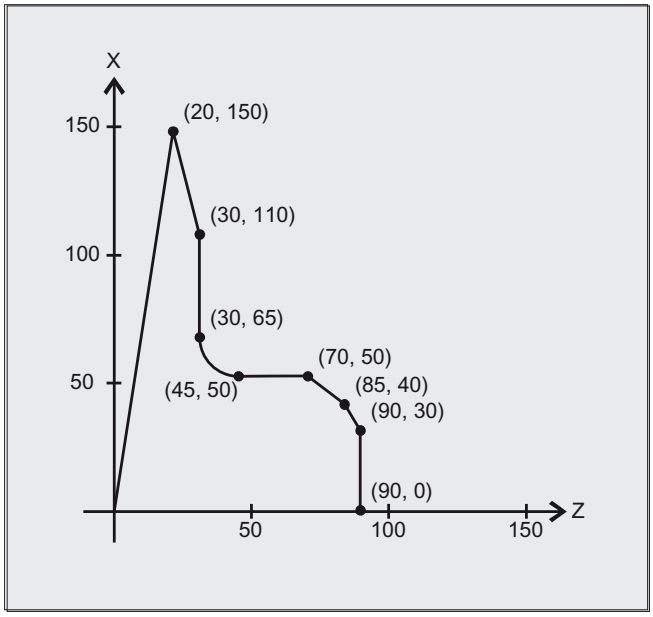
Význam

`CONTPRON`:	Předdefinovaná procedura pro aktivování přípravy kontury za účelem sestavení tabulky kontury		
`<tabulka kontury>`:	Název tabulky kontury		
`<druh opracování>`:	Parametr pro stanovení druhu opracování		
	Typ:	CHAR	
	Hodnota:	`"G"`:	podélné soustružení: vnitřní opracování
		`"L"`:	podélné soustružení: vnější opracování
		`"N"`:	příčné soustružení: vnitřní opracování
		`"P"`:	příčné soustružení: vnější opracování
`<podříznutí>`:	Varianta výsledku pro počet zjištěných prvků podříznutí		
	Typ:	INT	
`<směr opracování>`:	Parametr pro stanovení směru opracování		
	Typ:	INT	
	Hodnota:	0	Ve směru přípravy kontury (standardní hodnota)
		1	V obou směrech vzhledem ke směru přípravy kontury

Příklad 1

Sestavení tabulky kontury s následujícími parametry:

- Název "KTAB"
- max. 30 konturových prvků (kruhy, přímky)
- Proměnná pro počet zjištěných prvků podříznutí
- Proměnní pro chybová hlášení



NC program:

Programový kód	**Komentář**
N10 DEF REAL KTAB[30,11]	; tabulka kontury s názvem KTAB a s max. 30 konturovými prvky, hodnota parametru 11 (počet sloupců v tabulce) je pevná hodnota
N20 DEF INT ANZHINT	; proměnná pro počet prvků podříznutí s názvem ANZHINT
N30 DEF INT FEHLER	; proměnná pro zpětná chybová hlášení (0 = žádná chyba, 1 = chyba)
N40 G18	
N50 CONTPRON(KTAB,"G",ANZHINT)	; aktivování přípravy kontury
N60 G1 X150 Z20	; N60 až N120: popis kontury
N70 X110 Z30	
N80 X50 RND=15	
N90 Z70	
N100 X40 Z85	
N110 X30 Z90	
N120 X0	
N130 EXECUTE(FEHLER)	; ukončení plnění tabulky kontury, přepnutí do normálního programového režimu
N140 …	; další zpracovávání tabulky

Tabulka kontury KTAB:

Index Řádek	Sloupec									
(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

Vysvětlení obsahu sloupců:

(0) Ukazatel na nejbližší konturový prvek (na číslo jeho řádku)

(1) Ukazatel na předcházející konturový prvek

(2) Kódování režimu kontury pro pohyb

Možné hodnoty pro 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) Počáteční bod konturového prvku

(3) = abscisa, (4) = ordináta v aktuální rovině

(5), (6) Koncový bod konturového prvku

(5) = abscisa, (6) = ordináta v aktuální rovině

(7) Ukazatel max./min.: označuje lokální maxima a minima na kontuře

(8) Maximální hodnota mezi konturovým prvkem a abscisou)při podélném obrábění), příp. ordinátou (při příčném obrábění). Úhel je závislý na naprogramovaném druhu opracování.

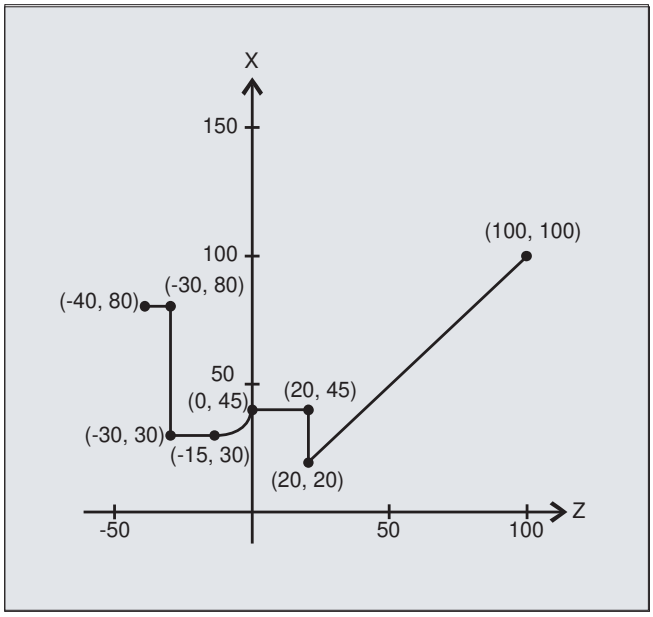
(9), (10) Souřadnice středu konturového prvku, jestliže se jedná o kruhový blok.

(9) = abscisa, (10) = ordináta

Příklad 2

Sestavení tabulky kontury s následujícími parametry:

- Název KTAB
- max. 92 konturových prvků (kruhy, přímky)
- Provozní režim: podélné obrábění, vnější obrábění
- Příprava směrem dopředu a dozadu



NC program:

Programový kód	**Komentář**
N10 DEF REAL KTAB[92,11]	; tabulka kontury s názvem KTAB a s max. 92 konturovými prvky, hodnota parametru 11 je pevná hodnota
N20 DEF CHAR BT="L"	; druh opracování CONTPRON: podélné obrábění, vnější obrábění
N30 DEF INT HE=0	; počet prvků podříznutí = 0
N40 DEF INT MODE=1	; příprava směrem dopředu a dozadu
N50 DEF INT ERR=0	; chybové hlášení
...	
N100 G18 X100 Z100 F1000	
N105 CONTPRON(KTAB,BT,HE,MODE)	; aktivování přípravy kontury
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)	; ukončení plnění tabulky kontury, přepnutí do normálního programového režimu
...	

Tabulka kontury KTAB:

Po ukončení přípravy kontury je kontura k dispozici v obou směrech.

Index	Sloupec										
Řádek	(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

Vysvětlení obsahu sloupců a poznámky k řádkům 0, 1, 6, 8, 83, 85 a 91

Pokud jde o obsah sloupců, platí vysvětlení uvedená u příkladu 1.

Vždy v řádku tabulky 0:

1) Předcházející blok: Řádek n obsahuje konec tabulky kontury směrem dopředu

2) Následující blok: Řádek n je koncem tabulky kontury směrem dopředu

Pokaždé v rámci prvku kontury směrem dopředu:

3) Předcházející blok: začátek kontury (směrem dopředu)

4) Následující blok: konec kontury (směrem dopředu)

Vždy na řádku konec tabulky kontury (směrem dopředu) + 1:

5) Předcházející blok: Počet podříznutí směrem dopředu

6) Následující blok: Počet podříznutí směrem dozadu

Pokaždé v rámci prvku kontury směrem dozadu:

7) Následující blok: konec kontury (směrem dozadu)

8) Předcházející blok: začátek kontury (směrem dozadu)

Vždy na posledním řádku tabulky:

9) Předcházející blok: Řádek n je počátek tabulky kontury (směrem dozadu)

10) Následující blok: Řádek n obsahuje počátek kontury (směrem dozadu)

Další informace

Povolené příkazy pohybu, souřadný systém

Pro programování kontury jsou přípustné následující příkazy G-funkcí:

- G-skupina 1: `G0`, `G1`, `G2`, `G3`

Kromě toho jsou k dispozici následující možnosti:

- Zaoblení a faseta
- Programování kruhových oblouků pomocí příkazů `CIP` a `CT`

Funkce pro spliny, polynomy a závity mají za následek chybu.

Změny souřadného systému prostřednictvím aktivování framu jsou v úseku mezi příkazy `CONTPRON` a `EXECUTE` nepřípustné. Totéž platí i v případě přepnutí mezi funkcemi `G70` a `G71`, příp. `G700` a `G710`.

Výměna geometrických os pomocí příkazu `GEOAX` v době, kdy probíhá příprava tabulky kontury má za následek alarm.

Prvky podříznutí

Popis kontury jednotlivých prvků podříznutí se může dle Vašeho přání uskutečnit buď v podprogramu nebo v jednotlivých blocích.

Oddělování třísky nezávisle na naprogramovaném směru kontury

Příprava kontury pomocí funkce `CONTPRON` byla rozšířena tak, aby po svém vyvolání byla tabulka kontury k dispozici nezávisle na směru, ve kterém byla naprogramována.

4.25.3 Sestavování kódované tabulky kontury (CONTPRON)

Při přípravě kontury aktivované příkazem `CONTDCON` jsou následně vyvolávané NC-bloky ukládány kódovaně do tabulky se 6 sloupci, což je výhodné z hlediska využití paměti. Každému konturovému prvku odpovídá řádek v tabulce kontury. Ze znalosti níže uvedených pravidel pro kódování můžete např. z řádků tabulky sestavit program v kódu DIN pro cyklus. V řádku tabulky s číslem 0 jsou uložena data výchozího bodu.

Syntaxe

Aktivování přípravy kontury:
`CONTDCON(<tabulka kontury>,<směr opracování>)`

Deaktivování přípravy kontury a návrat zpět do normálního režimu zpracování:
`EXECUTE(<FEHLER>)`

Viz "Ukončení přípravy kontury (EXECUTE) (Strana 1036)".

Význam

CONTDCON:	Předdefinovaná procedura pro aktivování přípravy kontury za účelem sestavení kódované tabulky kontury		
<tabulka kontury>:	Název tabulky kontury		
<směr opracování>:	Parametr pro stanovení směru opracování		
	Typ:	INT	
	Hodnota:	0	Příprava kontury podle posloupnosti konturových bloků (standardní hodnota)
		1	nepřípustné

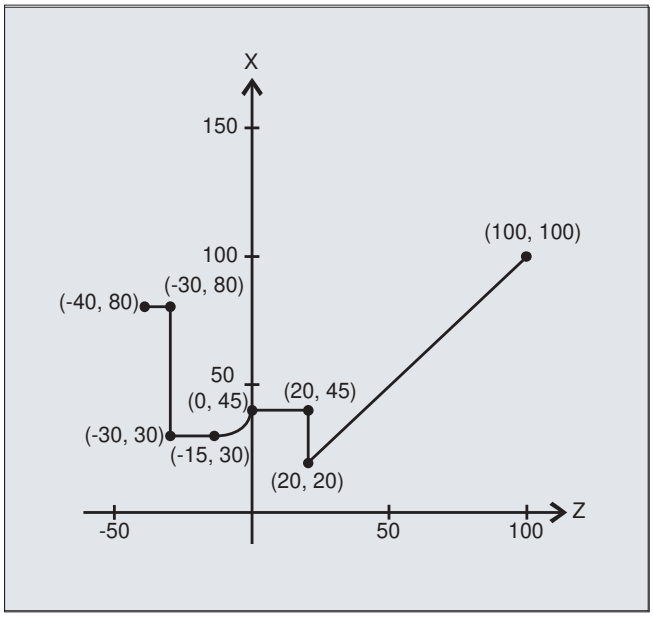
Poznámka

Spektrum příkazů G-funkcí, které jsou přípustné pro funkci CONTDCON v úseku programu, který se má uložit do tabulky, je širší než v případě příkazu CONTPRON. Kromě toho jsou s každým úsekem kontury ukládány také posuvy a typ posuvu.

Příklad:

Sestavení tabulky kontury s následujícími parametry:

- Název "KTAB"
- Konturové prvky (kruhy, přímky)
- Provozní režim: Soustružení
- Směr obrábění: Směrem dopředu



NC program:

Programový kód	**Komentář**
`N10 DEF REAL KTAB[9,6]`	`; Tabulka kontury s názvem KTAB a s 9 řádky, které umožňují 8 konturových bloků. Hodnota parametru 6 (počet sloupců v tabulce) je pevně danou veličinou.`
`N20 DEF INT MODE = 0`	`; proměnná pro stanovení směru opracování Standardní hodnota 0: jen v naprogramovaném směru kontury.`
`N30 DEF INT ERROR = 0`	`; proměnná pro chybové hlášení`
`...`	
`N100 G18 G64 G90 G94 G710`	
`N101 G1 Z100 X100 F1000`	
`N105 CONTDCON (KTAB, MODE)`	`; vyvolání přípravy tabulky (parametr MODE lze vypustit)`
`N110 G1 Z20 X20 F200`	`; popis kontury`
`N120 G9 X45 F300`	
`N130 Z0 F400`	
`N140 G2 Z-15 X30 K=AC(-15) I=AC(45)F100`	
`N150 G64 Z-30 F600`	
`N160 X80 F700`	
`N170 Z-40 F800`	
`N180 EXECUTE(ERROR)`	`; ukončení plnění tabulky kontury, přepnutí do normálního programového režimu`
`...`	

Tabulka kontury KTAB:

	Index sloupce					
	0	1	2	3	4	5
Index řádku	Režim kontury	Koncový bod abscisa	Koncový bod ordináta	Střed na abscise	Střed na ordinátě	Posuv
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

Vysvětlení obsahu sloupců:

Řádek 0: Kódování pro **počáteční bod**

Sloupec 0:	10^0 (místo jednotek): G0 = 0
	10^1 (místo desítek): G70 = 0, G71 = 1, G700 = 2, G710 = 3
Sloupec 1:	Počáteční bod, abscisa
Sloupec 2:	Počáteční bod, ordináta
Sloupec 3-4:	0
Sloupec 5:	Index řádku posledního úseku kontury v tabulce

Řádky 1-n: Údaje reprezentující **úseky kontury**

Sloupec 0:	10^0 (místo jednotek): G0 = 0, G1 = 1, G2 = 2, G3 = 3
	10^1 (místo desítek): G70 = 0, G71 = 1, G700 = 2, G710 = 3
	10^2 (místo stovek): G90 = 0, G91 = 1
	10^3 (místo tisíců): G93 = 0, G94 = 1, G95 = 2, G96 = 3
	10^4 (místo desítek tisíců): G60 = 0, G44 = 1, G641 = 2, G642 = 3
	10^5 (místo stovek tisíců): G9 = 1
Sloupec 1:	Koncový bod, abscisa
Sloupec 2:	Koncový bod, ordináta
Sloupec 3:	Střed kruhu na abscise při kruhové interpolaci
Sloupec 4:	Střed kruhu na ordinátě při kruhové interpolaci
Sloupec 5:	Posuv

Další informace

Povolené příkazy pohybu, souřadný systém

Pro programování kontury jsou přípustné následující G-skupiny a následující příkazy G-funkcí:

G-skupina 1:	`G0`, `G1`, `G2`, `G3`
G-skupina 10:	`G60`, `G64`, `G641`, `G642`
G-skupina 11:	`G9`
G-skupina 13:	`G70`, `G71`, `G700`, `G710`
G-skupina 14:	`G90`, `G91`
G-skupina 15:	`G93`, `G94`, `G95`, `G96`, `G961`

Kromě toho jsou k dispozici následující možnosti:

- Zaoblení a faseta
- Programování kruhových oblouků pomocí příkazů `CIP` a `CT`

Funkce pro spliny, polynomy a závity mají za následek chybu.

Změny souřadného systému prostřednictvím aktivování framu jsou v úseku mezi příkazy `CONTDCON` a `EXECUTE` nepřípustné. Totéž platí i v případě přepnutí mezi funkcemi `G70` a `G71`, příp. `G700` a `G710`.

Výměna geometrických os pomocí příkazu GEOAX v době, kdy probíhá příprava tabulky kontury má za následek alarm.

Směr obrábění

Při opracovávání pomocí tabulky kontury sestavené pomocí funkce CONTDCON se smí postupovat ve směru naprogramování kontury.

4.25.4 Zjištění průsečíku mezi dvěma konturovými prvky (INTERSEC)

Funkce INTERSEC zjišťuje dvou normovaných konturových prvků z tabulek kontury sestavených pomocí funkce CONTPRON.

Syntaxe

```
<Status>=INTERSEC(<tabulka kontury_1>[<prvek kontury_1>],
<tabulka kontury_2>[<prvek kontury_2>],<průsečík>,<druh opracování>)
```

Význam

INTERSEC:	Předem definovaná funkce pro výpočet průsečíku dvou konturových prvků z tabulek kontury vytvořených pomocí funkce CONTPRON.		
<Status>:	Proměnná pro stav průsečíku		
	Typ:	BOOL	
	Hodnota:	TRUE	Průsečík nalezen
		FALSE	Žádný průsečík nenalezen
<tabulka kontury_1>:	Název první tabulky kontury		
<prvek kontury_1>:	Číslo konturového prvku z první tabulky		
<tabulka kontury_2>:	Název druhé tabulky kontury		
<prvek kontury_2>:	Číslo konturového prvku ze druhé tabulky		
<průsečík>:	Souřadnice průsečíku v aktivní rovině (G17 / G18 / G19)		
	Typ:	REAL	
<druh opracování>:	Parametr pro stanovení druhu opracování		
	Typ:	INT	
	Hodnota:	0	Výpočet průsečíku v aktivní rovině dané parametrem 2 (standardní hodnota)
		1	Výpočet průsečíku nezávisle na předávané rovině

Poznámka

Mějte prosím na paměti, že proměnné musí být před svým použitím definovány.

Předávání kontur vyžaduje, aby byly dodrženy hodnoty definované příkazem CONTPRON.

Parametr	Význam
2	Kódování režimu kontury pro pohyb
3	Počáteční bod kontury, abscisa
4	Počáteční bod kontury, ordináta
5	Koncový bod kontury, abscisa
6	Koncový bod kontury, ordináta
9	Souřadnice středu pro abscisu (jen v případě kruhové kontury)
10	Souřadnice středu pro ordinátu (jen v případě kruhové kontury)

Příklad:

Zjištění průsečíku konturového prvku 3 z tabulky TABNAME1 a konturového prvku 7 z tabulky TABNAME2. Souřadnice průsečíku v aktivní rovině jsou uloženy v proměnné ISCOORD (1. prvek = abscisa, 2. prvek = ordináta). Jestliže neexistuje žádný průsečík, uskuteční se skok na návěští KEINSCH (žádný průsečík nenalezen).

```
Programový kód                                          Komentář
DEF REAL TABNAME1[12,11]                                ; tabulka kontury 1
DEF REAL TABNAME2[10,11]                                ; tabulka kontury 2
DEF REAL ISCOORD[2]                                     ; proměnná pro souřadnice průsečíku
DEF BOOL ISPOINT                                        ; proměnná pro stav průsečíku
DEF INT MODE                                            ; proměnná pro druh obrábění
…
MODE=1                                                  ; výpočet nezávisle na aktivní rovině
N10 ISPOINT=INTERSEC(TABNAME1[3],TABNAME2[7],           ; volání průsečíku konturových prvků
ISCOORD,MODE)
N20 IF ISPOINT==FALSE GOTOF KEINSCH                     ; skok na návěští KEINSCH
…
```

4.25.5 Pohyb po konturových prvcích v tabulce blok po bloku (EXECTAB)

Pomocí příkazu EXECTAB můžete uskutečnit pohyb blok po bloku po konturových prvcích tabulky, která byla vytvořena například příkazem CONTPRON.

Syntaxe

```
EXECTAB(<tabulka kontury>[<konturový prvek>])
```

Význam

EXECTAB:	Předdefinovaná procedura pro aktivování pohybu po konturových prvcích
<tabulka kontury>:	Název tabulky kontury
<konturový prvek>:	Číslo konturového prvku

Příklad:

Má být uskutečněn pohyb blok po bloku po konturových prvcích 0 až 2 z tabulky KTAB.

Programový kód	**Komentář**
N10 EXECTAB(KTAB[0])	; pohyb po prvku 0 z tabulky kontury KTAB
N20 EXECTAB(KTAB[1])	; pohyb po prvku 1 z tabulky kontury KTAB
N30 EXECTAB(KTAB[2])	; pohyb po prvku 2 z tabulky kontury KTAB

4.25.6 Výpočet parametrů kruhu (CALCDAT)

Pomocí příkazu CALCDAT můžete ze tří nebo čtyř známých bodů na kružnici vypočítat její rádius a souřadnice jejího středu. Zadávané body musí být různé.

V případě 4 bodů, které neleží na kružnici úplně přesně, se pro střed a rádius kružnice vypočítá střední hodnota.

Poznámka

Matematický předpis pro výpočet střední hodnoty

Výpočet kruhu se uskuteční 4x:

1. s body na kružnici 1, 2, 3
2. s body na kružnici 1, 2, 4
3. s body na kružnici 1, 3, 4
4. s body na kružnici 2, 3, 4

Hodnota abscisy a hodnota ordináty pro souřadnice středu kruhu se vypočítají tak, že se hodnoty abscisy, příp. hodnoty ordináty, ze čtyř výše uvedených výpočtů kruhu sečtou a podělí 4.

Rádius se vypočítá tak, že se určí odmocnina se součtu čtyř rádiusů dříve vypočítaných kruhů a výsledek se vynásobí 0,5.

Syntaxe

```
<Status>=CALCDAT(<body kružnice>[<počet>,<druh>],<počet>,<výsledek>)
```

Význam

CALCDAT:	Předdefinovaná funkce pro výpočet rádiusu a souřadnic středu kružnice na základě 3 nebo 4 bodů.		
<Status>:	Proměnná pro stav výpočtu kružnice		
	Typ:	BOOL	
	Hodnota:	TRUE	Zadané body leží na kružnici.
		FALSE	Zadané body **neleží** na kružnici.

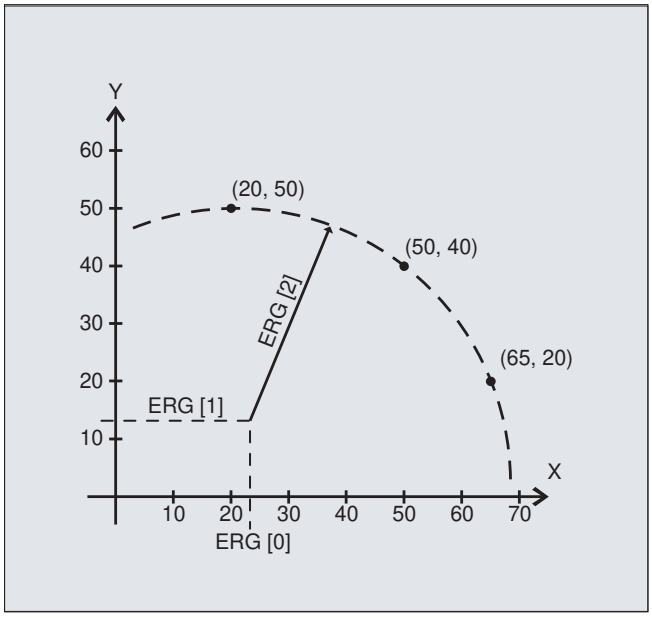
<body kružnice>[]:	Proměnná pro zadání bodů na kružnici s následujícími parametry:	
	<počet>:	Počet bodů na kružnici (3 nebo 4)
	<druh>:	Druh zadání souřadnic, např. 2 pro souřadnice 2 bodů
<počet>:	Parametr pro zadání počtu bodů použitých pro výpočet (3 nebo 4)	
<výsledek>[3]:	Proměnná pro výsledek Zadání souřadnic středu kružnice a rádiusu	
	0	Souřadnice středu kružnice: hodnota abscisy
	1	Souřadnice středu kružnice: hodnota ordináty
	2	Rádius

Poznámka

Mějte prosím na paměti, že proměnné musí být před svým použitím definovány.

Příklad:

U tří bodů se má zjistit, zda leží na kruhovém oblouku.



Programový kód	**Komentář**
N10 DEF REAL PKT[3,2]=(20,50,50,40,65,20)	; proměnná pro zadání bodů na kružnici
N20 DEF REAL ERG[3]	; proměnná pro výsledek
N30 DEF BOOL STATUS	; proměnná pro status
N40 STATUS=CALCDAT(PKT,3,ERG)	; vyvolání zjištěných parametrů kružnice
N50 IF STATUS == FALSE GOTOF ERROR	; skok v případě chyby

4.25.7 Ukončení přípravy kontury (EXECUTE)

Pomocí příkazu `EXECUTE` se příprava kontury ukončí a současně se přepne zpátky do normálního režimu zpracovávání programu.

Syntaxe

```
EXECUTE(<FEHLER>)
```

Význam

`EXECUTE`:	Předdefinovaná procedura pro ukončení přípravy kontury	
`<FEHLER>`:	Proměnná pro zpětné chybové hlášení	
	Typ:	INT
	Hodnota proměnné ukazuje, zda bylo možné zpracovat konturu bezchybně.	
	0	Chyba
	1	žádná chyba

Příklad:

Programový kód

```
...
N30 CONTPRON(...)
N40 G1 X... Z...
...
N100 EXECUTE(...)
...
```

4.26 Programování externích cyklů

4.26.1 Technologické cykly

4.26.1.1 Úvod

Obsah

Tyto kapitola obsahuje popis cyklů pro technologie soustružení, frézování a broušení.

Struktura

Popis cyklu má následující strukturu:

- **Syntaxe**
  Název cyklu a posloupnost vyvolávání předávaných parametrů
- **Parametry**
  Tabulka s vysvětleními jednotlivých parametrů

Popis parametrů

V tabulce jsou k jednotlivým parametrům uváděny následující údaje: Název, popis, rozsah hodnot a souvislosti s jinými parametry.

Sloupec s odkazem na parametr na obrazovce slouží pro usnadnění opětovného hledání hodnot naprogramovaných na řídícím systému při zpětném překladu externě generovaného volání cyklu.

Parametry "jen pro pracovní plochu"

V tabulce jsou některé parametry označeny příznakem "jen pro pracovní plochu". Tyto parametry nemají pro fungování cyklu žádný význam. Jsou zapotřebí jen k tomu, aby volání cyklu bylo možné dokonale zpětně přeložit. Pokud nejsou naprogramovány, může být cyklus i přesto zpětně přeložen, pole jsou ale potom odpovídajícím způsobem barevně označena a musí být vyplněna na obrazovce.

Parametry "rezervováno"

Pro parametry, které jsou popsány jako "rezervované", musí být naprogramována hodnota 0 nebo musí zůstat prázdné, aby přiřazení následujících parametrů volání odpovídalo interním parametrům cyklu. Výjimka: U parametrů typu STRING hodnota "" nebo prázdný znak.

Opakování cyklu na polohovacím vzoru

Vrtací a frézovací cykly mohou být opakovány na polohovacím vzoru (modální volání). Před cyklem je potom na tomtéž řádku zapotřebí napsat příkaz `MCALL`, např. `MCALL CYCLE83(...)`.

Poznámka

Pokud jsou určité předávané parametry (např. `<_VARI>`, `<_GMODE>`, `<_DMODE>`, `<_AMODE>`) naprogramovány nepřímo jako parametry, při zpětném překladu se otevře vstupní obrazovka, tu však není možné uložit, protože pro některá pole s volitelnou hodnotou neexistuje žádné jednoznačné přiřazení.

4.26.1.2 Přehled pro jednotlivé technologie

V následující tabulce se nachází přehled všech externě naprogramovatelných technologických cyklů, které jsou k dispozici a které jsou přiřazeny jednotlivým technologiím:

Technologie	Technologický cyklus
Vrtání	• CYCLE81 - Vrtání, navrtávání středicích důlků (Strana 1078) • CYCLE82 - Vrtání, čelní zahlubování (Strana 1079) • CYCLE85 - Vystružování (Strana 1088) • CYCLE86 - Vyvrtávání (Strana 1089) • CYCLE83 - Vrtání hlubokých děr 1 (Strana 1082) • CYCLE830 - Vrtání hlubokých děr 2 (Strana 1117) • CYCLE84 - Vrtání závitů bez vyrovnávací hlavičky (Strana 1085) • CYCLE840 - Vrtání závitů s vyrovnávací hlavičkou (Strana 1127) • CYCLE78 - Frézování vrtaných závitů (Strana 1074) • CYCLE802 - Libovolné pozice (Strana 1113) • HOLES1 - Polohovací vzor Přímka (Strana 1039) • CYCLE801 - Polohovací vzor Mřížka nebo Obdélník (Strana 1111) • HOLES2 - Polohovací vzor Kružnice nebo Kruhový oblouk (Strana 1040)
Soustružení	• CYCLE951 - Oddělování třísky (Strana 1137) • CYCLE930 - Zápich (Strana 1132) • CYCLE940 - Odlehčovací zápich tvaru E a F / závitový zápich (Strana 1134) • CYCLE99 - Soustružení závitu (Strana 1098) • CYCLE98 - Řetězec závitů (Strana 1094) • CYCLE92 - Upichování (Strana 1091)
Soustružení kontur	• CYCLE62 - volání kontury (Strana 1059) • CYCLE952 - Oddělování třísky / odstraňování zbytkového materiálu oddělováním třísky / zapichování / zapichování zbytkového materiálu / soustružení zápichu / odstraňování zbytkového materiálu soustružením zápichu (Strana 1140)
Frézování	• CYCLE61 - Rovinné frézování (Strana 1057) • POCKET3 - Pravoúhlá kapsa (Strana 1042) • POCKET4 - Kruhová kapsa (Strana 1045) • CYCLE76 - Pravoúhlý čep (Strana 1070) • CYCLE77 - Kruhový čep (Strana 1072) • CYCLE79 - Mnohohran (Strana 1076) • SLOT1 - Podélná drážka (Strana 1048) • SLOT2 - Kruhová drážka (Strana 1050) • CYCLE899 - Otevřená drážka (Strana 1129) • LONGHOLE - Podlouhlá díra (Strana 1053) • CYCLE70 - Frézování závitu (Strana 1064) • CYCLE60 - Gravírování (Strana 1055)

Technologie	Technologický cyklus
Frézování kontury	• CYCLE62 - volání kontury (Strana 1059) • CYCLE72 - Frézování po dráze (Strana 1066) • CYCLE63 - Frézování konturové kapsy / odstranění zbytkového materiálu z konturové kapsy / frézování konturového čepu / odstranění zbytkového materiálu z konturového čepu (Strana 1060) • CYCLE64 - Předvrtání konturové kapsy (Strana 1062)
Broušení	• CYCLE495 - Profilování (Strana 1103) • CYCLE435 - Nastavení souřadného systému orovnávače (Strana 1103) • CYCLE4071 - Podélné broušení s přísuvem v bodu obratu (Strana 1146) • CYCLE4072 - Podélné broušení s přísuvem v bodě obratu a se signálem pro přerušení (Strana 1148) • CYCLE4073 - Podélné broušení se spojitým přísuvem (Strana 1152) • CYCLE4074 - Podélné broušení se spojitým přísuvem a se signálem pro přerušení (Strana 1153) • CYCLE4075 - Plošné broušení s přísuvem v bodě obratu (Strana 1156) • CYCLE4077 - Plošné broušení s přísuvem na bod obratu a se signálem pro přerušení (Strana 1159) • CYCLE4078 - Plošné broušení se spojitým přísuvem (Strana 1163) • CYCLE4079 - Plošné broušení s přerušovaným přísuvem (Strana 1165)
Ostatní	• CYCLE782 - Přizpůsobení zatížení (Strana 1105) • CYCLE800 - Naklápění roviny / naklápění nástroje / nastavování orientace nástroje (Strana 1108) • CYCLE806 - Interpolační soustružení (Strana 1116) • CYCLE832 - Vysokorychlostní obrábění (Strana 1123)
Všechny	• GROUP_BEGIN - Začátek programového bloku (Strana 1167) • GROUP_END - konec programového bloku (Strana 1168) • GROUP_ADDEND - konec doplňkového bloku najíždění (Strana 1168)

4.26.1.3 HOLES1 - Polohovací vzor Přímka

Syntaxe

```
HOLES1(<SPCA>, <SPCO>, <STA1>, <FDIS>, <DBH>, <NUM>, <_VARI>,
<_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	X0	<SPCA>	REAL	Vztažný bod na 1. ose pro řadu děr (abs)		
2	Y0	<SPCO>	REAL	Vztažný bod na 2. ose pro řadu děr (abs)		
3	α0	<STA1>	REAL	Základní úhel otočení (úhel vzhledem k 1. ose)		
4	L0	<FDIS>	REAL	Vzdálenost 1. díry od vztažného bodu		
5	L	<DBH>	REAL	Vzdálenost mezi dírami		
6	N	<NUM>	INT	Počet děr		
7		<_VARI>	INT	rezervováno		
8		<_UMODE>	INT	rezervováno		
9		<_HIDE>	STRING [200]	Přeskakované pozice • max. 198 znaků • Údaj pořadového čísla pozice, např. "1,3" (pozice 1 a 3 se nebudou uskutečňovat)		
10		<_NSP>	INT	rezervováno		
11		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)

4.26.1.4 HOLES2 - Polohovací vzor Kružnice nebo Kruhový oblouk

Syntaxe

```
HOLES2(<CPA>, <CPO>, <RAD>, <STA1>, <INDA>, <NUM>, <_VARI>,
<_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)
```

Parametr

Č.	Obrazovka parametrů	Parametry interní	Datový typ	Význam	
1	X0	<CPA>	REAL	Střed kruhu na 1. ose pro díry na kruhovém oblouku (abs) Souřadnice vztažného bodu na 1. ose	(v případě XY) (v případě XA, YB, ZC)
2	Y0	<CPO>	REAL	Střed kruhu na 2. ose pro díry na kruhovém oblouku (abs) Souřadnice vztažného bodu na 2. ose	(v případě XY) (v případě XA, YB, ZC)
3	R	<RAD>	REAL	Rádius kruhového oblouku	(v případě XY)

Č.	Obrazovka parametrů	Parametry interní	Datový typ	Význam			
4	α0	<STA1>	REAL	Počáteční úhel nebo poloha 1. kruhové osy			(v případě XY) (v případě XA, YB, ZC)
5	α1	<INDA>	REAL	Úhlový krok, kterým jsou drážky od sebe vzdáleny (jen u kruhového oblouku)			(v případě XY, XA, YB, ZC)
					< 0 =	Ve směru hodinových ručiček	
					> 0 =	Proti směru hodinových ručiček	
6	N	<NUM>	INT	Počet pozic			
7		<_VARI>	INT	Způsob opracování			
				JEDNOTKY:	rezervováno		
				DESÍTKY:	Způsob najíždění na pozici		
					0 =	Najíždění na pozici po lineární dráze	
					1 =	Najíždění na pozici po kruhové dráze	
				STOVKY:	rezervováno		
				TISÍCE:	Kruhový vzor		
					0 =	Režim kompatibility, pokud je INDA = 0, pak celá kružnice, INDA <> 0, potom kruhový oblouk	
					1 =	Kružnice	
					2 =	Kruhový oblouk	
				DESÍTKY TISÍC:	Polohovací vzor s kruhovou osou		
					0 =	XY (bez kruhové osy)	(v případě XY)
					1 =	XA (osa X a kruhová osa oko osy X)	(jen v případě XA)
					2 =	YB (osa Y a kruhová osa oko osy Y)	(jen v případě YB)
					3 =	ZC (osa Z a kruhová osa oko osy C)	(jen v případě ZC)
				JEDNOTKY MILIONŮ + STOVKY TISÍC	Offset (pokud se více kruhových otáčí okolo stejné osy; pokud je index příliš veliký, potom 1. osa)		
					00 =	1. osa A, B nebo C	
					01 =	2. osa A, B nebo C	
					...		
					10 =	20. osa A, B nebo C	
8		<_UMODE>	INT	rezervováno			
9		<_HIDE>	STRING [200]	rezervováno			
10		<_NSP>	INT	rezervováno			

Č.	Obrazovka parametrů	Parametry interní	Datový typ	Význam		
11		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)

4.26.1.5 POCKET3 - Pravoúhlá kapsa

Syntaxe

```
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>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	RP	<_RTP>	REAL	Návratová rovina (abs)		
2	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)		
3	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
4	Z1	<_DP>	REAL	Hloubka kapsy (abs/link), viz <_AMODE>)		
5	L	<_LENG>	REAL	Délka kapsy (ink, je potřeba zadávat se znaménkem)		
6	W	<_WID>	REAL	Šířka kapsy (ink, je potřeba zadávat se znaménkem)		
7	R	<_CRAD>	REAL	Rohový rádius kapsy		
8	X0	<_PA>	REAL	Vztažný bod, 1. osa (abs)		
9	YO	<_PO>	REAL	Vztažný bod, 2. osa (abs)		
10	α0	<_STA>	REAL	Úhel otočení, úhel mezi podélnou osou (L) a 1. osou		
11	DZ	<_MID>	REAL	Maximální přísuv do hloubky		
12	UXY	<_FAL>	REAL	Přídavek rozměru pro opracování načisto v rovině		
13	UZ	<_FALD>	REAL	Přídavek rozměru pro opracování načisto na dně		
14	F	<_FFP1>	REAL	Posuv v rovině		
15	FZ	<_FFD>	REAL	Rychlost přísuvu do hloubky		
16		<_CDIR>	INT	Směr frézování:	0 =	Sousledné frézování
					1 =	Nesousledné frézování

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
17		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:		
					1 =	Obrábění nahrubo
					2 =	Obrábění načisto
					4 =	Obrábění okraje načisto
					5 =	Srážení hran
				DESÍTKY:		
					0 =	předvrtáno, přísuv s G0
					1 =	kolmo, přísuv s G1
					2 =	po šroubovici
					3 =	oscilačním (kyvným) pohybem po podélné ose kapsy
				STOVKY:	rezervováno	
18	DXY	<_MIDA>	REAL	Maximální přísuv v rovině, jednotky viz <_AMODE>		
19	L1	<_AP1>	REAL	Délka částečně obrobené kapsy (ink)		
20	W1	<_AP2>	REAL	Šířka částečně obrobené kapsy (ink)		
21	AZ	<_AD>	REAL	Hloubka částečně obrobené kapsy (ink)		
22	ER	<_RAD1>	REAL	Rádius spirální dráhy při zajíždění po šroubovici		
	EW			Maximální úhel zajíždění pro oscilační pohyb		
23	EP	<_DP1>	REAL	Stoupání spirály při zajíždění po spirální dráze		
24		<_UMODE>	INT	rezervováno		
25	FS	<_FS>	REAL	Šířka fasety (ink)		
26	ZFS	<_ZFS>	REAL	Hloubka zajíždění (špička nástroje) při srážení hran (abs/ink), viz <_AMODE>		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
27		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	rezervováno	
				STOVKY:	Volba opracování/pouze výpočet počátečního bodu	
					0 =	Režim kompatibility
					1 =	normální opracování
				TISÍCE:	Zadávání rozměrů pomocí středu/rohu	
					0 =	Režim kompatibility
					1 =	Zadávání rozměrů pomocí středu
					2 =	Kótování pomocí rohového bodu, poloha kapsy +LENG/+WID
					3 =	Kótování pomocí rohového bodu, poloha kapsy -LENG/+WID
					4 =	Kótování pomocí rohového bodu, poloha kapsy +LENG/-WID
					5 =	Kótování pomocí rohového bodu, poloha kapsy -LENG/-WID
				DESÍTKY TISÍC:	Kompletní opracování/dopracování	
					0 =	Režim kompatibility (s parametry <_AP1>, <_AP2> a <_AD> se zachází jako předtím)
					1 =	Kompletní opracování
					2 =	Dodatečné opracování

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
28		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	Druh posuvu: G-skupina (G94/G95) pro posuv v rovině a posuv do hloubky	
					0 =	Režim kompatibility
					1 =	Příkaz G-funkce stejný jako před voláním cyklu G94/G95 pro posuv v rovině a posuv do hloubky zůstávají stejné.
				STOVKY:	---	rezervováno
				TISÍCE:	---	rezervováno
				DESÍTKY TISÍC:	Odstupňování technologie na obrazovkách cyklů (Strana 1169)	
					0 =	Zadání: kompletní
					1 =	Zadání: jednoduché
29		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Hloubka kapsy (Z1)	
					0 =	absolutní (režim kompatibility)
					1 =	inkrementální
				DESÍTKY:	Jednotky pro přísuv v rovině (DXY)	
					0 =	mm
					1 =	procentuální hodnota z průměru nástroje
				STOVKY:	Hloubka zajíždění při srážení hran (ZFS)	
					0 =	absolutní
					1 =	inkrementální

4.26.1.6 POCKET4 - Kruhová kapsa

Syntaxe

```
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>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	RP	<_RTP>	REAL	Návratová rovina (abs)		
2	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)		
3	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
4	Z1	<_DP>	REAL	Hloubka kapsy (abs/ink), viz <_AMODE>		
5	Ø	<_CDIAM>	REAL	Průměr kapsy nebo rádius kapsy, viz <_DMODE>		
6	X0	<_PA>	REAL	Vztažný bod, 1. osa (abs)		
7	Y0	<_PO>	REAL	Vztažný bod, 2. osa (abs)		
8	DZ	<_MID>	REAL	maximální přísuv do hloubky, viz <_VARI> = po rovinách maximální stoupání spirály, viz <_VARI> = po šroubovici		
9	UXY	<_FAL>	REAL	Přídavek rozměru pro opracování načisto v rovině		
10	UZ	<_FALD>	REAL	Přídavek rozměru pro opracování načisto na dně		
11	F	<_FFP1>	REAL	Posuv pro opracovávání plochy		
12	FZ	<_FFD>	REAL	Rychlost přísuvu do hloubky		
13		<_CDIR>	INT	Směr frézování	0 =	Sousledné frézování
					1 =	Nesousledné frézování
14		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:	Opracování	
					1 =	Obrábění nahrubo
					2 =	Obrábění načisto
					4 =	Obrábění okraje načisto
					5 =	Srážení hran
				DESÍTKY:	Způsob přísuvu (obrábění nahrubo a obrábění načisto)	
					0 =	předvrtáno, přísuv s G0 (kapsa je předem opracována)
					1 =	kolmo, přísuv s G1
					2 =	po šroubovici
				STOVKY:	rezervováno	
				TISÍCE:		
					0 =	po rovinách
					1 =	po šroubovici
15	DXY	<_MIDA>	REAL	Maximální přísuv v rovině, viz <_AMODE>, 0 = 0,8 x průměr nástroje		
16	Ø	<_AP1>	REAL	Průměr/rádius částečně obrobené kapsy (ink)		
17	AZ	<_AD>	REAL	Hloubka částečně obrobené kapsy (ink)		
18	ER	<_RAD1>	REAL	Rádius spirální dráhy při zajíždění po šroubovici		
19	EP	<_DP1>	REAL	Stoupání spirály při zajíždění po spirální dráze		
20		<_UMODE>	INT	rezervováno		
21	FS	<_FS>	REAL	Šířka fasety (ink)		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
22	ZFS	<_ZFS>	REAL	Hloubka zajíždění (špička nástroje) při srážení hran (abs/ink), viz <_AMODE>		
23		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	rezervováno	
				STOVKY:	Opracování/výpočet počátečního bodu	
					0 =	Režim kompatibility
					1 =	normální opracování
				TISÍCE:	rezervováno	
				DESÍTKY TISÍC:	Kompletní opracování/dopracování	
					0 =	Režim kompatibility (s parametry <_AP1> a <_AD> se zachází jako předtím)
					1 =	Kompletní opracování
					2 =	Dodatečné opracování
24		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	Druh posuvu: G-skupina (G94/G95) pro posuv v rovině a posuv do hloubky	
					0 =	Režim kompatibility
					1 =	Příkaz G-funkce stejný jako před voláním cyklu G94/G95 pro posuv v rovině a posuv do hloubky zůstávají stejné.
				STOVKY:		
					0 =	Režim kompatibility (<_CDIAM>/<_AP1> zadat jako rádius)
					1 =	<_CDIAM>/<_AP1> se zadávají jako průměr
				TISÍCE:	---	rezervováno
				DESÍTKY TISÍC:	Odstupňování technologie na obrazovkách cyklů (Strana 1169)	
					0 =	Zadání: kompletní
					1 =	Zadání: jednoduché

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
25		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Hloubka kapsy (Z1)	
					0 =	absolutní (režim kompatibility)
					1 =	inkrementální
				DESÍTKY:	Jednotky pro přísuv do šířky (DXY)	
					0 =	mm
					1 =	procentuální hodnota z průměru nástroje
				STOVKY:	Hloubka zajíždění při srážení hran (ZFS)	
					0 =	absolutní
					1 =	inkrementální

4.26.1.7 SLOT1 - Podélná drážka

Syntaxe

```
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>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	RP	<RTP>	REAL	Návratová rovina (abs)
2	Z0	<RFP>	REAL	Vztažný bod osy nástroje (abs)
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)
4	Z1	<_DP>	REAL	Hloubka drážky (abs)
5		<_DPR>	REAL	Hloubka drážky (ink), vztažená na Z0 (zadává se bez znaménka)
6		<NUM>	INT	Počet drážek = 1
7	L	<LENG>	REAL	Délka drážky
8	W	<WID>	REAL	Šířka drážky
9	X0	<_CPA>	REAL	Vztažný bod, 1. osa roviny
10	Y0	<_CPO>	REAL	Vztažný bod, 2. osa roviny
11		<RAD>	REAL	rezervováno
12	α	<STA1>	REAL	Úhel otočení
13		<INDA>	REAL	rezervováno
14	FZ	<FFD>	REAL	Rychlost přísuvu do hloubky
15	F	<FFP1>	REAL	Posuv
16	DZ	<_MID>	REAL	Maximální přísuv do hloubky

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
17		<CDIR>	INT	Směr frézování	0 =	Sousledné frézování
					1 =	Nesousledné frézování
18	UXY	<_FAL>	REAL	Přídavek rozměru pro opracování načisto v rovině nebo na okrajích drážky		
19		<VARI>	INT	Způsob opracování		
				JEDNOTKY:		
					0 =	rezervováno
					1 =	Obrábění nahrubo
					2 =	Obrábění načisto
					4 =	Obrábění načisto okraje (opracování jen okraje)
					5 =	Srážení hran
				DESÍTKY:	Najíždění	
					0 =	předvrtáno, přísuv s G0 (drážka je předem opracována)
					1 =	kolmo, přísuv s G1
					2 =	po šroubovici
					3 =	oscilačním pohybem
				STOVKY:	rezervováno	
20	DZF	<_MIDF>	REAL	rezervováno		
21	FF	<FFP2>	REAL	rezervováno		
22	SF	<SSF>	REAL	rezervováno		
23	UZ	<_FALD>	REAL	Přídavek rozměru pro opracování načisto na dně		
24	ER	<_STA2>	REAL	Rádius spirální dráhy při zajíždění po šroubovici		
	EW			Maximální úhel zajíždění pro oscilační pohyb		
25	EP	<_DP1>	REAL	Hloubka zajíždění na jednu otáčku pro spirální pohyb		
26		<_UMODE>	INT	rezervováno		
27	FS	<_FS>	REAL	Šířka fasety (ink) u srážení hran		
28	ZFS	<_ZFS>	REAL	Hloubka zajíždění (špička nástroje) při srážení hran (abs/ink), viz <_AMODE>		
29		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	rezervováno	
				STOVKY:	Volba opracování nebo pouze výpočet počátečního bodu	
					1 =	normální opracování
				TISÍCE:	Kótování vztažného bodu, poloha drážky	
					0 =	ve středu
					1 =	vlevo uvnitř +L
					2 =	vpravo uvnitř -L
					3 =	levý okraj +L
					4 =	pravý okraj -L

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
30		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	rezervováno	
				STOVKY:	rezervováno	
				TISÍCE:	Identifikace verze SW	
					1 =	Rozšířené funkce SLOT1
				DESÍTKY TISÍC:	Odstupňování technologie na obrazovkách cyklů (Strana 1169)	
					0 =	Zadání: kompletní
					1 =	Zadání: jednoduché
31		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Konečná hloubka Z1 (abs/ink)	
					0 =	Kompatibilita
					1 =	Z1 (ink)
					2 =	Z1 (abs)
				DESÍTKY:	rezervováno	
				STOVKY:	Hloubka zajíždění při srážení hran ZFS	
					0 =	ZFS (abs)
					1 =	ZFS (ink)

Poznámka

Cyklus je oproti předcházejícím verzím SW vybaven novými funkcemi. To má za následek, že určité parametry se na vstupní obrazovce už nevypisují (<NUM>, <RAD>, <INDA>). Větší počet drážek na polohovacím vzoru je možné naprogramovat pomocí příkazu "MCALL" a voláním požadovaného polohovacího vzoru, např. HOLES2.

4.26.1.8 SLOT2 - Kruhová drážka

Syntaxe

```
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>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	RP	<RTP>	REAL	Návratová rovina (abs)		
2	Z0	<RFP>	REAL	Vztažný bod osy nástroje (abs)		
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
4	Z1	<_DP>	REAL	Hloubka drážky (abs)		
5		<_DPR>	REAL	Hloubka drážky (ink), vztažená na Z0 (zadává se bez znaménka)		
6	N	<NUM>	INT	Počet drážek		
7	α1	<AFSL>	REAL	Úhel kruhové výseče dané drážky		
8	W	<WID>	REAL	Šířka drážky		
9	X0	<_CPA>	REAL	Vztažný bod = střed kruhového oblouku, 1. osa roviny		
10	Y0	<_CPO>	REAL	Vztažný bod = střed kruhového oblouku, 2. osa roviny		
11	R	<RAD>	REAL	Rádius kruhového oblouku		
12	α0	<STA1>	REAL	Počáteční úhel		
13	α2	<INDA>	REAL	Úhlový krok		
14	FZ	<FFD>	REAL	Rychlost přísuvu do hloubky		
15	F	<FFP1>	REAL	Posuv		
16	DZ	<_MID>	REAL	Maximální přísuv do hloubky		
17		<CDIR>	INT	Směr frézování	0 =	Sousledné frézování
					1 =	Nesousledné frézování
18	UXY	<_FAL>	REAL	Přídavek rozměru pro opracování načisto v rovině nebo na okrajích drážky		
19		<VARI>	INT	Způsob opracování		
				JEDNOTKY:		
					0 =	Kompletní opracování
					1 =	Obrábění nahrubo
					2 =	Obrábění načisto
					3 =	Obrábění okraje načisto
					5 =	Srážení hran
				DESÍTKY:		
					0 =	najíždění na pomocné polohy po přímkách s G0
					1 =	najíždění na pomocné polohy po kruhové dráze
				STOVKY:	rezervováno	
				TISÍCE:		
					0 =	Režim kompatibility, pokud je <INDA> = 0, pak celá kružnice, <INDA> <> 0, potom kruhový oblouk
					1 =	Kružnice
					2 =	Kruhový oblouk
20	DZF	<_MIDF>	REAL	rezervováno		
21		<FFP2>	REAL	rezervováno		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
22		<SSF>	REAL	rezervováno		
23	FF	<_FFCP>	REAL	rezervováno		
24		<_UMODE>	INT	rezervováno		
25	FS	<_FS>	REAL	Šířka fasety (ink)		
26	ZFS	<_ZFS>	REAL	Hloubka zajíždění (špička nástroje) při srážení hran (abs/ink), viz <_AMODE>		
27		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	rezervováno	
				STOVKY:	Volba opracování nebo pouze výpočet počátečního bodu	
					0 =	Režim kompatibility
					1 =	normální opracování
28		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	rezervováno	
				STOVKY:	rezervováno	
				TISÍCE:	Identifikace verze SW	
					1 =	funkce SLOT2 od SW verze 2.5
				DESÍTKY TISÍC:	Odstupňování technologie na obrazovkách cyklů (Strana 1169)	
					0 =	Zadání: kompletní
					1 =	Zadání: jednoduché
29		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Konečná hloubka Z1 (abs/ink)	
					0 =	Kompatibilita
					1 =	Z1 (ink)
					2 =	Z1 (abs)
				DESÍTKY:	rezervováno	
				STOVKY:	Hloubka zajíždění při srážení hran ZFS	
					0 =	ZFS (abs)
					1 =	ZFS (ink)

4.26.1.9 LONGHOLE - Podlouhlá díra

Syntaxe

```
LONGHOLE(<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <LENG>, <_CPA>,
<_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <MID>, <_VARI>,
<_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	RP	<RTP>	REAL	Návratová rovina (abs)		
2	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)		
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
4	Z1	<_DP>	REAL	Hloubka podlouhlé díry (abs)		
5		<_DPR>	REAL	Hloubka podlouhlé díry (ink), vztažená na Z0 (zadává se bez znaménka)		
6		<NUM>	INT	Počet podlouhlých děr = 1		
7	L	<LENG>	REAL	Délka podlouhlé díry		
8	X0	<_CPA>	REAL	Vztažný bod, 1. osy v rovině		
9	Y0	<_CPO>	REAL	Vztažný bod, 2. osy v rovině		
10		<RAD>	REAL	rezervováno		
11	α0	<STA1>	REAL	Úhel otočení		
12		<INDA>	REAL	rezervováno		
13	FZ	<FFD>	REAL	Rychlost přísuvu do hloubky		
14	F	<FFP1>	REAL	Posuv		
15	DZ	<MID>	REAL	Maximální přísuv do hloubky		
16		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:	Druh přísuvu	
					1 =	kolmo s G1
					3 =	oscilačním pohybem
				STOVKY:	rezervováno	
17		<_UMODE>	INT	rezervováno		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
18		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	rezervováno	
				STOVKY:	Volba opracování nebo pouze výpočet počátečního bodu	
					0 =	Režim kompatibility
					1 =	normální opracování
				TISÍCE:	Kótování vztažného bodu, poloha drážky	
					0 =	ve středu
					1 =	vlevo uvnitř +L
					2 =	vpravo uvnitř -L
					3 =	levý okraj +L
					4 =	pravý okraj -L
19		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	Druh posuvu: G-skupina (G94/G95) pro posuv v rovině a posuv do hloubky	
					0 =	Režim kompatibility
					1 =	Příkaz G-funkce stejný jako před voláním cyklu G94/G95 pro posuv v rovině a posuv do hloubky zůstávají stejné.
				STOVKY:	rezervováno	
				TISÍCE:	Identifikace verze SW	
					1 =	Rozšíření funkce LONGHOLE (kótování vztažného bodu)
20		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Konečná hloubka Z1 (abs/ink)	
					0 =	Kompatibilita
					1 =	Z1 (ink)
					2 =	Z1 (abs)

Poznámka

Cyklus je oproti předcházejícím verzím SW vybaven novými funkcemi. To má za následek, že určité parametry se na vstupní obrazovce už nevypisují (<NUM>, <RAD>, <INDA>). Větší počet drážek na polohovacím vzoru je možné naprogramovat pomocí příkazu "MCALL" a voláním požadovaného polohovacího vzoru, např. HOLES2.

4.26.1.10 CYCLE60 - Gravírování

Syntaxe

```
CYCLE60(<_TEXT>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_PA>,
<_PO>, <_STA>, <_CP1>, <_CP2>, <_WID>, <_DF>, <_FFD>, <_FFP1>,
<_VARI>, <_CODEP>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1		<_TEXT>	STRING [200]	Text, který má být vygravírován (maximálně 100 znaků)
2	RP	<_RTP>	REAL	Návratová rovina (abs)
3	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)
4	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se k referenční rovině, zadává se bez znaménka)
5	Z1	<_DP>	REAL	Hloubka (abs), viz <_AMODE>
6	Z1	<_DPR>	REAL	Hloubka (ink), viz _<_AMODE>
7	X0	<_PA>	REAL	Vztažný bod, 1. osa v rovině (abs) - pravoúhlé souřadnice, viz <_VARI>
	R			Vztažný bod, délka (rádius) - polární souřadnice, viz <_VARI>
8	Y0	<_PO>	REAL	Vztažný bod, 2. osa v rovině (abs) - pravoúhlé souřadnice, viz <_VARI>
	α0			Vztažný bod, úhel vztažený na 1. osu - polární souřadnice, viz <_VARI>
9	α1	<_STA>	REAL	Směr textu, úhel textové čáry vztažený na 1. osu, viz <_VARI>
10	XM	<_CP1>	REAL	Střed textového kruhového oblouku, 1. osa v rovině (abs) - pravoúhlé souřadnice, viz <_VARI>
	LM			Střed textového kruhového oblouku, délka (rádius) vztažená na počátek WCS - polární souřadnice, viz <_VARI>
11	YM	<_CP2>	REAL	Střed textového kruhového oblouku, 2. osa v rovině (abs) - pravoúhlé souřadnice, viz <_VARI>
	αM			Střed textového kruhového oblouku, úhel vztažený na 1. osu - polární souřadnice, viz <_VARI>
12	W	<_WID>	REAL	Výška znaků (zadává se bez znaménka)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
13	DX1 DX2	<_DF>	REAL	Vzdálenost mezi znaky / celková šířka, viz <_VARI>		
	α2			Úhel kruhové výseče, viz <_VARI>		
14	FZ	<_FFD>	REAL	Posuv při přísuvu do hloubky, viz <_DMODE>		
15	F	<_FFP1>	REAL	Posuv pro opracovávání plochy		
16		<_VARI>	INT	Opracování (vyrovnání a vztažný bod gravírovaného textu)		
				JEDNOTKY:	Vztažný bod	
					0 =	pravoúhlý
					1 =	polární
				DESÍTKY:	Vyrovnání textu	
					0 =	Text na přímce
					1 =	Text shora na kruhovém oblouku
					2 =	Text zdola na kruhovém oblouku
				STOVKY:	rezervováno	
				TISÍCE:	Vztažný bod textu ve vodorovném směru	
					0 =	vlevo
					1 =	na střed
					2 =	vpravo
				DESÍTKY TISÍC:	Vztažný bod textu ve svislém směru	
					0 =	dole
					1 =	na střed
					2 =	nahoře
				STOVKY TISÍC:	Délka textu	
					0 =	Vzdálenost mezi znaky
					1 =	Celková šířka textu (jen u textu na přímce)
					2 =	Úhel kruhové výseče (jen u textu na kruhovém oblouku)
				JEDNOTKY MILIONŮ:	Střed kruhu	
					0 =	pravoúhlé (kartézské) souřadnice
					1 =	polární
				DESÍTKY MILIONŮ:	Zrcadlově převrácené písmo	
					0 =	Kompatibilita
					1 =	Zrcadlově převrácené písmo aktivováno
					2 =	Zrcadlově převrácené písmo deaktivováno
17		<_CODEP>	INT	Číslo kódové stránky pro písmo (v současnosti pouze 1252)		
18		<_UMODE>	INT	rezervováno		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
19		_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	rezervováno	
				STOVKY:	Volba opracování/pouze výpočet počátečního bodu	
					0 =	Režim kompatibility
					1 =	normální opracování
20		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	Druh posuvu: G-skupina (G94/G95) pro posuv v rovině a posuv do hloubky	
					0 =	Režim kompatibility
					1 =	Příkaz G-funkce stejný jako před voláním cyklu G94/G95 pro posuv v rovině a posuv do hloubky zůstávají stejné.
21		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Konečná hloubka (<_DP>, <_DPR>)	
					0 =	Kompatibilita
					1 =	inkrementálně (<_DPR>)
					2 =	absolutně (<_DP>)

4.26.1.11 CYCLE61 - Rovinné frézování

Syntaxe

```
CYCLE61(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_PA>, <_PO>, <_LENG>,
<_WID>, <_MID>, <_MIDA>, <_FALD>, <_FFP1>, <_VARI>, <_LIM>,
<_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	RP	<_RTP>	REAL	Návratová rovina (abs)
2	Z0	<_RFP>	REAL	Vztažný bod osy nástroje, výška surového obrobku (abs)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
3	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
4	Z1	<_DP>	REAL	Výška hotového obrobku (abs/ink), viz <_AMODE>		
5	X0	<_PA>	REAL	Rohový bod 1 na 1. ose (abs)		
6	Y0	<_PO>	REAL	Rohový bod 1 na 2. ose (abs)		
7	X1	<_LENG>	REAL	Rohový bod 2 na 1. ose (abs/ink), viz <_AMODE>		
8	Y1	<_WID>	REAL	Rohový bod 2 na 2. ose (abs/ink), viz <_AMODE>		
9	DZ	<_MID>	REAL	Maximální přísuv do hloubky		
10	DXY	<_MIDA>	REAL	Maximální přísuv v rovině (jednotky viz <_AMODE>)		
11	UZ	<_FALD>	REAL	Přídavek rozměru pro opracování načisto na dně		
12	F	<_FFP1>	REAL	Pracovní posuv		
13		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:	Opracování	
					1 =	Obrábění nahrubo
					2 =	Obrábění načisto
				DESÍTKY:	Směr obrábění	
					1 =	rovnoběžně s 1. osou, jeden směr
					2 =	rovnoběžně s 2. osou, jeden směr
					3 =	rovnoběžně s 1. osou, střídající se směr
					4 =	rovnoběžně s 2. osou, střídající se směr
14		<_LIM>	INT	Ohraničení		
				JEDNOTKY:	Ohraničení pro 1. osu ve směru mínus	
					0 =	ne
					1 =	ano
				DESÍTKY:	Ohraničení pro 1. osu ve směru plus	
					0 =	ne
					1 =	ano
				STOVKY:	Ohraničení pro 2. osu ve směru mínus	
					0 =	ne
					1 =	ano
				TISÍCE:	Ohraničení pro 2. osu ve směru plus	
					0 =	ne
					1 =	ano
15		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
16		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Konečná hloubka (<_DP>)	
					0 =	absolutní
					1 =	inkrementální
				DESÍTKY:	Jednotky pro přísuv v rovině (<_MIDA>)	
					0 =	mm
					1 =	procentuální hodnota z průměru nástroje
				STOVKY:	rezervováno	
				TISÍCE:	Délka plochy	
					0 =	inkrementální
					1 =	absolutní
				DESÍTKY TISÍC:	Šířka plochy	
					0 =	inkrementální
					1 =	absolutní

4.26.1.12 CYCLE62 - volání kontury

Syntaxe

```
CYCLE62(<_KNAME>,<_TYPE>, <_LAB1>, <_LAB2>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	PRG/CON	<_KNAME>	STRING [140]	Název kontury nebo podprogramu, nemusí být naprogramován, když je _TYPE = 2		
2		<_TYPE>	INT	Stanovení zadání kontury		
					0 =	Podprogram
					1 =	Název kontury
					2 =	Návěští
					3 =	Návěští v podprogramu
3	LAB1	<_LAB1>	STRING[32]	Návěští 1, začátek kontury		
4	LAB2	<_LAB2>	STRING[32]	Návěští 2, konec kontury		

4.26.1.13 CYCLE63 - Frézování konturové kapsy / odstranění zbytkového materiálu z konturové kapsy / frézování konturového čepu / odstranění zbytkového materiálu z konturového čepu

Syntaxe

```
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>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	PRG	<_PRG>	STRING [100]	Název programu pro odstraňování materiálu		
2		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:	Technologie opracování	
					1 =	Obrábění nahrubo
					3 =	Obrábění dna načisto
					4 =	Obrábění okraje načisto
					5 =	Srážení hran
				DESÍTKY:	Způsob přísuvu	
					0 =	Zajíždění nástroje středem
					1 =	Zajíždění po šroubovici
					2 =	Zajíždění oscilačním pohybem
				STOVKY:	rezervováno	
				TISÍCE:	Způsob pozvednutí nástroje	
					0 =	pozvednout na návratovou rovinu
					1 =	pozvednout na vztažný bod + bezpečnostní vzdálenost
				DESÍTKY TISÍC:	Počáteční bod při obrábění nahrubo a načisto na dně	
					0 =	automaticky
					1 =	manuálně
3	RP	<_RP>	REAL	Návratová rovina (abs)		
4	Z0	<_Z0>	REAL	Vztažný bod osy nástroje (abs)		
5	SC	<_SC>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
6	Z1	<_Z1>	REAL	Konečná hloubka, (viz JEDNOTKY parametru <_AMODE>)		
7	F	<_F>	REAL	Posuv v rovině při obrábění nahrubo/načisto		
8	FZ	<_FZ>	REAL	Rychlost přísuvu do hloubky		
9	DXY	<_DXY>	REAL	Přísuv v rovině - jednotky (viz DESÍTKY parametru <_AMODE>)		
10	DZ	<_DZ>	REAL	Přísuv do hloubky		
11	UXY	<_UXY>	REAL	Přídavek rozměru pro opracování načisto v rovině		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
12	UZ	<_UZ>	REAL	Přídavek rozměru pro opracování načisto na dně		
13		<_CDIR>	INT	Směr frézování	0 =	Sousledné frézování
					1 =	Nesousledné frézování
14	XS	<_XS>	REAL	Souřadnice X počátečního bodu, absolutně		
15	YS	<_YS>	REAL	Souřadnice Y počátečního bodu, absolutně		
16	ER	<_ER>	REAL	Zajíždění po spirále: Rádius		
17	EP	<_EP>	REAL	Zajíždění po spirále: Stoupání		
18	EW	<_EW>	REAL	Zajíždění oscilačním pohybem: Maximální úhel při zajíždění		
19	FS	<_FS>	REAL	Šířka fasety (ink) u srážení hran		
20	ZFS	<_ZFS>	REAL	Hloubka zajíždění špičky nástroje při srážení hran (viz STOVKY parametru <_AMODE>)		
21	TR	<_TR>	STRING[32]	Název referenčního nástroje při odstraňování zbytkového materiálu		
22	DR	<_DR>	INT	D-číslo referenčního nástroje při odstraňování zbytkového materiálu		
23		<_UMODE>	INT	rezervováno		
24		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	rezervováno	
				STOVKY:	Volba opracování/pouze výpočet počátečního bodu	
					0 =	normální opracování (není zapotřebí žádný režim kompatibility)
					1 =	normální opracování
					2 =	rezervováno

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
25		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	rezervováno	
				STOVKY:	technologický režim	
					1 =	Kapsa
					2 =	Čep
				TISÍCE:	Odstraňování zbytkového materiálu	
					0 =	ne
					1 =	ano
				DESÍTKY TISÍC:	Odstupňování technologie na obrazovkách cyklů (Strana 1169)	
					0 =	Zadání: kompletní
					1 =	Zadání: jednoduché
				STOVKY TISÍC:	Název programu automaticky	
					0 =	ne
					1 =	ano
26		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Koncová hloubka (Z1)	
					0 =	absolutní (režim kompatibility)
					1 =	inkrementální
				DESÍTKY:	Jednotky pro přísuv v rovině (DXY)	
					0 =	mm
					1 =	procentuální hodnota z průměru nástroje
				STOVKY:	Hloubka zajíždění při srážení hran (ZFS)	
					0 =	absolutní
					1 =	inkrementální
				TISÍCE:	---	rezervováno

4.26.1.14 CYCLE64 - Předvrtání konturové kapsy

Syntaxe

```
CYCLE64(<_PRG>, <_VARI>, <_RP>, <_Z0>, <_SC>, <_Z1>, <_F>, <_DXY>,
<_UXY>, <_UZ>, <_CDIR>, <_TR>, <_DR>, <_UMODE>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	PRG	<_PRG>	STRING [100]	Název programu pro vrtání/navrtávání středicích důlků		
2		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	rezervováno	
				STOVKY:	rezervováno	
				TISÍCE:	Způsob pozvednutí nástroje	
					0 =	pozvednout na návratovou rovinu
					1 =	pozvednout na vztažný bod + bezpečnostní vzdálenost
3	RP	<_RP>	REAL	Návratová rovina (abs)		
4	Z0	<_Z0>	REAL	Vztažný bod (abs)		
5	SC	<_SC>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
6	Z1	<_Z1>	REAL	Hloubka vrtání/navrtávání středicích důlků, (viz JEDNOTKY parametru <_AMODE>)		
7	F	<_F>	REAL	Posuv pro vrtání/navrtávání středicích důlků		
8	DXY	<_DXY>	REAL	Přísuv v rovině - jednotky (viz DESÍTKY parametru <_AMODE>)		
9	UXY	<_UXY>	REAL	Přídavek rozměru pro opracování načisto v rovině		
10	UZ	<_UZ>	REAL	Přídavek rozměru pro opracování načisto na dně		
11		<_CDIR>	INT	Směr frézování	0 =	Sousledné frézování
					1 =	Nesousledné frézování
12	TR	<_TR>	STRING[20]	Název referenčního nástroje		
13	DR	<_DR>	INT	D-číslo referenčního nástroje		
14		<_UMODE>	INT	rezervováno		
15		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	rezervováno	
				STOVKY:	Volba opracování/pouze výpočet počátečního bodu	
					0 =	normální opracování (není zapotřebí žádný režim kompatibility)
					1 =	normální opracování
					2 =	rezervováno

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
25		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	technologický režim	
					1 =	Předvrtání
					2 =	Navrtávání středicích důlků
				STOVKY:	---	rezervováno
				TISÍCE:	---	rezervováno
				DESÍTKY TISÍC:	---	rezervováno
				STOVKY TISÍC:	Název programu automaticky	
					0 =	ne
					1 =	ano
26		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Hloubka vrtání/navrtávání středicích důlků Z1	
					0 =	absolutní (režim kompatibility)
					1 =	inkrementální
				DESÍTKY:	Jednotky pro přísuv v rovině (DXY)	
					0 =	mm
					1 =	procentuální hodnota z průměru nástroje

4.26.1.15 CYCLE70 - Frézování závitu

Syntaxe

```
CYCLE70(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DIATH>, <_H1>, <_FAL>,
<_PIT>, <_NT>, <_MID>, <_FFR>, <_TYPTH>, <_PA>, <_PO>, <_NSP>,
<_VARI>, <_PITA>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	RP	<_RTP>	REAL	Návratová rovina (abs)
2	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
3	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
4	Z1	<_DP>	REAL	Délka závitu (abs, ink), viz <_AMODE> Zohledňovat vyjíždění na dně vyvrtané díry (minimálně polovina stoupání)		
5	∅	<_DIATH>	REAL	Jmenovitý průměr závitu		
6	H1	<_H1>	REAL	Hloubka závitu		
7	U	<_FAL>	REAL	Přídavek rozměru pro opracování načisto		
8	P	<_PIT>	REAL	Stoupání závitu (možnosti na výběr <_PITA>: mm, palce, MODUL, chodů/palec)		
9	NT	<_NT>	INT	Počet zubů na řezné destičce Délka nástroje je vždy vztažena na spodní zub!		
10	DXY	<_MID>	REAL	Maximální přísuv za řez <_MID> > <_H1>: vše s jedním řezem		
11	F	<_FFR>	REAL	Posuv při frézování		
12		<_TYPTH>	INT	Typ závitu	0 =	Vnitřní závit
					1 =	Vnější závit
13	X0	<_PA>	REAL	Střed kruhu, na 1. ose (abs)		
14	Y0	<_PO>	REAL	Střed kruhu, na 2. ose (abs)		
15	αS	<_NSP>	REAL	Počáteční úhel (vícechodý závit)		
16		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:		
					1 =	Obrábění nahrubo
					2 =	Obrábění načisto
				DESÍTKY:		
					1 =	shora dolů
					2 =	zdola nahoru
				STOVKY:		
					0 =	Pravý závit
					1 =	Levý závit
17		<_PITA>	INT	Vyhodnocování stoupání závitu		
					0 =	Režim kompatibility
					1 =	stoupání v mm
					2 =	stoupání v chodech na palec (TPI)
					3 =	stoupání v palcích
					4 =	stoupání jako MODUL
18		<_PITM>	STRING[15]	Řetězec jako ukazatel pro zadávání stoupání závitu (jen pro plochu)		
19		<_PTAB>	STRING[20]	Řetězec pro tabulku závitů ("", "ISO", "BSW", "BSP", "UNC") (jen pro pracovní plochu)		
20		<_PTABA>	STRING[20]	Řetězec pro volbu v tabulce závitů (např. "M 10", "M 12", ...) (pouze pro povrch)		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
21		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	rezervováno	
				STOVKY:	Opracování/výpočet počátečního bodu	
					0 =	Režim kompatibility
					1 =	normální opracování
22		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
23		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Délka závitu (<_DP>)	
					0 =	absolutní
					1 =	inkrementální

4.26.1.16 CYCLE72 - Frézování po dráze

Syntaxe

```
CYCLE72(<_KNAME>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_MID>, <_FAL>,
<_FALD>, <_FFP1>, <_FFD>, <_VARI>, <_RL>, <_AS1>, <_LP1>, <_FF3>,
<_AS2>, <_LP2>,<_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1		<_KNAME>	STRING [141]	Název podprogramu kontury
2	RP	<_RTP>	REAL	Návratová rovina (abs)
3	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)
4	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)
5	Z1	<_DP>	REAL	Koncový bod, konečná hloubka (abs/ink), viz <_AMODE>
6	DZ	<_MID>	REAL	Maximální přísuv do hloubky (ink, zadává se bez znaménka)
7	UXY	<_FAL>	REAL	Přídavek rozměru pro opracování načisto v rovině (ink), přídavek rozměru na okraji kontury
8	UZ	<_FALD>	REAL	Přídavek rozměru pro obrábění načisto do hloubky (ink), přídavek rozměru na dně (zadává se bez znaménka)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
9	FX	<_FFP1>	REAL	Posuv na kontuře		
10	FZ	<_FFD>	REAL	Posuv pro přísuv do hloubky (nebo přísuv v prostoru)		
11		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:	Opracování	
					1 =	Obrábění nahrubo
					2 =	Obrábění načisto
					5 =	Srážení hran
				DESÍTKY:		
					0 =	Pomocné úseky dráhy s G0
					1 =	Pomocné úseky dráhy s G1
				STOVKY:	Zpětný pohyb na konci kontury	
					0 =	Zpětný pohyb na konci kontury na vztažný bod
					1 =	Zpětný pohyb na konci kontury na vztažný bod + <_SDIS>
					2 =	Zpětný pohyb na konci kontury o <_SDIS>
					3 =	Žádný zpětný pohyb na konci kontury, na následující počáteční bod se najíždí s posuvem po kontuře
				TISÍCE:	rezervováno	
				DESÍTKY TISÍC:	Opracování kontury	
					0 =	Opracování kontury směrem dopředu
					1 =	Opracování kontury směrem pozpátku Omezení při pohybu směrem pozpátku • max. 170 konturových prvků (včetně faset nebo zaoblení) • jsou vyhodnocovány pouze hodnoty v rovině (X/Y) a hodnota F
12		<_RL>	INT	Směr obrábění		
					40 =	objíždění kontury středem nástroje (G40, najíždění a odjíždění: pouze po přímce nebo kolmo)
					41 =	vlevo od kontury (G41, najíždění a odjíždění: pouze po přímce nebo po kruhové dráze)
					42 =	vpravo od kontury (G42, najíždění a odjíždění: pouze po přímce nebo po kruhové dráze)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
13		<_AS1>	INT	Najížděcí pohyb na konturu		
				JEDNOTKY:		
					1 =	Přímka
					2 =	Čtvrtkruh
					3 =	Půlkruh
					4 =	Najíždění a odjíždění kolmo
				DESÍTKY:		
					0 =	poslední pohyb, v rovině
					1 =	poslední pohyb, v prostoru
14	L1	<_LP1>	REAL	Najížděcí dráha nebo najížděcí rádius (ink, zadává se bez znaménka)		
15	FZ	<_FF3>	REAL	Posuv pro pomocné úseky dráhy (G94/G95, jako na kontuře)		
16		<_AS2>	INT	Odjíždějící pohyb od kontury (nikoli při najíždění/odjíždění kolmo)		
				JEDNOTKY:		
					1 =	Přímka
					2 =	Čtvrtkruh
					3 =	Půlkruh
				DESÍTKY:		
					0 =	poslední pohyb, v rovině
					1 =	poslední pohyb, v prostoru
17	L2	<_LP2>	REAL	Odjíždějící dráha nebo odjíždějící rádius (ink, zadává se bez znaménka)		
18		<_UMODE>	INT	rezervováno		
19	FS	<_FS>	REAL	Šířka fasety (ink)		
20	ZFS	<_ZFS>	REAL	Hloubka zajíždění (špička nástroje) při srážení hran (abs/ink), viz <_AMODE>		
21		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	rezervováno	
				STOVKY:	Volba opracování/pouze výpočet počátečního bodu	
					0 =	Režim kompatibility
					1 =	normální opracování

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
22		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	Druh posuvu: G-skupina (G94/G95) pro posuv v rovině a posuv do hloubky	
					0 =	Režim kompatibility
					1 =	Příkaz G-funkce stejný jako před voláním cyklu G94/G95 pro posuv v rovině a posuv do hloubky zůstávají stejné.
				TISÍCE:		
					0 =	Režim kompatibility: Název kontury je uložen v <_KNAME>
					1 =	název kontury je naprogramován v cyklu CYCLE62 a předává se v parametru _SC_CONT_NAME
23		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Koncový bod Z1 (<_DP>)	
					0 =	absolutní (režim kompatibility)
					1 =	inkrementální
				DESÍTKY:	Jednotky pro přísuv v rovině	
					0 =	mm, palce
					1 =	rezervováno
				STOVKY:	Hloubka zajíždění při srážení hran (<_ZFS>)	
					0 =	absolutní
					1 =	inkrementální

Poznámka

Pokud jsou následující předávané parametry naprogramovány nepřímo (jako parametry), zpětný překlad vstupní obrazovky nebude možný:

<_VARI>, <_RL>, <_AS1>, <_AS2>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>

4.26.1.17 CYCLE76 - Pravoúhlý čep

Syntaxe

```
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>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	RP	<_RTP>	REAL	Návratová rovina (abs)		
2	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)		
3	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
4	Z1	<_DP>	REAL	Hloubka čepu (abs)		
5		<_DPR>	REAL	Hloubka čepu (ink), vztažená na Z0 (zadává se bez znaménka)		
6	L	<_LENG>	REAL	Délka čepu, viz <_GMODE> (zadává se bez znaménka)		
7	W	<_WID>	REAL	Šířka čepu, viz <_GMODE> (zadává se bez znaménka)		
8	R	<_CRAD>	REAL	Rádius v rohu čepu (zadává se bez znaménka)		
9	X0	<_PA>	REAL	Vztažný bod čepu, 1. osa v rovině (abs)		
10	Y0	<_PO>	REAL	Vztažný bod čepu, 2. osa v rovině (abs)		
11	α0	<_STA>	REAL	Úhel otočení, úhel mezi podélnou osou (L) a 1. osou roviny		
12	DZ	<_MID>	REAL	Maximální přísuv do hloubky (ink, zadává se bez znaménka)		
13	UXY	<_FAL>	REAL	Přídavek rozměru pro opracování načisto v rovině (ink), přídavek rozměru na okraji kontury		
14	UZ	<_FALD>	REAL	Přídavek rozměru pro obrábění načisto do hloubky (ink), přídavek rozměru na dně (zadává se bez znaménka)		
15	FX	<_FFP1>	REAL	Posuv na kontuře		
16	FZ	<_FFD>	REAL	Rychlost přísuvu do hloubky		
17		<_CDIR>	INT	Směr frézování (zadává se bez znaménka)		
				JEDNOTKY:		
					0 =	Sousledné frézování
					1 =	Nesousledné frézování
18		<_VARI>	INT	Opracování		
				JEDNOTKY:		
					1 =	Obrábění nahrubo
					2 =	Obrábění načisto
					5 =	Srážení hran
19	L1	<_AP1>	REAL	Délka surového čepu		
20	W1	<_AP2>	REAL	Šířka surového čepu		
21	FS	<_FS>	REAL	Šířka fasety (ink)		
22	ZFS	<_ZFS>	REAL	Hloubka zajíždění (špička nástroje) při srážení hran (abs, ink), viz <_AMODE>		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
23		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	rezervováno	
				STOVKY:	Volba opracování nebo pouze výpočet počátečního bodu	
					0 =	Režim kompatibility
					1 =	normální opracování
				TISÍCE:	Kótování čepu pomocí středu nebo rohu	
					0 =	Režim kompatibility
					1 =	Zadávání rozměrů pomocí středu
					2 =	Kótování pomocí rohového bodu, čep +L +W
					3 =	Kótování pomocí rohového bodu, čep -L +W
					4 =	Kótování pomocí rohového bodu, čep +L -W
					5 =	Kótování pomocí rohového bodu, čep -L -W
				DESÍTKY TISÍC:	Kompletní opracování nebo dopracování	
					0 =	Režim kompatibility
					1 =	Kompletní opracování
					2 =	Dodatečné opracování
24		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	---	rezervováno
				STOVKY:	---	rezervováno
				TISÍCE:	---	rezervováno
				DESÍTKY TISÍC:	Odstupňování technologie na obrazovkách cyklů (Strana 1169)	
					0 =	Zadání: kompletní
					1 =	Zadání: jednoduché

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
25		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Koncová hloubka Z1 (DP)	
					0 =	Kompatibilita
					1 =	inkrementální
					2 =	absolutní
				DESÍTKY:	rezervováno	
				STOVKY:	Hloubka zajíždění při srážení hran (ZFS)	
					0 =	absolutní
					1 =	inkrementální

4.26.1.18 CYCLE77 - Kruhový čep

Syntaxe

```
CYCLE77(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_CDIAM>, <_PA>,
<_PO>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>, <_VARI>,
<_AP1>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	RP	<_RTP>	REAL	Návratová rovina (abs)		
2	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)		
3	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
4	Z1	<_DP>	REAL	Hloubka čepu (abs)		
5		<_DPR>	REAL	Hloubka čepu (ink), vztažená na Z0 (zadává se bez znaménka)		
6	∅	<_CDIAM>	REAL	Průměr čepu (zadává se bez znaménka)		
7	X0	<_PA>	REAL	Vztažný bod čepu, 1. osa v rovině (abs)		
8	Y0	<_PO>	REAL	Vztažný bod čepu, 2. osa v rovině (abs)		
9	DZ	<_MID>	REAL	Maximální přísuv do hloubky (ink, zadává se bez znaménka)		
10	UXY	<_FAL>	REAL	Přídavek rozměru pro opracování načisto v rovině (ink), přídavek rozměru na okraji kontury		
11	UZ	<_FALD>	REAL	Přídavek rozměru pro obrábění načisto do hloubky (ink), přídavek rozměru na dně (zadává se bez znaménka)		
12	FX	<_FFP1>	REAL	Posuv na kontuře		
13	FZ	<_FFD>	REAL	Rychlost přísuvu do hloubky		
14		<_CDIR>	INT	Směr frézování (zadává se bez znaménka)		
				JEDNOTKY:		
					0 =	Sousledné frézování
					1 =	Nesousledné frézování

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
15		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:	Opracování	
					1 =	Obrábění nahrubo až na přídavek rozměru pro opracování načisto
					2 =	Obrábění načisto (přídavek rozměru X/Y/Z = 0)
					5 =	Srážení hran
16	∅1	<_AP1>	REAL	Průměr surového čepu		
17	FS	<_FS>	REAL	Šířka fasety (ink)		
18	ZFS	<_ZFS>	REAL	Hloubka zajíždění (špička nástroje) při srážení hran (abs/ink), viz <_AMODE>		
19		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	rezervováno	
				STOVKY:	Volba opracování/pouze výpočet počátečního bodu	
					0 =	Režim kompatibility
					1 =	normální opracování
				TISÍCE:	rezervováno	
				DESÍTKY TISÍC:	Kompletní opracování/dopracování	
					0 =	Režim kompatibility (s parametry <_AP1> se zachází jako předtím)
					1 =	Kompletní opracování
					2 =	Dodatečné opracování
20		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	---	rezervováno
				STOVKY:	---	rezervováno
				TISÍCE:	---	rezervováno
				DESÍTKY TISÍC:	Odstupňování technologie na obrazovkách cyklů (Strana 1169)	
					0 =	Zadání: kompletní
					1 =	Zadání: jednoduché

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
21		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Koncová hloubka Z1 (DP)	
					0 =	absolutní (režim kompatibility)
					1 =	inkrementální
					2 =	absolutní
				DESÍTKY:	rezervováno	
				STOVKY:	Hloubka zajíždění při srážení hran (ZFS)	
					0 =	absolutní
					1 =	inkrementální

4.26.1.19 CYCLE78 - Frézování vrtaných závitů

Syntaxe

```
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>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	RP	<_RTP>	REAL	Návratová rovina (abs)		
2	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)		
3	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
4	Z1	<_DP>	REAL	Konečná vrtaná hloubka (abs/ink), viz <_AMODE>		
5		<_ADPR>	REAL	Navrtávaná hloubka se sníženou hodnotou posuvu při vrtání (ink), v platnosti podle místa desítek tisíc parametru <_VARI>		
6	D	<_FDPR>	REAL	Maximální přísuv do hloubky (ink) D ≥ Z1 --> jeden přísuv až na konečnou vrtanou hloubku D < Z1 --> cyklus pro vrtání hlubokých děr s větším počtem přísuvů a s odstraňováním třísek		
7	ZR	<_LDPR>	REAL	Zbytková vrtaná hloubka při provrtávání (ink), s posuvem FR		
8	Ø	<_DIAM>	REAL	Jmenovitý průměr závitu		
9	P	<_PIT>	REAL	Stoupání závitu jako číselná hodnota		
10		<_PITA>	INT	Vyhodnocování stoupání závitu P		
					1 =	stoupání v mm/ot
					2 =	stoupání v chodech/palec
					3 =	stoupání v palcích/ot
					4 =	stoupání jako MODUL

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
11	DF	<_DAM>	REAL	Absolutní hodnota/procentuální změna pro každý další přísuv (degrese), viz <_AMODE>		
12	V1	<_MDEP>	REAL	minimální přísuv (ink), jen když je používána degrese		
13		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	Odstraňování třísek před frézováním závitu	
					0 =	žádné odstraňování třísek před frézováním závitu (uplatňuje se pouze na konečné vrtané hloubce)
					1 =	odstraňování třísek před frézováním závitu (uplatňuje se pouze na konečné vrtané hloubce)
				STOVKY:	Pravý/levý závit	
					0 =	Pravý závit
					1 =	Levý závit
				TISÍCE:	Zbývající vrtaná hloubka s posuvem pro vrtání	
					0 =	žádná zbytková vrtaná hloubka s posuvem při vrtání FR
					1 =	zbytková vrtaná hloubka s posuvem při vrtání FR
				DESÍTKY TISÍC:	Navrtávání se sníženou hodnotou posuvu	
					0 =	žádné navrtávání se sníženou hodnotou posuvu
					1 =	Navrtávání se sníženou hodnotou posuvu posuv při navrtávání = 0,3 F1, pokud je F1 < 0,15 mm/ot posuv při navrtávání = 0,1 mm/ot, pokud je F1 ≥0,15 mm/ot
14		<_CDIR>	INT	Směr frézování	0 =	Sousledné frézování
					1 =	Nesousledné frézování
					4 =	nesousledné + sousledné (kombinace obrábění nahrubo + načisto)
15	Z2	<_GE>	REAL	Vzdálenost zpětného pohybu před frézováním závitu (ink)		
16	F1	<_FFD>	REAL	Posuv při vrtání (mm/min, příp. palce/min nebo mm/ot)		
17	FR	<_FRDP>	REAL	Posuv při vrtání pro zbytkovou vrtanou hloubku (mm/min nebo mm/ot)		
18	F2	<_FFR>	REAL	Posuv pro frézování závitu (mm/min nebo mm/zub)		
19	FS	<_FFP2>	REAL	Posuv při obrábění načisto pro <_CDIR> = 4 (mm/min nebo mm/zub)		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
20		<_FFA>	INT	Vyhodnocování posuvu		
				JEDNOTKY:	posuv při vrtání F1	
				DESÍTKY:	posuv při vrtání pro zbytkovou vrtanou hloubku FR	
				STOVKY:	Posuv pro frézování závitu F2	
				TISÍCE:	Posuv při obrábění načisto FS	
21		<_PITM>	STRING[15]	Řetězec jako ukazatel pro zadávání stoupání závitu (jen pro plochu)[1)]		
22		<_PTAB>	STRING[20]	Řetězec pro tabulku závitů ("", "ISO", "BSW", "BSP", "UNC") (jen pro pracovní plochu)[1)]		
23		<_PTABA>	STRING[20]	Řetězec pro volbu v tabulce závitů (např. "M 10", "M 12", ...) (pouze pro povrch)[1)]		
24		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty), rezervováno		
25		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
26		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Vrtaná hloubka = konečná vrtaná hloubka Z1 (abs/ink)	
					0 =	absolutní
					1 =	inkrementální
				DESÍTKY:	Absolutní hodnota/procentuální hodnota DF pro každý další přísuv (degrese)	
					0 =	absolutní hodnota
					1 =	procentuální změna (0,001 až 100%)

Poznámka

[1)] Parametry 21, 22 a 23 se používají pouze při volbě závitu v tabulce závitů ve vstupní obrazovce. Přístup k tabulkám závitů prostřednictvím definice cyklů v době, kdy jsou cykly zpracovávány, není možný.

4.26.1.20 CYCLE79 - Mnohohran

Syntaxe

```
CYCLE79(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_NUM>, <_SWL>, <_PA>,
<_PO>, <_STA>, <_RC>, <_AP1>, <_MIDA>, <_MID>, <_FAL>, <_FALD>,
```

```
<_FFP1>, <_CDIR>, <_VARI>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	RP	<_RTP>	REAL	Návratová rovina (abs)		
2	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)		
3	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
4	Z1	<_DP>	REAL	Hloubka mnohohranu (abs/ink), viz <_AMODE>		
5	N	<_NUM>	INT	Počet hran (1...n)		
6	SW/L	<_SWL>	REAL	Velikost klíče nebo délka hrany (v závislosti na <_VARI>) ("SW" v případě velikosti klíče, "L" v případě délky hrany) Velikost klíče jen u sudého počtu hran a jednohranu		
7	X0	<_PA>	REAL	Vztažný bod čepu, 1. osa (abs)		
8	Y0	<_PO>	REAL	Vztažný bod čepu, 2. osa (abs)		
9	α0	<_STA>	REAL	Úhel otočení středu hrany vzhledem k 1. ose (osa X)		
10	R1/FS1	<_RC>	REAL	Rohové zaoblení v případě <_NUM> > 2 (rádius/faseta, viz <_AMODE>) (ink, zadává se bez znaménka) ("R1" v případě rádiusu, "FS1" v případě fasety)		
11	∅	<_AP1>	REAL	Průměr surového čepu		
12	DXY	<_MIDA>	REAL	Maximální přísuv do šířky (jednotky viz <_AMODE>)		
13	DZ	<_MID>	REAL	Maximální přísuv do hloubky		
14	UXY	<_FAL>	REAL	Přídavek rozměru pro opracování načisto v rovině		
15	UZ	<_FALD>	REAL	Přídavek rozměru pro opracování načisto na dně		
16	F	<_FFP1>	REAL	Pracovní posuv		
17		<_CDIR>	INT	Směr frézování	0 =	Sousledné frézování
					1 =	Nesousledné frézování
18		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:	Opracování	
					1 =	Obrábění nahrubo
					2 =	Obrábění načisto
					3 =	Obrábění okraje načisto
					5 =	Srážení hran
				DESÍTKY:	Velikost klíče nebo délka hrany	
					0 =	Šířka klíče
					1 =	Délka hrany
19	FS	<_FS>	REAL	Šířka fasety (ink)		
20	ZFS	<_ZFS>	REAL	Hloubka zajíždění (špička nástroje) při srážení hran (abs/ink), viz <_AMODE>		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
21		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	rezervováno	
				STOVKY:	Volba opracování nebo pouze výpočet počátečního bodu	
					1 =	normální opracování
22		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	---	rezervováno
				STOVKY:	---	rezervováno
				TISÍCE:	---	rezervováno
				DESÍTKY TISÍC:	Odstupňování technologie na obrazovkách cyklů (Strana 1169)	
					0 =	Zadání: kompletní
					1 =	Zadání: jednoduché
23		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Konečná hloubka (<_DP>)	
					0 =	absolutní
					1 =	inkrementální
				DESÍTKY:	Jednotky pro přísuv v rovině (<_MIDA>)	
					0 =	mm
					1 =	procentuální hodnota z průměru nástroje
				STOVKY:	Hloubka zajíždění při srážení hran (<_ZFS>)	
					0 =	absolutní
					1 =	inkrementální
				TISÍCE:	Zaoblením v rohu (<_RC>)	
					0 =	Rádius
					1 =	Faseta

4.26.1.21 CYCLE81 - Vrtání, navrtávání středicích důlků

Syntaxe

```
CYCLE81(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	RP	<RTP>	REAL	Návratová rovina (abs)		
2	Z0	<RFP>	REAL	Vztažný bod (abs)		
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
4	Z1/Ø	<DP>	REAL	Vrtaná hloubka (abs)/průměr středicího důlku (abs), viz <_GMODE>		
5	Z1	<DPR>	REAL	Vrtaná hloubka (ink)		
6	DT	<DTB>	REAL	Doba prodlevy na konečné vrtané hloubce, viz <_AMODE>		
7		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	Navrtávání středicího důlku je vztaženo na hloubku/průměr	
					0 =	Kompatibilita, hloubka
					1 =	Průměr
8		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
9		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Vrtaná hloubka Z1 (abs/ink)	
					0 =	kompatibilita, podle naprogramování DP/DPR
					1 =	inkrementální
					2 =	absolutní
				DESÍTKY:	Doba prodlevy na konečné vrtané hloubce DT v sekundách/otáčkách	
					0 =	kompatibilita, vyplývá ze znaménka DTB (> 0 sekundy nebo < 0 otáčky)
					1 =	v sekundách
					2 =	v otáčkách

4.26.1.22 CYCLE82 - Vrtání, čelní zahlubování

Syntaxe

```
CYCLE82(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>,
<_DMODE>, <_AMODE>, <_VARI>, <S_ZA>, <S_FA>, <S_ZD>, <S_FD>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	RP	<RTP>	REAL	Návratová rovina (abs)		
2	Z0	<RFP>	REAL	Vztažný bod (abs)		
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
4	Z1	<DP>	REAL	Vrtaná hloubka (abs), viz <_AMODE>		
5	Z1	<DPR>	REAL	Vrtaná hloubka (ink), viz <_AMODE>		
6	DT	<DTB>	REAL	Doba prodlevy na konečné vrtané hloubce, viz <_AMODE>		
7		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	Vrtaná hloubka vztažena na špičku/stopku	
					0 =	Kompatibilita, špička
					1 =	Stopka
8		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	rezervováno	
				STOVKY:	rezervováno	
				TISÍCE:	rezervováno	
				DESÍTKY TISÍC:	Odstupňování technologie na obrazovkách cyklů (Strana 1169)	
					0 =	Zadání: kompletní
					1 =	Zadání: jednoduché

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
9		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Vrtaná hloubka Z1 (abs/ink)	
					0 =	Kompatibilita, podle naprogramování DP/DPR
					1 =	inkrementální
					2 =	absolutní
				DESÍTKY:	Doba prodlevy DT na konečné vrtané hloubce v sekundách/otáčkách	
					0 =	kompatibilita, vyplývá ze znaménka DT (> 0 sekundy / < 0 otáčky)
					1 =	v sekundách
					2 =	v otáčkách
				STOVKY:	Hloubka pro navrtávání ZA abs/ink	
					0 =	inkrementální
					1 =	absolutní
				TISÍCE:	Vyhodnocování posuvu při navrtávání	
					0 =	v % z posuvu při vrtání
					1 =	F/min
					2 =	F/ot
				DESÍTKY TISÍC:	Zbytková vrtaná hloubka ZD abs/ink	
					0 =	inkrementální
					1 =	absolutní
				STOVKY TISÍC:	Vyhodnocování zbytkového posuvu při vrtání	
					0 =	v % z posuvu při vrtání
					1 =	F/min
					2 =	F/ot
10		<_VARI>	INT	Způsob opracování při navrtávání/provrtávání		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	rezervováno	
				STOVKY:	rezervováno	
				TISÍCE:	Provrtání	
					0 =	Provrtávání "ne"
					1 =	Provrtávání "ano"
				DESÍTKY TISÍC:	Navrtání	
					0 =	Navrtávání „ne"
					1 =	Navrtávání „ano“
11	ZA	<S_ZA>	REAL	Navrtávaná hloubka vztažena inkrementálně na vztažný bod nebo absolutně (viz <_AMODE> - STOVKY)		
12	FA	<S_FA>	REAL	Posuv při navrtávání jako hodnota nebo v % (ve spojení s místem TISÍCŮ parametru <_AMODE>)		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
13	ZD	<S_ZD>	REAL	Zbytková vrtaná hloubka vztažena inkrementálně na konečnou vrtanou hloubku nebo absolutně (ve spojení s místem DESÍTEK TISÍC parametru <_AMODE>)
14	FD	<S_FD>	REAL	Posuv pro zbytkovou vrtanou hloubku jako hodnota nebo v % (ve spojení s místem STOVEK TISÍCŮ parametru <_AMODE>)

4.26.1.23 CYCLE83 - Vrtání hlubokých děr 1

Syntaxe

```
CYCLE83(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <FDEP>, <FDPR>, <_DAM>,
<DTB>, <DTS>, <FRF>, <VARI>, <_AXN>, <_MDEP>, <_VRT>, <_DTD>,
<_DIS1>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	RP	<RTP>	REAL	Návratová rovina (abs)		
2	Z0	<RFP>	REAL	Vztažný bod (abs)		
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
4	Z1	<DP>	REAL	Konečná vrtaná hloubka (abs), viz <_AMODE>		
5	Z1	<DPR>	REAL	Konečná vrtaná hloubka (ink), viz <_AMODE>		
6	D	<FDEP>	REAL	1. vrtaná hloubka (abs), viz <_AMODE>		
7	D	<FDPR>	REAL	1. vrtaná hloubka (ink), viz <_AMODE>		
8	DF	<_DAM>	REAL	Degrese/procentuální hodnota pro každý další přísuv, viz <_AMODE>		
9	DTB	<DTB>	REAL	Doba prodlevy na vrtané hloubce, viz <_AMODE>		
10	DTS	<DTS>	REAL	Doba prodlevy v počátečním bodě (jen při odstraňování třísek), viz <_AMODE>		
11	FD1	<FRF>	REAL	Procentuální hodnota pro posuv při prvním přísuvu, viz <_AMODE>		
12		<VARI>	INT	Způsob opracování		
				JEDNOTKY:	Ulamování třísky / odstraňování třísek	
					0 =	Ulamování třísky
					1 =	Odstraňování třísek
13		<_AXN>	INT	Osa nástroje		
					0 =	3. geometrická osa
					1 =	1. geometrická osa
					2 =	2. geometrická osa
					> 2	3. geometrická osa
14	V1	<_MDEP>	REAL	minimální přísuv (jen když je degrese v procentech)		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
15	V2	<_VRT>	REAL	Vzdálenost zpětného pohybu po každém obrobení (jen při ulamování třísky)		
					> 0	proměnná vzdálenost zpětného pohybu
					0 =	standardní hodnota 1 mm
16	DT	<_DTD>	REAL	Doba prodlevy na konečné vrtané hloubce, viz <_AMODE>		
17	V3	<_DIS1>	REAL	Chráněná vzdálenost (jen při odstraňování třísek), viz <_AMODE>		
18		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	Vrtaná hloubka vztažena na špičku/stopku	
					0 =	Špička
					1 =	Stopka
19		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	---	rezervováno
				STOVKY:	---	rezervováno
				TISÍCE:	---	rezervováno
				DESÍTKY TISÍC:	Odstupňování technologie na obrazovkách cyklů (Strana 1169)	
					0 =	Zadání: kompletní
					1 =	Zadání: jednoduché

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
20		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Vrtaná hloubka = konečná vrtaná hloubka Z1 (abs/ink)	
					0 =	Kompatibilita, podle naprogramování <DP>/<DPR>
					1 =	inkrementální
					2 =	absolutní
				DESÍTKY:	Doba prodlevy na vrtané hloubce DTB v sekundách/otáčkách	
					0 =	Kompatibilita, vyplývá ze znaménka DTB (> 0 sekundy nebo < 0 otáčky)
					1 =	v sekundách
					2 =	v otáčkách
				STOVKY:	Doba prodlevy v počátečním bodě před DTS v sekundách/otáčkách	
					0 =	kompatibilita, vyplývá ze znaménka DTS (> 0 sekundy nebo < 0 otáčky)
					1 =	v sekundách
					2 =	v otáčkách
				TISÍCE:	Doba prodlevy na konečné vrtané hloubce DTD v sekundách/otáčkách	
					0 =	Kompatibilita, vyplývá ze znaménka DTD (> 0 sekundy nebo < 0 otáčky)
					1 =	v sekundách
					2 =	v otáčkách
				DESÍTKY TISÍC:	1. vrtaná hloubka D (abs/ink)	
					0 =	Kompatibilita, podle naprogramování <FDEPF>/<DPR>
					1 =	inkrementální
					2 =	absolutní
				STOVKY TISÍC:	Degrese/procentuální hodnota <_DAM> pro každý další přísuv	
					0 =	Kompatibilita, vyplývá ze znaménka <_DAM> (> 0 hodnota degrese nebo < 0 faktor 0,001 až 1,0)
					1 =	Absolutní hodnota degrese
					2 =	Procentuální změna (0,001 až 100%)
				JEDNOTKY MILIONŮ:	Chráněná vzdálenost V3 automaticky/manuálně	

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
					0 =	kompatibilita, vyplývá ze znaménka <_DIS1> (= 0 automaticky nebo > 0 manuálně)
					1 =	automaticky (vypočítává se v cyklu)
					2 =	manuálně (naprogramovaná hodnota)
				DESÍTKY MILIONŮ:	Faktor posuvu pro první přísuv <FRF> jako faktor/procentuální změna	
					0 =	kompatibilita, jako faktor (0,001 až 1,0, FRF = 0 znamená 100%)
					1 =	Procentuální změna (0,001 až 999,999%)

4.26.1.24 CYCLE84 - Vrtání závitů bez vyrovnávací hlavičky

Syntaxe

```
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>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	RP	<RTP>	REAL	Návratová rovina (abs)
2	Z0	<RFP>	REAL	Vztažný bod (abs)
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)
4	Z1	<DP>	REAL	Vrtaná hloubka = konečná vrtaná hloubka (abs), viz <_AMODE>
5	Z1	<DPR>	REAL	Vrtaná hloubka = konečná vrtaná hloubka (ink), viz <_AMODE>
6	DT	<DTB>	REAL	Doba prodlevy na vrtané hloubce v sekundách
7	SDE	<SDAC>	INT	Směr otáčení po skončení cyklu
8		<MPIT>	REAL	Velikost závitu, pouze pro "ISO metrický" (stoupání se vypočítává interně v průběhu zpracovávání programu)
9	P	<PIT>	REAL	Stoupání závitu jako hodnota, rozměrové jednotky viz <_PITA>
10	αS[1)]	<POSS>	REAL	Poloha pro orientované zastavení vřetena
11	S	<SST>	REAL	Otáčky vřetena pro vrtání závitu
12	SR	<SST1>	REAL	Otáčky vřetena pro zpětný pohyb

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam			
13		<_AXN>	INT	Osa vrtání	0 =	3. Geometrická osa	
					1 =	1. Geometrická osa	
					2 =	2. Geometrická osa	
					≥ 3 =	3. Geometrická osa	
14		<_PITA>	INT	Rozměrové jednotky pro stoupání závitu (vyhodnocování <PIT> a <MPIT>)			
					0 =	stoupání v mm	- vyhodnocování <MPIT>/<PIT>
					1 =	stoupání v mm	- vyhodnocování <PIT>
					2 =	stoupání v TPI	- vyhodnocování <PIT> (počet chodů závitu na palec)
					3 =	stoupání v palcích	- vyhodnocování <PIT>
					4 =	MODUL	- vyhodnocování <PIT>
15		<_TECHNO>	INT	Technologie[1)]			
				JEDNOTKY:	Režim přesného najetí		
					0 =	je aktivní chování přesného najetí, jaké bylo před voláním cyklu	
					1 =	Přesné najetí G601	
					2 =	Přesné najetí G602	
					3 =	Přesné najetí G603	
				DESÍTKY:	Dopředná regulace		
					0 =	s/bez dopředné regulace, jak bylo aktivní před voláním cyklu	
					1 =	s dopřednou regulací FFWON	
					2 =	bez dopředné regulace FFWOF	
				STOVKY:	Zrychlení		
					0 =	SOFT/BRISK/DRIVE, jak bylo aktivní před voláním cyklu	
					1 =	s omezením ryvu SOFT	
					2 =	bez omezení ryvu BRISK	
					3 =	snížené zrychlení, DRIVE	
				TISÍCE:	Režim vřetena při MCALL		
					0 =	s příkazem MCALL se opět aktivuje režim vřetena	
					1 =	s příkazem MCALL zůstat v režimu polohové regulace	

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
16		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:	0 =	1 řez
					1 =	ulamování třísky (vrtání závitů v hlubokých dírách)
					2 =	odstraňování třísek (vrtání závitů v hlubokých dírách)
				TISÍCE:	Režim ISO/SIEMENS pro vstupní obrazovku není relevantní	
					0 =	vyvolávání na základě kompatibility s ISO
					1 =	vyvolávání na základě kontextu firmy SIEMENS
17	D	<_DAM>	REAL	Maximální přísuv do hloubky (jen při odstraňování třísky/ulamování třísky)		
18	V2	<_VRT>	REAL	Vzdálenost zpětného pohybu po každém obrobení (jen při ulamování třísky) viz <_AMODE>		
19		<_PITM>	STRING[15]	Řetězec jako ukazatel pro zadávání stoupání závitu[2)]		
20		<_PTAB>	STRING[5]	Řetězec pro tabulku závitů ("", "ISO", "BSW", "BSP", "UNC")[2)]		
21		<_PTABA>	STRING[20]	Řetězec pro volbu v tabulce závitů (např. "M 10", "M 12", ...)[2)]		
22		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	rezervováno	
23		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	rezervováno	
				STOVKY:	rezervováno	
				TISÍCE:	Režim kompatibility (jen pro vstupní obrazovku pro zpětný překlad), pokud je MD 52216 Bit0 = 1[1)]	
					0 =	technologické parametry se vypisují (kompatibilita): parametr TECHNO je v platnosti
					1 =	technologické parametry se nevypisují: platí technologie, jaká byla naprogramována před voláním cyklu
				DESÍTKY TISÍC:	Odstupňování technologie na obrazovkách cyklů (Strana 1169)	
					0 =	Zadání: kompletní
					1 =	Zadání: jednoduché

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
24		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Vrtaná hloubka = konečná vrtaná hloubka Z1 (abs/ink)	
					0 =	Kompatibilita, podle naprogramování <DP>/<DPR>
					1 =	inkrementální
					2 =	absolutní
				DESÍTKY:	rezervováno	
				STOVKY:	rezervováno	
				TISÍCE:	směr otáčení závitu pravý/levý	
					0 =	kompatibilita, vyplývá ze znaménka parametru PIT/MPIT
					1 =	vpravo
					2 =	vlevo
				DESÍTKY TISÍC:	rezervováno	
				STOVKY TISÍC:	rezervováno	
				JEDNOTKY MILIONŮ:	Vzdálenost zpětného pohybu po každém obrobení V2 manuálně/automaticky	
					0 =	kompatibilita, podle naprogramování parametru <_VRT> (> 0 proměnná hodnota nebo ≤ 0 standardní hodnota 1 mm/ 0,0394 palce)
					1 =	automaticky (standardní hodnota 1 mm/ 0,0394 palce)
					2 =	manuálně (jak bylo naprogramováno pro V2)

[1] V závislosti na nastavovaném parametru SD52216 $MCS_FUNCTION_MASK_DRILL může být zobrazování polí technologie blokováno.

[2] Parametry 19, 22 a 21 se používají pouze při volbě závitu v tabulce závitů ve vstupní obrazovce. Přístup k tabulkám závitů prostřednictvím definice cyklů v době, kdy jsou cykly zpracovávány, není možný.

4.26.1.25 CYCLE85 - Vystružování

Syntaxe

```
CYCLE85(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <FFR>, <RFF>,
<_GMODE>, <_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	RP	<RTP>	REAL	Návratová rovina (abs)
2	Z0	<RFP>	REAL	Vztažný bod (abs)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
4	Z1	<DP>	REAL	Vrtaná hloubka (abs), viz <_AMODE>		
5	Z1	<DPR>	REAL	Vrtaná hloubka (ink), viz <_AMODE>		
6	DT	<DTB>	REAL	Doba prodlevy na konečné vrtané hloubce, viz <_AMODE>		
7	F	<FFR>	REAL	Posuv		
8	FR	<RFF>	REAL	Posuv při zpětném pohybu		
9		<_GMODE>	INT	rezervováno		
10		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
11		<_AMODE>	INT	Alternativní režim (vrtání)		
				JEDNOTKY:	Vrtaná hloubka Z1 (abs/ink)	
					0 =	kompatibilita, podle naprogramování DP/DPR
					1 =	inkrementální
					2 =	absolutní
				DESÍTKY:	Doba prodlevy DT na konečné vrtané hloubce v sekundách/otáčkách	
					0 =	kompatibilita, vyplývá ze znaménka DT (> 0 sekundy nebo < 0 otáčky)
					1 =	v sekundách
					2 =	v otáčkách

4.26.1.26 CYCLE86 - Vyvrtávání

Syntaxe

```
CYCLE86(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDIR>, <RPA>,
<RPO>, <RPAP>, <POSS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	RP	<RTP>	REAL	Návratová rovina (abs)
2	Z0	<RFP>	REAL	Vztažný bod (abs)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
4	Z1	<DP>	REAL	Vrtaná hloubka (abs), viz <_AMODE>		
5	Z1	<DPR>	REAL	Vrtaná hloubka (ink), viz <_AMODE>		
6	DT	<DTB>	REAL	Doba prodlevy na konečné vrtané hloubce, viz <_AMODE>		
7	DIR	<SDIR>	INT	Otáčení vřetena	3 =	M3
					4 =	M4
8	DX	<RPA>	REAL	Vzdálenost pro pozvednutí ve směru X		
9	DY	<RPO>	REAL	Vzdálenost pro pozvednutí ve směru Y		
10	DZ	<RPAP>	REAL	Vzdálenost pro pozvednutí ve směru Z		
11	SPOS	<POSS>	REAL	Poloha vřetena pro pozvednutí (pro orientované zastavení vřetena, ve stupních)		
12		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	Způsob pozvednutí nástroje	
					0 =	pozvednutí, kompatibilita
					1 =	nepozvednout
13		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
14		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Vrtaná hloubka Z1 (abs/ink)	
					0 =	Kompatibilita, podle naprogramování <DP>/<DPR>
					1 =	inkrementální
					2 =	absolutní
				DESÍTKY:	Doba prodlevy na konečné vrtané hloubce DT v sekundách/otáčkách	
					0 =	kompatibilita, vyplývá ze znaménka DT (> 0 sekundy nebo < 0 otáčky)
					1 =	v sekundách
					2 =	v otáčkách

4.26.1.27 CYCLE92 - Upichování

Syntaxe

```
CYCLE92(<_SPD>, <_SPL>, <_DIAG1>, <_DIAG2>, <_RC>, <_SDIS>, <_SV1>,
<_SV2>, <_SDAC>, <_FF1>, <_FF2>, <_SS2>, <_DIAGM>, <_VARI>, <_DN>,
<_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	X0	<_SPD>	REAL	Souřadnice vztažného bodu, (abs, vždycky průměr)		
2	Y0	<_SPL>	REAL	Vztažný bod (abs)		
3	X1	<_DIAG1>	REAL	Hloubka pro snížení otáček, viz parametr <_AMODE> (JEDNOTKY)		
4	X2	<_DIAG2>	REAL	Konečná hloubka, viz parametr <_AMODE> (DESÍTKY)		
5	R/FS	<_RC>	REAL	Rádius zaoblení nebo šířka fasety, viz <_AMODE> (TISÍCE)		
6	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
7	S	<_SV1>	REAL	Konstantní otáčky vřetena, viz <_AMODE> (DESÍTKY TISÍC)		
	V			konstantní řezná rychlost		
8	SV	<_SV2>	REAL	Maximální otáčky při konstantní řezné rychlosti		
9	DIR	<_SDAC>	INT	Otáčení vřetena	3 =	pro M3
					4 =	pro M4
10	F	<_FF1>	REAL	Posuv až do hloubky pro snížení otáček		
11	FR	<_FF2>	REAL	Snížená hodnota posuvu až do konečné hloubky		
12	SR	<_SS2>	REAL	Snížená hodnota otáček až do konečné hloubky		
13	XM	<_DIAGM>	REAL	Hloubka pro vyjíždění podavače součástek (abs, vždycky průměr)		
14		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:	Návratová dráha	
					0 =	Zpětný pohyb na rovinu <_SPD> + <_SDIS>
					1 =	žádný zpětný pohyb na konci
				DESÍTKY:	Podavač obrobků	
					0 =	ne, nevyvolávat žádný M-příkaz
					1 =	ano, vyvolání funkce CUST_TECHCYC(101) - vyjíždění zásuvky, CUST_TECHCYC(102) - zasunutí zásuvky
15		<_DN>	INT	D-číslo pro 2. břit nástroje, pokud není naprogramováno ⇒ D+1		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
20		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
21		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Hloubka pro snížení otáček (<_DIAG1>)	
					0 =	absolutně, hodnota příčné osy se zadává jako průměr
					1 =	inkrementálně, hodnota příčné osy se zadává jako rádius
				DESÍTKY:	Konečná hloubka (<_DIAG2>)	
					0 =	absolutně, hodnota příčné osy se zadává jako průměr
					1 =	inkrementálně, hodnota příčné osy se zadává jako rádius
				STOVKY:	rezervováno	
				TISÍCE:	Rádius / faseta (<_RC>)	
					0 =	Rádius
					1 =	Faseta
				DESÍTKY TISÍC:	Otáčky vřetena/řezná rychlost (<_SV1>)	
					0 =	konstantní otáčky vřetena
					1 =	konstantní řezná rychlost

4.26.1.28 CYCLE95 - Oddělování třísky od kontury

Syntaxe

```
CYCLE95(<NPP>, <MID>, <FALZ>, <FALX>, <FAL>, <FF1>, <FF2>, <FF3>,
<_VARI>, <DT>, <DAM>, <_VRT>, <_GMODE>, <_DMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	CON	<NPP>	STRING [140]	Název kontury
2	D	<MID>	REAL	Maximální přísuv do hloubky při obrábění nahrubo, viz <_GMODE>
3	UZ	<FALZ>	REAL	Přídavek rozměru pro opracování načisto ve směru Z
4	UX	<FALX>	REAL	Přídavek rozměru pro opracování načisto ve směru X

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
5	U	<FAL>	REAL	Přídavek rozměru pro opracování načisto rovnoběžně s konturou (uplatňuje se u obou os)		
6	F	<FF1>	REAL	Posuv pro obrábění nahrubo		
7	FY	<FF2>	REAL	Přísuv při opracovávání podříznutí		
8	FS	<FF3>	REAL	Posuv pro obrábění načisto		
9		<_VARI>	INT	Způsob opracování		
				JEDNOTKY a DESÍTKY		
					1 =	Obrábění nahrubo v podélném směru zvnějšku
					2 =	Obrábění nahrubo v příčném směru zvnějšku
					3 =	Obrábění nahrubo v podélném směru zevnitř
					4 =	Obrábění nahrubo v příčném směru zevnitř
					5 =	Obrábění načisto v podélném směru zvnějšku
					6 =	Obrábění načisto v příčném směru zvnějšku
					7 =	Obrábění načisto v podélném směru zevnitř
					8 =	Obrábění načisto v příčném směru zevnitř
					9 =	Kompletní opracování v podélném směru zvnějšku
					10 =	Kompletní opracování v příčném směru zvnějšku
					11 =	Kompletní opracování v podélném směru zevnitř
					12 =	Kompletní opracování v příčném směru zevnitř
				STOVKY:		
					0 =	s vyjížděním podél kontury, bez růžků zbytkového materiálu
					1 =	bez vyjíždění podél kontury
					2 =	vyjíždění podél kontury jen k předcházejícímu průsečíku, mohou vznikat růžky zbytkového materiálu
10	DT	<DT>	REAL	Doba prodlevy při přerušení posuvu		
11	DI	<DAM>	REAL	Vzdálenost přerušení posuvu		
12	VRT	<_VRT>	REAL	Dráha pozvednutí od kontury		
					0 =	interně se bude používat pozvednutí o 1 mm nezávisle na systému jednotek (palce/mm)
					> 0 =	dráha pozvednutí nástroje

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
13		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	Vyhodnocování přísuvné hloubky	
					0 =	Přísuvná hloubka se započítává v souladu s příkazy G-skupiny DIAMON/DIAMOF
					1 =	Přísuvná hloubka se uplatňuje jako hodnota rádiusu (nezávisle na DIAMON/DIAMOF)
14		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				TISÍCE:		
					0 =	Režim kompatibility název kontury je uložen v NPP
					1 =	název kontury je naprogramován v cyklu CYCLE62 a předává se v parametru _SC_CONT_NAME

4.26.1.29 CYCLE98 - Řetězec závitů

Syntaxe

```
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>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	Z0	<_PO1>	REAL	Souřadnice Z vztažného bodu (abs)
2	X0	<_DM1>	REAL	Souřadnice X vztažného bodu, (abs), jako průměr
3	Z1	<_PO2>	REAL	Vnitřní bod 1, osa Z (abs/ink), viz <_AMODE> (JEDNOTKY)
4	X1	<_DM2>	REAL	Vnitřní bod 1, osa X (abs/ink), viz <_AMODE> (DESÍTKY) nebo
	X1α			Úhel šikmého závitu 1 (-90° až 90°) abs vždy jako průměr, ink vždy jako rádius
5	Z2	<_PO3>	REAL	Vnitřní bod 2, osa Z (abs/ink), viz <_AMODE> (STOVKY)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
6	X2	<_DM3>	REAL	Vnitřní bod 2, osa X (abs/ink), viz <_AMODE> (TISÍCE) nebo		
	X2α			Úhel šikmého závitu 2 (-90° až 90°) abs vždy jako průměr, ink vždy jako rádius		
7	Z3	<_PO4>	REAL	Koncový bod, osa Z (abs/ink), viz <_AMODE> (DESÍTKY TISÍC)		
8	X3	<_DM4>	REAL	Koncový bod, osa X (abs/ink), viz <_AMODE> (STOVKY TISÍC) nebo		
	X3α			Úhel šikmého závitu 3 (-90° až 90°) abs vždy jako průměr, ink vždy jako rádius		
9	LW	<APP>	REAL	Předsunutý náběh závitu (ink, zadává se bez znaménka)		
10	LR	<ROP>	REAL	Výběh závitu (ink, zadává se bez znaménka)		
11	H1	<TDEP>	REAL	Hloubka závitu (ink, zadává se bez znaménka)		
12	U	<FAL>	REAL	Přídavek rozměru pro opracování načisto ve směrech X a Z		
13	DP	<_IANG>	REAL	Šikmý přísuv jako vzdálenost nebo úhel, viz <_AMODE> (JEDNOTKY MILIONŮ)		
	αP			Šikmý přísuv se uplatňuje v souladu s nastavením parametru <_VARI> (STOVKY)		
					Definice pro parametr<_VARI_HUNDERTER = 0 - režim kompatibility:	
					> 0 =	přísuv po jedné stěně
					0 =	přísuv kolmo do závitu
					< 0 =	přísuv střídavě po obou stěnách
					Definice pro parametr _VARI (STOVKY) <> 0:	
					> 0 =	přísuv po kladné stěně
					0 =	přísuv středem
					< 0 =	přísuv po záporné stěně
14	α0	<NSP>	REAL	Úhlové posunutí počátečního bodu pro 1. chod závitu		
15		<NRC>	INT	Počet průchodů nástroje při obrábění nahrubo, viz <_VARI> (DESÍTKY TISÍC)		
16	NN	<NID>	INT	Počet průchodů nástroje naprázdno		
17	P0	<_PP1>	REAL	Stoupání závitu pro 1. úsek závitu, viz <_PITA>		
18	P1	<_PP2>	REAL	Stoupání závitu pro 2. úsek závitu, viz <_PITA>		
19	P2	<_PP3>	REAL	Stoupání závitu pro 3. úsek závitu, viz <_PITA>		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
20		<_VARI>	INT	Opracování		
				JEDNOTKY:	Technologie	
					1 =	vnější závit s lineárním přísuvem
					2 =	vnitřní závit s lineárním přísuvem
					3 =	vnější závit s degresivním přísuvem, průřez třísky zůstává konstantní
					4 =	vnitřní závit s degresivním přísuvem, průřez třísky zůstává konstantní
				DESÍTKY:	rezervováno	
				STOVKY:	Způsob přísuvu	
					0 =	Režim kompatibility pro <_IANG>
					1 =	přísuv na jedné straně
					2 =	přísuv střídavě na obou stranách
				TISÍCE:	rezervováno	
				DESÍTKY TISÍC:	Alternativní přísuv do hloubky	
					0 =	kompatibilita, zadání počtu průchodů nástroje při hrubování (<_NRC>)
					1 =	předdefinovaná hodnota pro 1. přísuv (<_MID>)
				STOVKY TISÍC:	Způsob opracování	
					0 =	Kompatibilita (obrábění nahrubo a obrábění načisto)
					1 =	Obrábění nahrubo
					2 =	Obrábění načisto
					3 =	Obrábění nahrubo a obrábění načisto
				JEDNOTKY MILIONŮ:	Posloupnost opracování u vícechodých závitů	
					0 =	vzestupné pořadí chodů
					1 =	obrácené pořadí chodů
21	N	<_NUMTH>	INT	Počet chodů závitu		
22		<_VRT>	REAL	Návratová vzdálenost (ink)		
					0 =	interně se bude používat pozvednutí o 1 mm nezávisle na systému jednotek (palce/mm)
					> 0 =	dráha pozvednutí nástroje
23	D1	<_MID>	REAL	První přísuv, viz <_VARI> (DESÍTKY TISÍC)		
24	DA	<_GDEP>	REAL	Střídavá hloubka chodů závitu (uplatňuje se jen u nastavení "vícechodý")		
					0 =	na střídavou hloubku chodů závitu nebrat ohled
					> 0 =	na střídavou hloubku chodů závitu brát ohled

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
25		<_IFLANK>	REAL	Šikmý přísuv jako šířka (jen pro pracovní plochu)		
26		<_PITA>	INT	Vyhodnocování stoupání závitu		
					0 =	Režim kompatibility pro stoupání závitu, vyhodnocování <_PP1> až <_PP3> jako dříve v souladu s aktivním systémem měřicích jednotek (metrický/palce)
					1 =	stoupání v mm
					2 =	stoupání v TPI (počet chodů závitu na palec)
					3 =	stoupání v palcích
					4 =	MODUL
27		<_PITM1>	STRING[15]	Řetězec jako ukazatel pro zadávání stoupání závitu (jen pro plochu)		
28		<_PITM2>	STRING[15]	Řetězec jako ukazatel pro zadávání stoupání závitu (jen pro plochu)		
29		<_PITM3>	STRING[15]	Řetězec jako ukazatel pro zadávání stoupání závitu (jen pro plochu)		
30		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	---	rezervováno
				STOVKY:	---	rezervováno
				TISÍCE:	---	rezervováno
				DESÍTKY TISÍC:	Odstupňování technologie na obrazovkách cyklů (Strana 1169)	
					0 =	Zadání: kompletní
					1 =	Zadání: jednoduché

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
31		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	1. Vnitřní bod, souřadnice Z (Z1)	
					0 =	absolutní
					1 =	inkrementální
				DESÍTKY:	1. Vnitřní bod, souřadnice X (X1)	
					0 =	absolutní
					1 =	inkrementální
					2 =	α
				STOVKY:	2. Vnitřní bod, souřadnice Z (Z2)	
					0 =	absolutní
					1 =	inkrementální
				TISÍCE:	2. Vnitřní bod, souřadnice X (X2)	
					0 =	absolutní
					1 =	inkrementální
					2 =	α
				DESÍTKY TISÍC:	koncový bod v ose Z (Z3)	
					0 =	absolutní
					1 =	inkrementální
				STOVKY TISÍC:	koncový bod, souřadnice X (X3)	
					0 =	absolutní
					1 =	inkrementální
					2 =	α
				JEDNOTKY MILIONŮ:	Volba šikého přísuvu jako úhlu nebo šířky	
					0 =	Úhel přísuvu <_IANG>
					1 =	Šikmá přísuvná plocha <_IFLANK>
				DESÍTKY MILIONŮ:	jednochový/vícechodý	
					0 =	režim kompatibility (počáteční úhel <_NSP> se vyhodnocuje)
					1 =	jednochový (s úhlovým posunutím počátečního bodu <_NSP>)
					2 =	vícechodý

4.26.1.30 CYCLE99 - Soustružení závitu

Syntaxe

```
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>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	Z0	<_SPL>	REAL	Vztažný bod (abs)		
2	X0	<_SPD>	REAL	Souřadnice vztažného bodu, (abs, vždycky průměr)		
3	Z1	<_FPL>	REAL	Koncový bod ve spojení s parametrem <_AMODE> (JEDNOTKY)		
4	X1	<_FPD>	REAL	Koncový bod ve spojení s parametrem <_AMODE> (DESÍTKY)		
5	LW/LW2	<_APP>	REAL	Náběh závitu ve spojení s parametrem <_AMODE> (STOVKY) nebo Náběh závitu = výběh závitu ve spojení s parametrem <_AMODE> (STOVKY)		
6	LR	<_ROP>	REAL	Výběh závitu		
7	H1	<_TDEP>	REAL	Hloubka závitu		
8	U	<_FAL>	REAL	Přídavek rozměru pro opracování načisto ve směrech X a Z		
9	DP	<_IANG>	REAL	Šikmý přísuv jako vzdálenost nebo úhel ve spojení s parametrem <_AMODE> (TISÍCE)		
	αP				> 0 =	přísuv po kladné stěně
					< 0 =	přísuv po záporné stěně
					0 =	přísuv středem
10	α0	<_NSP>	REAL	Úhlové posunutí počátečního bodu (uplatňuje se jen u nastavení "jednochodý")		
11	ND	<_NRC>	INT	Počet průchodů nástroje při obrábění nahrubo, ve spojení s <_VARI> (DESÍTKY TISÍC)		
12	NN	<_NID>	INT	Počet průchodů nástroje naprázdno		
13	P	<_PIT>	REAL	Stoupání závitu jako hodnota, ve spojení s parametrem <_PITA>		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
14		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:	Technologie	
					1 =	vnější závit s lineárním přísuvem
					2 =	vnitřní závit s lineárním přísuvem
					3 =	vnější závit s degresivním přísuvem, průřez třísky zůstává konstantní
					4 =	vnitřní závit s degresivním přísuvem, průřez třísky zůstává konstantní
				DESÍTKY:	rezervováno	
				STOVKY:	Způsob přísuvu	
					1 =	přísuv na jedné straně
					2 =	přísuv střídavě na obou stranách
				TISÍCE:	rezervováno	
				DESÍTKY TISÍC:	Alternativní přísuv do hloubky	
					0 =	zadání počtu průchodů nástroje při hrubování (<_NRC>)
					1 =	předdefinovaná hodnota pro 1. přísuv (<_MID>)
				STOVKY TISÍC:	Způsob opracování	
					1 =	Obrábění nahrubo
					2 =	Obrábění načisto
					3 =	Obrábění nahrubo a obrábění načisto
				JEDNOTKY MILIONŮ:	Posloupnost opracování u vícechodých závitů	
					0 =	vzestupné pořadí chodů
					1 =	obrácené pořadí chodů
15	N	<_NUMTH>	INT	Počet chodů závitu		
16	VR	<_SDIS>	REAL	Návratová vzdálenost, ink		
17	D1	<_MID>	REAL	První přísuvná hloubka, ve spojení s parametrem <_VARI> (DESÍTKY TISÍC)		
18	DA	<_GDEP>	REAL	Střídavá hloubka chodů závitu (uplatňuje se jen u nastavení "vícechodý")		
					0 =	na střídavou hloubku chodů závitu nebrat ohled
					> 0 =	na střídavou hloubku chodů závitu brát ohled
19	G	<_PIT1>	REAL	Změna stoupání na otáčku		
					0 =	stoupání závitu je konstantní (G33)
					> 0 =	stoupání závitu se zvětšuje (G34)
					< 0 =	stoupání závitu se zmenšuje (G35)
20		<_FDEP>	REAL	Hloubka zajíždění nástroje (zadává se bez znaménka)		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
21	N1	<_GST>	INT	Počáteční chod N1 = 1...N, ve spojení s parametrem <_AMODE> (STOVKY TISÍC)		
22		<_GUD>	INT	rezervováno		
23		<_IFLANK>	REAL	Šikmý přísuv jako šířka (jen pro pracovní plochu)		
24		<_PITA>	INT	Rozměrové jednotky pro stoupání závitu (vyhodnocování PIT a/nebo MPIT)		
					0 =	stoupání v mm - vyhodnocování MPIT/PIT
					1 =	stoupání v mm - vyhodnocování PIT
					2 =	stoupání v TPI - vyhodnocování PIT (počet chodů závitu na palec)
					3 =	stoupání v palcích - vyhodnocování PIT
					4 =	MODUL - vyhodnocování PIT
25		<_PITM>	STRING[15]	Řetězec jako ukazatel pro zadávání stoupání závitu (jen pro plochu)[1)]		
26		<_PTAB>	STRING[20]	Řetězec pro tabulku závitů (jen pro pracovní plochu)[1)]		
27		<_PTABA>	STRING[20]	Řetězec pro volbu v tabulce závitů (jen pro pracovní plochu)[1)]		
28		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	Druh závitu	
					0 =	Podélný závit
					1 =	Příčný závit
					2 =	Kuželový závit
				STOVKY:	---	rezervováno
				TISÍCE:	---	rezervováno
				DESÍTKY TISÍC:	Odstupňování technologie na obrazovkách cyklů (Strana 1169)	
					0 =	Zadání: kompletní
					1 =	Zadání: jednoduché

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
29		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Délka závitu ve směru Z	
					0 =	absolutní
					1 =	inkrementální
				DESÍTKY:	Délka závitu ve směru X	
					0 =	absolutně, hodnota příčné osy se zadává jako průměr
					1 =	inkrementálně, hodnota příčné osy se zadává jako rádius
					2 =	α
				STOVKY:	Vyhodnocování náběhu <_APP>	
					0 =	Předsunutý náběh závitu <_APP>
					1 =	Náběh závitu = výběh závitu <_APP> = -<_ROP>
					2 =	Je zadán náběh závitu <_APP> = -<_APP>
				TISÍCE:	Volba šikého přísuvu jako úhlu nebo šířky	
					0 =	Úhel přísuvu <_IANG>
					1 =	Šikmá přísuvná plocha <_IFLANK>
				DESÍTKY TISÍC:	jednochový/vícechodý	
					0 =	jednochový (s úhlovým posunutím počátečního bodu <_NSP>)
					1 =	vícechodý
				STOVKY TISÍC:	Počáteční chod <_GST>	
					0 =	kompletní opracování
					1 =	zahájit opracování od tohoto chodu
					2 =	opracovat jen tento chod
				JEDNOTKY MILIONŮ:	Kompenzace průhybu u podélných závitů	
					0 =	Výška segmentu sudovitého závitu XS
					1 =	Rádius sudovitého závitu RS
30	XS / RS	<_S_XRS>	REAL	Kompenzace průhybu u podélných závitů ve spojení s parametrem <_AMODE>: JEDNOTKY MILIONŮ		

Poznámka

[1)] Parametry <_PITM>, <_PTAB> a <_PTABA> se používají pouze při volbě závitu v tabulce závitů ve vstupní obrazovce.
Přístup k tabulkám závitů prostřednictvím definice cyklů v době, kdy jsou cykly zpracovávány, není možný.

4.26.1.31 CYCLE435 - Nastavení souřadného systému orovnávače

Syntaxe

```
CYCLE435(<_T>, <_DD>, <S_TA>, <S_DA>, <S_AD>, <S_AL>, <S_PVD>,
<S_PVL>, <S_PD>, <S_PL>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1		<_T>	STRING[32]	Název nástroje brusného kotouče		
2		<_DD>	INT	Číslo břitu brusného kotouče		
3		<S_TA>	STRING[32]	Vztažný bod orovnávače - název orovnávače		
4		<S_DA>	INT	Číslo břitu orovnávače		
5		<S_AD>	REAL	Hodnota orovnávání - průměr		
6		<S_AL>	REAL	Hodnota orovnávání - v příčné rovině		
7		<S_PVD>	REAL	Posunutí profilovacího nástroje - průměr		
8		<S_PVL>	REAL	Posunutí profilovacího nástroje - v příčné rovině		
9		<S_PD>	REAL	Přídavek rozměru pro profilování - průměr		
10		<S_PL>	REAL	Přídavek rozměru pro profilování - v příčné rovině		
11		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Aktivní nástroj na konci cyklu	
					0 =	Orovnávač aktivní
					1 =	Kotouč aktivní

4.26.1.32 CYCLE495 - Profilování

Syntaxe

```
CYCLE495(<_T>, <_DD>, <_SC>, <_F>, <_VARI>, <_D>, <_DX>, <_DZ>,
<S_PA>, <S_N>, <_DMODE>, <_AMODE>, <S_FW>, <S_HW>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1		<_T>	STRING[20]	Název nástroje brusného kotouče
2		<_DD>	INT	Číslo břitu brusného kotouče
3		<_SC>	REAL	Vzdálenost pro pozvednutí při objíždění překážky, inkrementálně
4		<_F>	REAL	Posuv při profilování

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
5		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:	Druh profilování	
					1 =	rovnoběžně s osami
					2 =	rovnoběžně s konturou
				DESÍTKY:	Směr obrábění	
					0 =	vytahování možné s polohami břitu 1 až 4
					1 =	tlakem možné s polohami břitu 1 až 4
					2 =	střídavě možné s polohami břitu 1 až 8
					3 =	začátek → konec možné s polohami břitu 1 až 8
					4 =	konec → začátek možné s polohami břitu 1 až 8
				STOVKY:	Směr přísuvu	
					1 =	Přísuv ve směru X- v případě roviny G18, příp. Y- v případě roviny G19
					2 =	Přísuv ve směru X+ v případě roviny G18, příp. Y+ v případě roviny G19
					3 =	Přísuv ve směru Z- v případě rovin G18 a G19
					4 =	Přísuv ve směru Z+ v případě rovin G18 a G19
6		<_D>	REAL	Orovnávaná hodnota v případě druhu profilování rovnoběžně s osami		
7		<_DX>	REAL	Orovnávaná hodnota ve směru X v případě roviny G18, příp. Y v případě roviny G19, při druhu profilování rovnoběžně s konturou		
8		<_DZ>	REAL	Orovnávaná hodnota ve směru Z v případě rovin G18 a G19, při druhu profilování rovnoběžně s konturou		
9		<S_PA>	REAL	Přídavek rozměru pro profilování		
10		<S_N>	INT	Počet zdvihů v profilovacím programu		
11		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
12		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Volba nové profilování/pokračování	
					1 =	Nový
					2 =	Pokračovat
				DESÍTKY:	Volba přídavku rozměru pro profilování	
					0 =	od kontury surového obrobku až po nejhlubší bod na kontuře
					1 =	od kontury surového obrobku až po nejvyšší bod na kontuře
13		<S_FW>	REAL	Úhel volného řezání orovnávače		
14		<S_HW>	REAL	Úhel držáku orovnávače		

Poznámka

Interpretace kontury při profilování pomocí cyklu CYCLE495 se provádí prostřednictvím předem definované procedury CONTDCON. Protože procedura CONTDCON je v době, kdy je aktivní korekce rádiusu nástroje (G41/G42), **nepřípustná**, musí být před voláním cyklu CYCLE495 korekce rádiusu nástroje deaktivována příkazem G40.

4.26.1.33 CYCLE782 - Přizpůsobení zatížení

Poznámka

Abyste mohli cyklus CYCLE782 používat, je zapotřebí licence pro volitelný doplněk „Inteligentní přizpůsobování zátěže".

Syntaxe

```
CYCLE782(<S_MODE>, <S_TESTAXIS>, <S_VALUE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1		<S_MODE>	INT	Režim zpracování		
				JEDNOTKY:	Aktivování/deaktivování	
					0 =	Deaktivování přizpůsobování
					1 =	Aktivování přizpůsobování
					2 =	Zjištění a aktivování základní setrvačnosti dané osy (funkce pro uvádění do provozu)
				DESÍTKY:	Varianta měření (jen tehdy, je-li <S_MODE> MÍSTO STOVEK = 0)	
					0 =	Normální režim
					1 =	Režim přesného měření (např. s vyhodnocováním tření atd.).
				STOVKY	Měření nebo zadání hodnoty setrvačnosti	
					0 =	Měření V tomto režimu jsou uskutečňovány pohybové operace sloužící pro zjištění zatížení.
					1 =	Zadání hodnoty setrvačnosti V tomto režimu nejsou prováděny žádné pohybové operace sloužící pro zjištění zatížení. Místo toho se předává dříve zjištěná, např. prostřednictvím automatické optimalizace servomechanismu (AST), hodnota setrvačnosti.
				TISÍCE	Vypisování výsledků měření	
					0 =	Zrušení zobrazování obrazovky s výsledky měření
					1 =	Aktivování zobrazování obrazovky s výsledky měření
				DESÍTKY TISÍC	Doba trvání zobrazování výsledků měření	
					1 =	Automatické deaktivování zobrazování obrazovky po 8 s
					3 =	Potvrzování pomocí tlačítka NC-Start
				STOVKY TISÍC	Volba zatížení [1]	
					0 =	Zjištění celkového zatížení: Osa samotná + obrobek
					1 =	Zjištění samotného zatížení: Obrobek
2		<S_TESTAXIS>	AXIS	Kanálová osa: • má být aktivována pro přizpůsobování • pro kterou má být zjištěna setrvačnost (jen tehdy, je-li <S_MODE> MÍSTO STOVEK = 0)		
3		<S_VALUE>	REAL	Hodnota setrvačnosti (jen tehdy, je-li <S_MODE> MÍSTO STOVEK = 1)		

[1] Hodnota se automaticky převezme ze strojního parametru 52212 $MCS_FUNCTION_MASK_TECH, bit 18.

Poznámka

Funkce pro uvádění do provozu „Zjištění a aktivování základní setrvačnosti osy" (<S_MODE> MÍSTO JEDNOTEK = 2) není uživatelským rozhraním podporována. Pro tento účel musí být naprogramováno samostatné volání cyklu. Výsledek měření se zapisuje do strojního parametru 53350 $MAS_ILC_BASE_VALUE.

Hodnota parametru „Volba zatížení" (<S_MODE> MÍSTO STOVEK TICÍCŮ) se z uživatelského rozhraní načítá ze strojního parametru 52212 $MCS_FUNCTION_MASK_TECH, bit 18, a zapisuje se v rámci volání cyklu.

Zatížení může být zjišťováno jak pro lineární, tak i pro kruhové osy nebo může být zadáváno předem. Pro lineární osy se na obrazovce výsledků měření vždy vypisuje, případně se předem zadává na uživatelském rozhraní, hmotnost a pro kruhové osy vždy setrvačnost.

Nezávisle na typu osy se do strojního parametru 53350 $MAS_ILC_BASE_VALUE a do obrazovky Přizpůsobení vždy zapisuje setrvačnost.

Příklady

Příklad 1:

Na začátku NC programu má být zjištěno zatížení osy MX1 a podle něho mají být aktualizována přizpůsobování. Výsledek se má zobrazit a zobrazovaný údaj má být potvrzen tlačítkem NC-Start.

Programový kód

```
...
CYCLE782(31011,MX1)
...
```

Příklad 2:

Hodnota setrvačnosti zjištěná např. pomocí procedury AST má být v průběhu zpracování programu přenesena do proměnné _MY_VALUE, s touto hodnotou má být vyvolán cyklus pro zjištění zatížení a se zjištěnými hodnotami má být aktivováno přizpůsobení pro ose MX2. Nemají se vypisovat žádné výsledky.

Programový kód

```
DEF REAL _MY_VALUE
_MY_VALUE=...
...
CYCLE782(101,MX2,_MY_VALUE)
...
```

Příklad 3:

Přizpůsobení pro osu MX2 mají být deaktivována.

Programový kód

```
...
CYCLE782(0,MX2)
```

```
Programový kód
...
```

4.26.1.34 CYCLE800 - Naklápění roviny / naklápění nástroje / nastavování orientace nástroje

Syntaxe

```
CYCLE800(<_FR>, <_TC>, <_ST>, <_MODE>, <_X0>, <_Y0>, <_Z0>, <_A>,
<_B>, <_C>, <_X1>, <_Y1>, <_Z1>, <_DIR>, <_FR_I>, <_DMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1		<_FR>	INT	Režim volného vyjíždění:	0 =	žádné volné vyjíždění
					1 =	volné vyjíždění strojní osy Z
					2 =	volné vyjíždění strojní osy Z a potom XY
					3 =	rezervováno
					4 =	Maximální volné vyjíždění ve směru nástroje
					5 =	Inkrementální volné vyjíždění ve směru nástroje
2		<_TC>	STRING[32]	Název datového bloku naklápění:	""	"" (žádný název), pokud existuje jen 1 datový blok naklápění
					"0"	deaktivování datového bloku naklápění (vymazání framu naklápění)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
3		<_ST>	INT	Status transformací		
				JEDNOTKY:		
					0 =	nová, naklopená rovina se vymaže a znovu se vypočítá s aktuálními parametry
					1 =	aditivní, naklopená rovina se přičte k aktivní naklopené rovině
				DESÍTKY:	Špička nástroje na jednom místě ano/ne (aktivní jen tehdy, je-li instalována funkce IBN SCHWENKEN)	
					0 =	špička nástroje nezůstává na jednom místě
					1 =	špička nástroje zůstává na jednom místě (TRAORI)
				STOVKY:	Nastavování polohy/orientace (funkce se zobrazuje ve vstupní obrazovce NAKLÁPĚNÍ)	
					0 =	nástroj se nenastavuje
					1 =	nástroj se nastavuje (zejména fréza)
					2 =	nastavit orientaci **soustružnického** nástroje (pokud je ve funkci IBN SCHWENKEN instalována kinematika osy B pro technologii soustružení)
					3 =	nastavit orientaci **frézovacího** nástroje (pokud je ve funkci IBN SCHWENKEN instalována kinematika osy B pro technologii soustružení)
				TISÍCE:	Interní parametr naklápění v režimu JOG	
				DESÍTKY TISÍC:	Viz parametr pro směr <_DIR>	
					0 =	Naklápění „ano“
					1 =	naklápění "ne", směr "mínus" [3]
					2 =	naklápění "ne", směr "plus" [3]
				STOVKY TISÍC:	Viz parametr pro směr <_DIR>	
					0 =	Kompatibilita
					1 =	Volba směru "mínus" optimalizována (jen pro uživatelské rozhraní) [4]
					2 =	Volba směru „plus" optimalizována (jen pro uživatelské rozhraní) [4]

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
4		<_MODE> [5]	INT	Režim naklápění: Vyhodnocování úhlu naklápění a posloupnosti naklápění (bitové kódování!)		
				Bit: 7 6	0 0:	Úhel naklopení po osách --> viz parametry <_A>, <_B>, <_C>
					0 1:	Prostorový úhel --> viz parametry <_A>, <_B> [1]
					1 0:	Projekční úhel --> viz parametry <_A>, <_B>, <_C> [1]
					1 1:	Režim naklápění, kruhové osy přímo --> viz parametry <_A>, <_B> [1]
				Bit: 5 4 3 2 1 0 (v případě prostorového úhlu nemají žádný význam!)	x x x x 0 1	1. otáčení _A okolo osy X
					x x x x 1 0	1. otáčení _A okolo osy Y
					x x x x 1 1	1. otáčení _A okolo osy Z
					x x 0 1 x x	2. otáčení _B okolo osy X
					x x 1 0 x x	2. otáčení _B okolo osy Y
					x x 1 1 x x	2. otáčení _B okolo osy Z
					0 1 x x x x	3. otáčení _C okolo osy X
					1 0 x x x x	3. otáčení _C okolo osy Y
					1 1 x x x x	3. otáčení _C okolo osy Z
5	X0	<_X0>	REAL	Souřadnice X vztažného bodu před otáčením		
6	Y0	<_Y0>	REAL	Souřadnice Y vztažného bodu před otáčením		
7	Z0	<_Z0>	REAL	Souřadnice Z vztažného bodu před otáčením		
8	X(A)	<_A>	REAL	1. Otáčení podle nastavení v parametru <_MODE>		
9	Y(B)	<_B>	REAL	2. Otáčení podle nastavení v parametru <_MODE>		
10	Z(C)	<_C>	REAL	3. Otáčení podle nastavení v parametru <_MODE>		
11	X1	<_X1>	REAL	Souřadnice X vztažného bodu po otáčení		
12	Y1	<_Y1>	REAL	Souřadnice Y vztažného bodu po otáčení		
13	Z1	<_Z1>	REAL	Souřadnice Z vztažného bodu po otáčení		
14	- nebo +	<_DIR>	INT	Spustit pohyb kruhových os pracovním posuvem (předdefinovaná hodnota = -1!)		
					-1 =	najíždět na menší hodnotu kruhové osy 1 nebo 2 [2]
					+1 =	najíždět na větší hodnotu kruhové osy 1 nebo 2 [2]
					0 =	žádné naklápění (jen vypočítat frame naklápění) [1] [3]
15	FR	<_FR_I>	REAL	Hodnota (ink): inkrementální volné vyjíždění ve směru nástroje		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
16		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	Zobrazování hodnot úhlu beta při nastavování orientace nástroje	
					0 =	Hodnota
					1 =	Šipka

Poznámka

Pokud jsou následující předávané parametry naprogramovány nepřímo (jako parametry), zpětný překlad vstupní obrazovky nebude možný: <_FR>, <_ST>, <_TC>, <_MODE>, <_DIR>

[1)] Volba je možná, pokud je instalována funkce IBN SCHWENKEN.

[2)] Volba je možná, pokud je ve funkci IBN SCHWENKEN nastaven vztažný směr kruhové osy 1 nebo 2.

žádné pole pro volbu vztažného směru: není

[3)] Zobrazování volby naklápění "ne" může být blokováno (SD 55221 Bit 0).

Naklápění "ne" ve směnu "mínus" odpovídá <_DIR> = 0 a _ST DESÍTKY TISÍC = 1

Naklápění "ne" ve směnu "plus" odpovídá <_DIR> = 0 a _ST DESÍTKY TISÍC = 2

[4)] Volba směru kruhové osy 1 nebo 2 se uskutečňuje také tehdy, když se kruhová osa se vztažným směrem nachází v poloze pólu (hodnota polohy se rovná nule).

[5)] Příklad kódování: otáčení po osách, posloupnost otáčení ZYX

binárně: 00011011; decimálně: 27

Identifikátory os XYZ odpovídají geometrickým osám NC kanálu. Otáčení okolo os XYZ se smí uskutečňovat jednotlivě. Např. posloupnost otáčení ZXZ je ve volání cyklu CYCLE800 nepřípustná.

4.26.1.35 CYCLE801 - Polohovací vzor Mřížka nebo Obdélník

Syntaxe

```
CYCLE801(<_SPCA>, <_SPCO>, <_STA>, <_DIS1>, <_DIS2>, <_NUM1>,
<_NUM2>, <_VARI>, <_UMODE>, <_ANG1>, <_ANG2>, <_HIDE>, <_NSP>,
<_DMODE>)
```

Parametr

Č.	Obrazovka parametrů	Parametry interní	Datový typ	Význam		
1	X0	<_SPCA>	REAL	Vztažný bod pro polohovací vzor (mřížka/obdélník) na 1. ose (abs)		
2	Y0	<_SPCO>	REAL	Vztažný bod pro polohovací vzor (mřížka/obdélník) na 2. ose (abs)		
3	α0	<_STA>	REAL	Základní úhel otočení (úhel vzhledem k 1. ose)	< 0 =	Otáčení ve směru hodinových ručiček
					> 0 =	Opisování proti směru hodinových ručiček
4	L1	<_DIS1>	REAL	Vzdálenost mezi sloupci (vzdálenost mezi pozicemi na 1. ose, zadává se bez znaménka)		
5	L2	<_DIS2>	REAL	Vzdálenost řádků (vzdálenost mezi pozicemi na 2. ose, zadává se bez znaménka)		
6	N1	<_NUM1>	INT	Počet sloupců		
7	N2	<_NUM2>	INT	Počet řádků		
8		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:	Polohovací vzor	
					0 =	Mřížka
					1 =	Obdélník
				DESÍTKY:	rezervováno	
				STOVKY:	rezervováno	
9		<_UMODE>	INT	rezervováno		
10	αX	<_ANG1>	REAL	Úhel skosení vzhledem k 1. ose (zešikmení řádků vzhledem k 1. ose)		
					< 0 =	měření ve směru hodinových ručiček (0 až -90 stupňů)
					> 0 =	měření proti směru hodinových ručiček (0 až 90 stupňů)
11	αY	<_ANG2>	REAL	Úhel skosení vzhledem k 2. ose (zešikmení sloupců vzhledem ke 2. ose)		
					< 0 =	měření ve směru hodinových ručiček (0 až -90 stupňů)
					> 0 =	měření proti směru hodinových ručiček (0 až 90 stupňů)
12		<_HIDE>	STRING [200]	Přeskakované pozice • max. 198 znaků • Údaj pořadového čísla pozice, např. "1,3" (pozice 1 a 3 se nebudou uskutečňovat)		
13		<_NSP>	INT	rezervováno		

Č.	Obrazovka parametrů	Parametry interní	Datový typ	Význam		
14		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)

4.26.1.36 CYCLE802 - Libovolné pozice

Syntaxe

```
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>)
```

Parametr

Č.	Obrazovka parametrů	Parametry interní	Datový typ	Význam		
1		<_XA>	INT	Alternativa pro všechny pozice na ose X (9-místné desítkové číslo) Číslo místa: 876543210 (místo odpovídá poloze vrtané díry Xn)		
				Hodnota místa:	1 =	absolutní (1. naprogramovaná pozice vždy absolutně)
					2 =	inkrementální
2		<_YA>	INT	Alternativa pro všechny pozice na ose Y (9-místné desítkové číslo) Číslo místa: 876543210 (místo odpovídá poloze vrtané díry Yn)		
				Hodnota místa:	1 =	absolutní (1. naprogramovaná pozice vždy absolutně)
					2 =	inkrementální
3	X0	<_X0>	REAL	1. poloha X		
4	Y0	<_Y0>	REAL	1. poloha Y		
5	X1	<_X1>	REAL	2. poloha X		
6	Y1	<_Y1>	REAL	2. poloha Y		
7	X2	<_X2>	REAL	3. poloha X		
8	Y2	<_Y2>	REAL	3. poloha Y		
9	X3	<_X3>	REAL	4. poloha X		
10	Y3	<_Y3>	REAL	4. poloha Y		
11	X4	<_X4>	REAL	5. poloha X		
12	Y4	<_Y4>	REAL	5. poloha Y		

Č.	Obrazovka parametrů	Parametry interní	Datový typ	Význam
13	X5	<_X5>	REAL	6. poloha X
14	Y5	<_Y5>	REAL	6. poloha Y
15	X6	<_X6>	REAL	7. poloha X
16	Y6	<_Y6>	REAL	7. poloha Y
17	X7	<_X7>	REAL	8. poloha X
18	Y7	<_Y7>	REAL	8. poloha Y
19	X8	<_X8>	REAL	9. poloha X
20	Y8	<_Y8>	REAL	9. poloha Y

Č.	Obrazovka parametrů	Parametry interní	Datový typ	Význam		
21		<_VARI>	INT	Opracování		
				STOVKY:	(jen pro vyvolávání z aplikace JobShop) (zatím jsou vyhodnocovány jen 0 a 2)	
					0 =	žádné zablokování vřetena
					1 =	vřeteno zablokovat jen při kolmém zajíždění nástroje s příkazem G00 nebo G01
					2 =	vřeteno zablokovat během celého opracování
				TISÍCE:	rezervováno	
				DESÍTKY TISÍC:	Polohovací vzor s/bez kruhové osy – kombinace os (s nastavením <_VARI> - STOVKY TISÍC)	
					0 =	XY (jen XY, bez kruhové osy, kompatibilita)
					1 =	X, Y nebo Z a kruhová osa: XA, YB, ZC (1 kruhová osa s geometrickou osou, okolo které se kruhová osa otáčí)
					2 =	XY a kruhová osa: XYA, XYB, XYC (1 kruhová osa s 1. a 2. geometrickou osou, bez transformace TRACYL)
				STOVKY TISÍC:	Kruhová osa	
					0 =	bez kruhové osy (jen XY, kompatibilita)
					1 =	osa A (kruhová osa oko osy X)
					2 =	osa B (kruhová osa oko osy Y)
					3 =	osa C (kruhová osa oko osy Z)
				DESÍTKY MILIONŮ + MILIONY:	Polohovací vzor s kruhovou osou - offset (pokud se více kruhových otáčí okolo stejné osy; pokud je index příliš veliký, potom 1. osa)	
					00 =	1. osa A, B nebo C nebo při kompatibilitě
					01 =	2. osa A, B nebo C
					...	
					19 =	20. osa A, B nebo C
22		<_UMODE>	INT	Volba vřetena, které má být zablokováno: (jen při vyvolávání z aplikace JobShop) (vyvolání uživatelského cyklu CUST_TECHCYC)		
					3 =	Zablokování/uvolnění hlavního vřetena
					23 =	Zablokování/uvolnění protivřetena

Č.	Obrazovka parametrů	Parametry interní	Datový typ	Význam		
23		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
24		<S_ABA>	INT	Alternativa pro všechny pozice AB (9-místné desetinné číslo) Číslo místa: 876543210 (místo odpovídá poloze ABn)		
				Hodnota místa:	1 =	absolutní (1. naprogramovaná pozice vždy absolutně)
					2 =	inkrementální
25	A0	<S_AB0>	REAL	1. Poloha kruhové osy v případě polohovacího vzoru s kruhovou osou (ve spojení s příkazem <_VARI>)		
26	A1	<S_AB1>	REAL	2. Poloha kruhové osy v případě polohovacího vzoru s kruhovou osou		
27	A2	<S_AB2>	REAL	3. Poloha kruhové osy v případě polohovacího vzoru s kruhovou osou		
28	A3	<S_AB3>	REAL	4. Poloha kruhové osy v případě polohovacího vzoru s kruhovou osou		
29	A4	<S_AB4>	REAL	5. Poloha kruhové osy v případě polohovacího vzoru s kruhovou osou		
30	A5	<S_AB5>	REAL	6. Poloha kruhové osy v případě polohovacího vzoru s kruhovou osou		
31	A6	<S_AB6>	REAL	7. Poloha kruhové osy v případě polohovacího vzoru s kruhovou osou		
32	A7	<S_AB7>	REAL	8. Poloha kruhové osy v případě polohovacího vzoru s kruhovou osou		
33	A8	<S_AB8>	REAL	9. Poloha kruhové osy v případě polohovacího vzoru s kruhovou osou		

Poznámka

Nepotřebné pozice dané parametry X1/Y1/A1 až X8/Y8/A8 mohou být vypuštěny. Alternativní hodnoty pro parametry <_XA>, <_YA> a <S_ABA> je však zapotřebí úplně zadat pro všech 9 pozic.

V případě polohovacího vzoru XA, ZB nebo ZC (jedna geometrická osa a kruhová osa) je potřeba, aby osa pracovní roviny, se kterou se prostřednictvím polohovacího vzoru nepohybuje (v případě G17 a XA je to osa Y) byla nastavena do požadované polohy ještě před voláním cyklu.

4.26.1.37 CYCLE806 - Interpolační soustružení

Poznámka

Interpolační soustružení s cyklem CYCLE806 je volitelný doplněk, k jehož využívání je nutno zakoupit licenci!

Syntaxe

```
CYCLE806(<_MODE>, <S_XC>, <S_YC>, <_STA>, <_SV1>, <_SV2>, <_SDIR>,
<_AMODE>, <_DMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1		<_MODE>	INT	Režim		
				JEDNOTKY:	Aktivování/deaktivování	
					0 =	Deaktivování interpolačního soustružení
					1 =	Aktivování interpolačního soustružení
2	XC	<S_XC>	REAL	Střed na ose X		
3	YC	<S_YC>	REAL	Střed na ose Y		
4	Gama	<_STA>	REAL	Úhel gama (otočení okolo nástrojové osy); rezervováno		
5	S	<_SV1>	REAL	Konstantní otáčky vřetena, příp. konstantní řezná rychlost (viz JEDNOTKY parametru <_AMODE>)		
6	SV	<_SV2>	REAL	Maximální otáčky při konstantní řezné rychlosti (rezervováno)		
7		<_SDIR>	INT	rezervováno		
					0 =	Směr otáčení z parametrů nástroje
					3 =	Směr otáčení M3
					4 =	směr otáčení M4
8		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Pohyb vřetena (<_SV1>)	
					0 =	konstantní otáčky vřetena
					1 =	konstantní řezná rychlost
9		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					1 =	G17 (aktivní jen v cyklu)
10	TC	<S_TRA>	STRING	• Název transformace • prázdné: Použije se 1. kinematická transformace TRAINT		

4.26.1.38 CYCLE830 - Vrtání hlubokých děr 2

Syntaxe

```
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>)
```

Parametry

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	RP	<RTP>	REAL	Návratová rovina (abs)		
2	Z0	<RFP>	REAL	Vztažný bod (abs)		
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, bez znaménka)		
4	Z1	<_DP>	REAL	Konečná vrtaná hloubka, abs/ink (viz <_AMODE> - JEDNOTKY)		
5	D	<FDEP>	REAL	1. vrtaná hloubka kvůli ulamování třísky/odstraňování třísek absolutně nebo inkrementálně vzhledem ke vztažnému bodu při navrtávání/bez navrtávání nebo vzhledem ke hloubce předvrtané díry (viz <_AMODE> - DESÍTKY TISÍC)		
6	DF	<_DAM>	REAL	Absolutní hodnota/procentuální změna pro každý další přísuv - degrese/ procentuální změna (viz <_AMODE> - STOVKY TISÍC)		
7	DTB	<DTB>	REAL	Doba prodlevy na každé vrtané hloubce (viz <_AMODE> - DESÍTKY)		
8	DTS	<DTS>	REAL	Doba prodlevy při odstraňování třísek v počátečním bodě (viz <_AMODE> - STOVKY)		
9	FD1	<FRF>	REAL	Procentuální hodnota pro posuv při prvním přísuvu (viz <_AMODE> - DESÍTKY MILIONŮ)		
10		<VARI>	INT	Opracování		
				JEDNOTKY:	Ulamování třísky/odstraňování třísek	
					0 =	v jednom průchodu nástroje
					1 =	Ulamování třísky
					2 =	Odstraňování třísek
					3 =	Ulamování třísky a odstraňování třísek
				DESÍTKY:	Zpětný pohyb při odstraňování třísek	
					0 =	na hloubku předvrtané díry
					1 =	na bezpečnostní vzdálenost
				STOVKY:	měkké zaříznutí nástroje	
					0 =	ne
					1 =	ano
				TISÍCE:	Provrtání	
					0 =	ne
					1 =	ano
				DESÍTKY TISÍC:	Navrtávání/předvrtaná díra	
					0 =	bez navrtání
					1 =	s navrtáním
					2 =	s předvrtáním
				STOVKY TISÍC:	Zpětný pohyb	
					0 =	na hloubku předvrtané díry
					1 =	na návratovou rovinu
11	V1	<_MDEP>	REAL	minimální přísuv inkrementálně (jen když je degrese v procentech)		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
12	V2	<_VRT>	REAL	Vzdálenost zpětného pohybu po každém obrobení inkrementálně (jen při ulamování třísky)		
					0 =	standardní hodnota 1 mm
					> 0 =	proměnná vzdálenost zpětného pohybu
13	DT	<_DTD>	REAL	Doba prodlevy na konečné vrtané hloubce (viz <_AMODE> - TISÍCE)		
14	V3	<_DIS1>	REAL	Chráněná vzdálenost inkrementálně, jen při odstraňování třísek (viz <_AMODE> - JEDNOTKY MILIONŮ)		
15	FP	<S_FP>	REAL	Posuv pro najíždění v předvrtané díře jako hodnota nebo v % (ve spojení s místem STOVEK parametru <S_AMODE2>)		
16		<S_SDAC2>	INT	Směr otáčení vřetena při najíždění		
					3 =	M3
					4 =	M4
					5 =	M5 (předdefinované)
17	SP	<S_SV2>	REAL	Najíždění s	konstantními otáčkami vřetena (viz <S_AMODE2> - DESÍTKY MILIONŮ)	
	V4				konstantní řezná rychlost	
				Otáčky vřetena v % z otáček vrtáku		
18	F	<S_FB>	REAL	Posuv při vrtání (viz JEDNOTKY parametru <S_AMODE2>)		
19		<_SDAC>	REAL	Směr otáčení vřetena při vrtání		
					3 =	M3
					4 =	M4
20	S	<_SV1>	REAL	Vrtání s	konstantními otáčkami vřetena (viz <S_AMODE2> - JEDNOTKY MILIONŮ)	
	V5				konstantní řezná rychlost	
21	SPOS	<S_SPOS>	REAL	Poloha vřetena, jen při najíždění s M5		
22	ZA	<S_ZA>	REAL	Navrtávaná hloubka vztažena inkrementálně na vztažný bod nebo absolutně (viz <S_AMODE3> - JEDNOTKY)		
23	FA	<S_FA>	REAL	Posuv při navrtávání jako hodnota nebo v % (ve spojení s místem DESÍTEK TISÍCŮ parametru <S_AMODE2>)		
24	ZP	<S_ZP>	REAL	Předvrtání inkrementálně vzhledem ke vztažnému bodu, absolutně nebo pomocí faktoru vzhledem k průměru vrtané díry (viz <S_AMODE3> - DESÍTKY)		
25	FS	<S_FS>	REAL	Posuv při při zařezávání nástroje jako hodnota nebo v % (ve spojení s místem TISÍCŮ parametru <S_AMODE2>)		
26	ZS1	<S_ZS1>	REAL	Hloubka každého zaříznutí nástroje s konstantním posuvem (ink)		
27	ZS2	<S_ZS2>	REAL	Hloubka každého zaříznutí nástroje pro zvýšení hodnoty posuvu (ink)		
28	N	<S_N>	INT	Počet zdvihů kvůli ulomení třísky před každým odstraněním třísek		
29	ZD	<S_ZD>	REAL	Zbytková vrtaná hloubka vztažena inkrementálně na konečnou vrtanou hloubku nebo absolutně (ve spojení s místem STOVEK parametru <S_AMODE3>)		
30	FD	<S_FD>	REAL	Posuv při vrtání zbytkové hloubky jako hodnota nebo v % (ve spojení s místem DESÍTKY TISÍCŮ parametru <S_AMODE2>)		
31	FR	<S_FR>	REAL	Posuv při zpětném pohybu (ve spojení s parametrem <S_AMODE2> - STOVKY TISÍC)		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
32		<S_SDAC3>	INT	Směr otáčení vřetena při zpětném pohybu		
					3 =	M3
					4 =	M4
					5 =	M5
33	SR	<S_SV3>	REAL	Zpětný pohyb s	konstantními otáčkami vřetena (viz <S_AMODE2> - STOVKY MILIONŮ)	
	V6				konstantní řezná rychlost	
				Otáčky vřetena v % z otáček vrtáku		
34	Chladicí kapalina zapnuta	<S_CON>	STRING[10]	Chladicí kapalina zapnuta, M-příkaz nebo volání podprogramu		
35	Chladicí kapalina vypnuta	<S_COFF>	STRING[10]	Chladicí kapalina vypnuta, M-příkaz nebo volání podprogramu		
36		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	Vrtaná hloubka vztažena na špičku/stopku	
					0 =	Špička
					1 =	Stopka
37		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	rezervováno	
				STOVKY:	rezervováno	
				TISÍCE:	rezervováno	
				DESÍTKY TISÍC:	Odstupňování technologie na obrazovkách cyklů (Strana 1169)	
					0 =	Zadání: kompletní
					1 =	Zadání: jednoduché

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
38		<_AMODE>	INT	Alternativní režim 1		
				JEDNOTKY:	Vrtaná hloubka = konečná vrtaná hloubka Z1 (abs/ink)	
					0 =	inkrementální
					1 =	absolutní
				DESÍTKY:	Doba prodlevy na každé vrtané hloubce DTB v sekundách/otáčkách	
					0 =	v sekundách
					1 =	v otáčkách
				STOVKY:	Doba prodlevy kvůli odstranění třísek DTS v sekundách/otáčkách	
					0 =	v sekundách
					1 =	v otáčkách
				TISÍCE:	Doba prodlevy na konečné vrtané hloubce DT v sekundách/otáčkách	
					0 =	v sekundách
					1 =	v otáčkách
				DESÍTKY TISÍC:	1. vrtaná hloubka D abs/ink	
					0 =	inkrementální
					1 =	absolutní
				STOVKY TISÍC:	Absolutní hodnota/procentuální hodnota DF pro každý další přísuv (degrese)	
					0 =	absolutní hodnota
					1 =	Procentuální změna (0,001 až 100%)
				JEDNOTKY MILIONŮ:	Chráněná vzdálenost V3 automaticky/manuálně	
					0 =	automaticky (vypočítává se v cyklu)
					1 =	manuálně (naprogramovaná hodnota)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
39		<S_AMODE2>	INT	Alternativní režim 2		
				JEDNOTKY:	JEDNOTKY: Posuv při vrtání F	
					0 =	F/min
					1 =	F/ot
				DESÍTKY:	Vyhodnocování posuvu při navrtávání FA	
					0 =	v % z posuvu při vrtání
					1 =	F/min
					2 =	F/ot
				STOVKY:	Vyhodnocování posuvu pro najíždění na předvrtanou díru FP	
					0 =	v % z posuvu při vrtání
					1 =	F/min
					2 =	F/ot
				TISÍCE:	Vyhodnocování posuvu pro zařízнutí nástroje FS	
					0 =	v % z posuvu při vrtání
					1 =	F/min
					2 =	F/ot
				DESÍTKY TISÍC:	Vyhodnocování posuvu pro provrtávání FD	
					0 =	v % z posuvu při vrtání
					1 =	F/min
					2 =	F/ot
				STOVKY TISÍC:	Posuv při zpětném pohybu FR	
					0 =	F/min
					1 =	Rychlý posuv
				JEDNOTKY MILIONŮ:	Vrtání - otáčky vřetena/řezná rychlost (S/V5)	
					0 =	konstantní otáčky vřetena
					1 =	konstantní řezná rychlost
				DESÍTKY MILIONŮ:	Najíždění s otáčkami vřetena/řeznou rychlostí (SP/V4)	
					0 =	konstantní otáčky vřetena
					1 =	konstantní řezná rychlost
					2 =	Otáčky vřetena v % z otáček vrtáku
				STOVKY MILIONŮ:	Zpětný pohyb - otáčky vřetena/řezná rychlost (SR/V6)	
					0 =	konstantní otáčky vřetena
					1 =	konstantní řezná rychlost
					2 =	Otáčky vřetena v % z otáček vrtáku

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
40		<S_AMODE3>	INT	Alternativní režim 3		
				JEDNOTKY:	Hloubka pro navrtávání ZA abs/ink	
					0 =	inkrementální
					1 =	absolutní
				DESÍTKY:	Hloubka předvrtané díry ZP	
					0 =	inkrementální
					1 =	absolutní
					2 =	Faktor pro vynásobení průměru díry
				STOVKY:	Zbytková vrtaná hloubka ZD ink/abs	
					0 =	inkrementální
					1 =	absolutní
41	ZPV	<S_ZPV>	REAL	Chráněná vzdálenost inkrementálně na hloubku předvrtané díry		

4.26.1.39 CYCLE832 - Vysokorychlostní obrábění

Syntaxe

```
CYCLE832(<S_TOL>, <S_TOLM>, <S_OTOL>)
```

Poznámka

Cyklus CYCLE832 nezbavuje výrobce stroje povinnosti uskutečnit nezbytné optimalizační úkoly při uvádění stroje do provozu. To se týká optimalizace os podílejících se na obrábění a nastavení parametrů NCU (dopředná regulace, omezení ryvu atd.).

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	Tolerance	<S_TOL>	REAL	Tolerance kontury Tolerance kontury odpovídá toleranci geometrických os.		
2		<S_TOLM>	INT	Druh obrábění (technologie)		
				JEDNOTKY:		
					0 =	Deaktivování
					1 =	Obrábění načisto (FINISH)
					2 =	Předběžné obrábění načisto (SEMI-FINISH)
					3 =	Obrábění nahrubo (ROUGH)
					4 =	Jemné obrábění načisto (Precision)
				DESÍTKY:		
					0 =	kompatibilita[1], příp. žádná tolerance orientace
					1 =	Tolerance orientace v parametru <S_OTOL>
				STOVKY ... STOVKY TISÍC	obsazeno kvůli kompatibilitě	
				JEDNOTKY MILIONŮ:		
					0 =	Kompatibilita. Automaticky se použije nejlepší funkce pro výrobu forem, jaká je k dispozici: • Volitelný doplněk Top Surface neaktivní: ⇒ Advanced Surface • Volitelný doplněk Top Surface aktivní: ⇒ Top Surface s vyhlazením
					1 =	Top Surface bez vyhlazení
					2 =	Top Surface s vyhlazením
3	Tolerance ORI	<S_OTOL>	REAL	Tolerance orientace, příp. identifikace verze, CYCLE832 Parametr tolerance pro orientaci nástroje. Je zapotřebí při zpracovávání programů pro vysokorychlostní obrábění na strojích s možností dynamické transformace orientace (např. při obrábění v 5 osách). Parametr <S_OTOL> **musí** být naprogramován. To platí i pro aplikace na strojích se 3 osami u programů bez možnosti nastavovat orientaci nástroje (<S_OTOL> = 1).		

[1] Tolerance orientace odvozená z nastavovaných parametrů cyklů SD55451 ... SD55454 (tolerance orientace pro dynamický režim ...), příp. SD55445 ... SD55449 (tolerance kontury pro dynamický režim ...) vynásobená faktorem z parametrů SD55441 ... SD55444.
Další informace: Příručka pro uvedení do provozu, SINUMERIK Operate

Zadávání prostého textu

Aby se zlepšila čitelnost volání cyklů, může být parametr <S_TOLM> (způsob obrábění) zadáván také prostým textem. Prosté texty závisejí na nastaveném jazyce. Přípustné jsou následující možnosti zadání:

_OFF	pro	0	Deaktivování
_FINISH	pro	1	Obrábění načisto
_SEMIFIN	pro	2	Předběžné obrábění načisto
_ROUGH	pro	3	Obrábění nahrubo
_PRECISION	pro	4	Jemné obrábění načisto
_ORI_FINISH	pro	11	Obrábění načisto se zadáním tolerance orientace
_ORI_SEMIFIN	pro	12	Předběžné obrábění načisto se zadáním tolerance orientace
_ORI_ROUGH	pro	13	Obrábění nahrubo se zadáním tolerance orientace
_ORI_PRECISION	pro	14	Jemné obrábění načisto se zadáním tolerance orientace
_TOP_SURFACE_SMOOTH_OFF	pro	1000000	Top Surface bez vyhlazení
_TOP_SURFACE_SMOOTH_ON	pro	2000000	Top Surface s vyhlazením

V případě zadávání prostého textu u funkce Top Surface se prosté texty kombinují způsobem uvedeným v následujícím příkladu:

```
CYCLE832(0.1, _TOP_SURFACE_SMOOTH_OFF+_ORI_FINISH, 1)
```

Poznámka

Prosté texty jsou založeny na názvech funkcí z G-skupiny 59 (režim dynamiky pro dráhovou interpolaci). Pomocí těchto prostých textů jsou stroje se 3 osami a stroje s transformací orientace s více osami (TRAORI) při aplikacích jasně odděleny.

Deaktivování cyklu CYCLE832

Při deaktivování cyklu CYCLE832 musí být parametr <S_TOL> předáván s nulovou hodnotou.

Příklad: `CYCLE832(0,0,1)`

Syntaxe `CYCLE832()` je pro deaktivování cyklu CYCLE832 přípustná také.

Příklady

Příklad 1: CYCLE832 na stroji se 3 osami bez transformace orientace

a) Volání cyklu se zadáváním prostých textů

Programový kód	Komentář
G710	; systém jednotek je metrický

Programový kód	**Komentář**
CYCLE832(0.004,_FINISH,1)	; volání cyklu CYCLE832 s parametry: Tolerance kontury = 0,004 mm, druh opracování: Obrábění načisto
...	; zpracování programu pro vysokorychlostní obrábění

b) Volání cyklu bez zadávání prostých textů

Programový kód	**Komentář**
G710	; viz výše
CYCLE832(0.004,1,1)	; viz výše
...	; viz výše

Příklad 2: CYCLE832 na stroji s 5 osami a s transformací orientace

a) Volání a deaktivování cyklu se zadáváním prostých textů

Programový kód	**Komentář**
G710	; systém jednotek je metrický
TRAORI	; aktivování transformace orientace
CYCLE832(0.3,_ORI_ROUGH,0.8)	; volání cyklu CYCLE832 s parametry: Tolerance kontury = 0,3 mm, druh opracování: Obrábění nahrubo se zadáním tolerance orientace, tolerance orientace = 0,8 stupně
...	; zpracování programu pro vysokorychlostní obrábění
CYCLE832(0,_OFF,1)	; tolerance kontury = 0, Druh opracování: Deaktivování cyklu CYCLE832, Tolerance orientace = 0 stupňů

b) Volání a deaktivování cyklu bez zadávání prostých textů

Programový kód	**Komentář**
G710	; viz výše
TRAORI	; viz výše
CYCLE832(0.3,13,0.8)	; viz výše
...	; viz výše
CYCLE832(0,0,1)	; viz výše

4.26.1.40 CYCLE840 - Vrtání závitů s vyrovnávací hlavičkou

Syntaxe

```
CYCLE840(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDR>, <SDAC>,
<ENC>, <MPIT>, <PIT>, <_AXN>, <_PITA>, <_TECHNO>, <_PITM>, <_PTAB>,
<_PTABA>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam			
1	RP	<RTP>	REAL	Návratová rovina (abs)			
2	Z0	<RFP>	REAL	Vztažný bod (abs)			
3	SC	<SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)			
4	Z1	<DP>	REAL	Vrtaná hloubka (abs), viz <_AMODE>			
5	Z1	<DPR>	REAL	Vrtaná hloubka (ink), viz <_AMODE>			
6	DT	<DTB>	REAL	Doba prodlevy na vrtané hloubce/na bezpečnostní vzdálenosti po zpětném pohybu v sekundách, viz <ENC>			
7		<SDR>	INT	Směr otáčení pro zpětný pohyb			
8	SDE	<SDAC>	INT	Směr otáčení po skončení cyklu			
9		<ENC>	INT	Vrtání závitu s vřetenovým snímačem (G33)/vrtání závitu bez vřetenového snímače (G63)			
					0 =	s vřetenovým snímačem	- stoupání z <MPIT>/<PIT> - bez DT
					20 =	s vřetenovým snímačem	- stoupání z <MPIT>/<PIT> - s DT po zpětném pohybu na bezpečnostní vzdálenost
					11 =	bez vřetenového snímače	- stoupání z <MPIT>/<PIT> - s DT na vrtané hloubce
					1 =	bez vřetenového snímače	- Stoupání z naprogramovaného posuvu - s DT na vrtané hloubce (posuv = otáčky stoupání)
10		<MPIT>	REAL	Velikost závitu, pouze pro "ISO metrický" (stoupání se vypočítává interně v průběhu zpracovávání programu) Rozsah hodnot: 3 až 48 (pro M3 až M48), alternativa k <PIT>			
11		<PIT>	REAL	Stoupání závitu jako hodnota, rozměrové jednotky viz <_PITA> Rozsah hodnot: > 0, alternativa k MPIT			

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam			
12		<_AXN>	INT	Osa vrtání	0 =	3. Geometrická osa	
					1 =	1. Geometrická osa	
					2 =	2. Geometrická osa	
					≥ 3 =	3. Geometrická osa	
13		<_PITA>	INT	Rozměrové jednotky pro stoupání závitu (vyhodnocování <PIT> a <MPIT>)			
					0 =	stoupání v mm	- vyhodnocování <MPIT>/<PIT>
					1 =	stoupání v mm	- vyhodnocování <PIT>
					2 =	stoupání v TPI	- vyhodnocování <PIT> (počet chodů závitu na palec)
					3 =	stoupání v palcích	- vyhodnocování <PIT>
					4 =	MODUL	- vyhodnocování <PIT>
14		<_TECHNO>	INT	Technologie[1)]			
				JEDNOTKY:	Režim přesného najetí		
					0 =	je aktivní chování přesného najetí, jaké bylo před voláním cyklu	
					1 =	Přesné najetí G601	
					2 =	Přesné najetí G602	
					3 =	Přesné najetí G603	
				DESÍTKY:	Dopředná regulace		
					0 =	s/bez dopředné regulace, jak bylo aktivní před voláním cyklu	
					1 =	s dopřednou regulací FFWON	
					2 =	bez dopředné regulace FFWOF	
15		<_PITM>	STRING[15]	Řetězec jako ukazatel pro zadávání stoupání závitu[2)]			
16		<_PTAB>	STRING[5]	Řetězec pro tabulku závitů ("", "ISO", "BSW", "BSP", "UNC")[2)]			
17		<_PTABA>	STRING[20]	Řetězec pro volbu v tabulce závitů (např. "M 10", "M 12", ...)[2)]			
18		<_GMODE>	INT	rezervováno			

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
19		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	rezervováno	
				STOVKY:	rezervováno	
				TISÍCE:	Režim kompatibility (jen pro vstupní obrazovku pro zpětný překlad), pokud je MD 52216 Bit0 = 1	
					0 =	technologické parametry se vypisují (kompatibilita): parametr TECHNO je v platnosti
					1 =	technologické parametry se nevypisují: platí technologie, jaká byla naprogramována před voláním cyklu
				DESÍTKY TISÍC:	Odstupňování technologie na obrazovkách cyklů (Strana 1169)	
					0 =	Zadání: kompletní
					1 =	Zadání: jednoduché
20		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Vrtaná hloubka Z1 (abs/ink)	
					0 =	Kompatibilita, podle naprogramování <DP>/<DPR>
					1 =	inkrementální
					2 =	absolutní

[1)]V závislosti na nastavovaném parametru SD52216 MCS_FUNCTION_MASK_DRILL může být zobrazování polí technologie blokováno.

[2)] Parametry 15, 16 a 17 se používají pouze při volbě závitu v tabulce závitů ve vstupní obrazovce. Přístup k tabulkám závitů prostřednictvím definice cyklů v době, kdy jsou cykly zpracovávány, není možný!

4.26.1.41 CYCLE899 - Otevřená drážka

Syntaxe

```
CYCLE899(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_LENG>, <_WID>, <_PA>,
<_PO>, <_STA>, <_MID>, <_MIDA>, <_FAL>, <_FALD>, <_FFP1>, <_CDIR>,
<_VARI>, <_GMODE>, <_DMODE>, <_AMODE>, <_UMODE>, <_FS>, <_ZFS>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	RP	<_RTP>	REAL	Návratová rovina (abs)		
2	Z0	<_RFP>	REAL	Vztažný bod osy nástroje (abs)		
3	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (přičítá se ke vztažnému bodu, zadává se bez znaménka)		
4	Z1	<_DP>	REAL	Hloubka drážky (abs/ink), viz <_AMODE>		
5	L	<_LENG>	REAL	Délka drážky (ink)		
6	W	<_WID>	REAL	Šířka drážky (ink)		
7	X0	<_PA>	REAL	Vztažný bod/počáteční pozice 1. osy (abs)		
8	Y0	<_PO>	REAL	Vztažný bod/počáteční pozice 2. osy (abs)		
9	α0	<_STA>	REAL	Úhel otočení vzhledem k 1. ose		
10	DZ	<_MID>	REAL	Maximální přísuvná hloubka (ink), pouze u trochoidálního frézování		
11	DXY	<_MIDA>	REAL	Maximální přísuv v rovině (viz <_AMODE>)		
12	UXY	<_FAL>	REAL	Přídavek rozměru pro opracování načisto v rovině		
13	UZ	<_FALD>	REAL	Přídavek rozměru pro opracování načisto na dně		
14	F	<_FFP1>	REAL	Posuv		
15		<_CDIR>	INT	Směr frézování		
				JEDNOTKY:		
					0 =	Sousledné frézování
					1 =	Nesousledné frézování
					4 =	Střídavě
16		<_VARI>	INT	Opracování		
				JEDNOTKY:		
					1 =	Obrábění nahrubo
					2 =	Obrábění načisto
					3 =	Obrábění dna načisto
					4 =	Obrábění okraje načisto
					5 =	Předběžné obrábění načisto
					6 =	Srážení hran
				DESÍTKY:	rezervováno	
				STOVKY:	rezervováno	
				TISÍCE:		
					1 =	Trochoidální frézování
					2 =	Zapichovací frézování

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
17		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	rezervováno	
				STOVKY:	Volba opracování/pouze výpočet počátečního bodu	
					1 =	normální opracování
				TISÍCE:	Zadávání rozměrů pomocí středu / hrany	
					0 =	Zadávání rozměrů pomocí středu
					1 =	Kótování pomocí hrany "vlevo" ("-" směr 1. osy)
					2 =	Kótování pomocí hrany "vpravo" ("+" směr 1. osy)
18		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	---	rezervováno
				STOVKY:	---	rezervováno
				TISÍCE:	---	rezervováno
				DESÍTKY TISÍC:	Odstupňování technologie na obrazovkách cyklů (Strana 1169)	
					0 =	Zadání: kompletní
					1 =	Zadání: jednoduché
19		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Hloubka drážky Z1	
					0 =	absolutní
					1 =	inkrementální
				DESÍTKY:	Jednotky pro přísuv v rovině (<_MIDA>) DXY	
					0 =	mm
					1 =	procentuální hodnota z průměru nástroje
				STOVKY:	Hloubka zajíždění při srážení hran ZFS	
					0 =	absolutní
					1 =	inkrementální
20		<_UMODE>	INT	rezervováno		
21	FS	<_FS>	REAL	Šířka fasety (ink)		
22	ZFS	<_ZFS>	REAL	Hloubka zajíždění (špička nástroje) při srážení hran (abs/ink), viz <_AMODE>		

4.26.1.42 CYCLE930 - Zápich

Syntaxe

```
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>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
1	X0	<_SPD>	REAL	Souřadnice vztažného bodu v příčné ose (vždycky průměr)		
2	Z0	<_SPL>	REAL	Souřadnice vztažného bodu v podélné ose		
3	B1	<_WIDG>	REAL	Šířka zápichu dole		
4	B2	<_WIDG2>	REAL	Šířka zápichu nahoře (jen pro pracovní plochu)		
5	T1	<_DIAG>	REAL	Hloubka zápichu ve vztažném bodě, v případě absolutních rozměrů a podélného obrábění = průměr, jinak inkrementální rozměr		
6	T2	<_DIAG2>	REAL	Hloubka zápichu naproti vztažného bodu (jen pro pracovní plochu), v případě absolutních rozměrů a podélného obrábění = průměr, jinak inkrementální rozměr		
7	α0	<_STA>	REAL	Úhel šikmé plochy (-180 ≤ <_STA> ≤ 180)		
8	α1	<_ANG1>	REAL	Úhel stěny zápichu 1 (0 ≤ <_ANG1> < 90) na straně zápichu určeného vztažným bodem		
9	α2	<_ANG2>	REAL	Úhel stěny zápichu 2 (0 ≤ <_ANG2> < 90) naproti vztažného bodu		
10	R1/FS1	<_RCO1>	REAL	Rádius zaoblení nebo šířka fasety 1, vně u vztažného bodu		
11	R2/FS2	<_RCI1>	REAL	Rádius zaoblení nebo šířka fasety 2, uvnitř u vztažného bodu		
12	R3/FS3	<_RCI2>	REAL	Rádius zaoblení nebo šířka fasety 3, uvnitř naproti vztažného bodu		
13	R4/FS4	<_RCO2>	REAL	Rádius zaoblení nebo šířka fasety 4, vně naproti vztažného bodu		
14	U	<_FAL>	REAL	Přídavek rozměru pro opracování načisto ve směru X a Z, viz <_VARI> (DESÍTKY TISÍC) (zadává se bez znaménka)		
15	D	<_IDEP1>	REAL	Maximální přísuv do hloubky při zajíždění nástroje (zadává se bez znaménka)		
					0 =	1. průchod nástroje přímo na celou hloubku
					> 0 =	1. průchod nástroje na hloubku <_IDEP1>, 2. průchod nástroje 2 · <_IDEP1> atd.
16	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost (zadává se bez znaménka)		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
17		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	Technologie opracování	
					1 =	Obrábění nahrubo
					2 =	Obrábění načisto
					3 =	Obrábění nahrubo a obrábění načisto
				STOVKY:	Poloha v podélném/příčném směru, vnější/vnitřní +Z/-Z, příp +X/-X	
					1 =	podélná/vnější +Z
					2 =	příčná/vnitřní -X
					3 =	podélná/vnitřní +Z
					4 =	příčná/vnitřní +X
					5 =	podélná/vnější -Z
					6 =	příčná/vnější -X
					7 =	podélná/vnitřní -Z
					8 =	příčná/vnější +X
				TISÍCE:	Poloha vztažného bodu	
					0 =	Vztažný bod nahoře
					1 =	Vztažný bod dole
				DESÍTKY TISÍC:	Definice platí jako přídavek rozměru pro opracování načisto	
					0 =	Přídavek rozměru pro opracování načisto U rovnoběžně s konturou
					1 =	Přídavek rozměru pro opracování načisto UX a UZ odděleně
18		<_DN>	INT	D-číslo pro 2. břit nástroje		
					> 0 =	D-číslo pro korekci nástroje pro 2. břit zapichovacího nože
					0 =	není naprogramován žádný 2. břit
19	N	<_NUM>	INT	Počet zápichů (0 = 1 zápich)		
20	DP	<_DBH>	REAL	Vzdálenost zápichů (je zapotřebí jen tehdy, pokud je <_NUM> > 1)		
21	F	<_FF1>	REAL	Posuv		
22		<_NR>	INT	Identifikace tvaru zápichu (odpovídá programovým tlačítkům ve svislém pruhu obrazovky)		
					0 =	90° stěny zápichů bez faset/ zaoblení
					1 =	šikmé stěny s fasetami/zaobleními (bez α0)
					2 =	jako 1, ale na kuželu (s α0)
23	UX	<_FALX>	REAL	Přídavek rozměru pro opracování načisto ve směru osy X, viz <_VARI> (DESÍTKY TISÍC) (zadává se bez znaménka)		
24	UZ	<_FALZ>	REAL	Přídavek rozměru pro opracování načisto ve směru osy Z, viz <_VARI> (DESÍTKY TISÍC) (zadává se bez znaménka)		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
25		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
26		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Zadávání rozměrů hloubky (jen pro pracovní plochu)	
					0 =	na vztažný bod
					1 =	naproti vztažného bodu
				DESÍTKY:	Hloubka	
					0 =	absolutní
					1 =	inkrementální
				STOVKY:	Zadávání rozměrů šířky (jen pro pracovní plochu)	
					0 =	na vnějším průměru (nahoře)
					1 =	na vnitřním průměru (dole)
				TISÍCE:	Rádius / faseta 1 (<_RCO1>)	
					0 =	Rádius
					1 =	Faseta
				DESÍTKY TISÍC:	Rádius / faseta 2 (<_RCI1>)	
					0 =	Rádius
					1 =	Faseta
				STOVKY TISÍC:	Rádius / faseta 3 (<_RCI2>)	
					0 =	Rádius
					1 =	Faseta
				JEDNOTKY MILIONŮ:	Rádius / faseta 4 (<_RCO2>)	
					0 =	Rádius
					1 =	Faseta

4.26.1.43 CYCLE940 - Odlehčovací zápich tvaru E a F / závitový zápich

Pomocí cyklu CYCLE940 mohou být naprogramovány různé odlehčovací zápichy. Pokud jde o dosazování parametrů, jsou mezi nimi značné rozdíly.

Přidané sloupce v tabulce ukazují, které parametry jsou u jednotlivých tvarů zápichu zapotřebí. Odpovídají programovým tlačítkům ve svislém pruhu na obrazovce cyklu.

- E: Odlehčovací zápich:tvar E
- F: Odlehčovací zápich:tvar F

- A-D: Závitový zápich podle DIN (tvary A - D)
- T: Závitový zápich (volná definice tvaru)

Syntaxe

```
CYCLE940(<_SPD>, <_SPL>, <_FORM>, <_LAGE>, <_SDIS>, <_FFP>, <_VARI>,
<_EPD>, <_EPL>, <_R1>, <_R2>, <_STA>, <_VRT>, <_MID>, <_FAL>,
<_FALX>, <_FALZ>, <_PITI>, <_PTAB>, <_PTABA>, <_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Progr. u tvaru				Význam		
				E	F	A-D	T			
1	X0	<_SPD>	REAL	x	x	x	x	Souřadnice vztažného bodu v příčné ose (vždycky průměr)		
2	Z0	<_SPL>	REAL	x	x	x	x	Vztažný bod, podélná osa (abs)		
3	FORM	<_FORM>	CHAR	x	x	x	x	Tvar zápichu (velká písmena, např. "T") Volba, ze které tabulky se mají brát hodnoty parametrů zápichu		
									A =	vnější, podle DIN 76, A = normální
									B =	vnější, podle DIN 76, B = krátký
									C =	vnitřní, podle DIN 76, C = normální
									D =	vnitřní, podle DIN 76, D = krátký
									E =	Podle DIN 509
									F =	Podle DIN 509
									T =	volný tvar
4	LAGE	<_LAGE>	INT	x	x	x	x	Poloha zápichu (rovnoběžně s osou Z)	0 =	Vnější +Z: ____|
									1 =	Vnější -Z: |____/
									2 =	Vnitřní +Z: /-----|
									3 =	Vnitřní-Z: |-----\
5	SC	<_SDIS>		x	x	x	x	Bezpečnostní vzdálenost (ink)		
6	F	<_FFP>		x	x	x	x	Pracovní posuv (mm/ot)		
7		<_VARI>	INT	-	-	x	x	Způsob opracování		
								JEDNOTKY:	Opracování	
									1 =	Obrábění nahrubo
									2 =	Obrábění načisto
									3 =	Obrábění nahrubo + načisto
								DESÍTKY:	Strategie obrábění	
									0 =	rovnoběžně s konturou
									1 =	podélný
				Zápichy tvarů E a F jsou vždy opracovávány v jednom tahu jako při obrábění načisto						

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Progr. u tvaru				Význam			
8	X1	<_EPD>		x	x	-	-	Přídavek rozměru pro opracování načisto ve směru X (abs/ink), viz <_AMODE>			
				-	-	-	x	Hloubka zápichu (abs/ink), viz <_AMODE>			
9	Z1	<_EPL>		-	x	-	-	Přídavek rozměru pro opracování načisto ve směru Z			
				-	-	-	x	Šířka zápichu (abs/ink), viz <_AMODE>			
10	R1	<_R1>		-	-	-	x	Rádius zaoblení na stěnách			
11	R2	<_R2>		-	-	-	x	Rádius zaoblení na rozích			
12	α	<_STA>		-	-	x	x	Úhel zajíždění do obrobku			
13	VX	<_VRT>		x	x	-	-	Příčný pohyb ve směru X (abs/ink), viz <_AMODE>			
				-	-	x	x	Příčný pohyb ve směru X při obrábění načisto (abs/ink), viz <_AMODE>			
14	D	<_MID>		-	-	x	x	Přísuv do hloubky			
15	U	<_FAL>		-	-	x	x	Přídavek rozměru pro obrábění načisto rovnoběžně s konturou, viz<_AMODE>			
16	UX	<_FALX>		-	-	x	x	Přídavek rozměru pro obrábění načisto ve směru X			
17	UZ	<_FALZ>		-	-	x	x	Přídavek rozměru pro obrábění načisto ve směru Z			
18	P	<_PITI>	INT	-	-	x	-	Volba stoupání, tvar A - D, odpovídá M1 ... M68			
								0 = 0.20 1 = 0.25 2 = 0.30 3 = 0.35 4 = 0.40 5 = 0.45	6 = 0.50 7 = 0.60 8 = 0.70 9 = 0.75 10 = 0.80 11 = 1.00	12 = 1.25 13 = 1.50 14 = 1.75 15 = 2.00 16 = 2.50 17 = 3.00	18 = 3.50 19 = 4.00 20 = 4.50 21 = 5.00 22 = 5.50 23 = 6.00
				x	x	-	-	Volba rádiusu/hloubky, tvar E, F			
								0 = 0.6 · 0.3 1 = 1.0 · 0.4 2 = 1.0 · 0.2 3 = 1.6 ··0.3	4 = 2.5 · 0.4 5 = 4.0 · 0.5 6 = 0.4 · 0.2 7 = 0.6 · 0.2	8 = 0.1 · 0.1 9 = 0.2 ··0.1	
19		<_PTAB>	STRING [5]					Řetězec pro tabulku závitů ("", "ISO", "BSW", "BSP", "UNC") (jen pro pracovní plochu)			
20		<_PTABA>	STRING [20]					Řetězec pro volbu v tabulce závitů (např. "M 10", "M 12", ...) (pouze pro uživatelské rozhraní)			
21		<_DMODE>	INT					Režim zobrazování			
				x	x	x	x	JEDNOTKY:	Rovina obrábění G17/G18/G19		
									0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu	
									1 =	G17 (aktivní jen v cyklu)	
									2 =	G18 (aktivní jen v cyklu)	
									3 =	G19 (aktivní jen v cyklu)	

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Progr. u tvaru				Význam		
22		<_AMODE>	INT					Alternativní režim		
				x	x	-	x	JEDNOTKY:	Parametr <_EPD>, přídavek rozměru ve směru X nebo hloubka odlehčovacího zápichu	
									0 =	absolutně (vždy průměr)
									1 =	inkrementální
				x	x	-	x	DESÍTKY:	Parametr <_EPL> , přídavek rozměru ve směru Z nebo šířka odlehčovacího zápichu	
									0 =	absolutní
									1 =	inkrementální
				x	x	x	x	STOVKY:	Parametr <_VRT>, příčný pohyb ve směru X	
									0 =	absolutně (vždy průměr)
									1 =	inkrementální
				-	-	x	x	TISÍCE:	Přídavek rozměru pro opracování načisto	
									0 =	Přídavek rozměru pro obrábění načisto rovnoběžně s konturou (<_FAL>)
									1 =	Přídavek rozměru pro opracování načisto odděleně (<_FALX>/<_FALZ>)

4.26.1.44 CYCLE951 - Oddělování třísky

Syntaxe

```
CYCLE951(<_SPD>, <_SPL>, <_EPD>, <_EPL>, <_ZPD>, <_ZPL>, <_LAGE>,
<_MID>, <_FALX>, <_FALZ>, <_VARI>, <_RF1>, <_RF2>, <_RF3>, <_SDIS>,
<_FF1>, <_NR>, <_DMODE>, <_AMODE>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	X0	<_SPD>	REAL	Souřadnice vztažného bodu, (abs, vždycky průměr)
2	Z0	<_SPL>	REAL	Vztažný bod (abs)
3	X1	<_EPD>	REAL	Koncový bod
4	Z1	<_EPL>	REAL	Koncový bod
5	XM α1 α2	<_ZPD>	REAL	Vnitřní bod, viz <_DMODE> (DESÍTKY)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
6	ZM α1 α2	<_ZPL>	REAL	Vnitřní bod, viz <_DMODE> (DESÍTKY)		
7	Poloha	<_LAGE>	INT	Poloha pro obráběného rohu	0 =	vnější/vzadu
					1 =	vnější/vpředu
					2 =	vnitřní/vzadu
					3 =	vnitřní/vpředu
8	D	<_MID>	REAL	Maximální přísuv do hloubky při zajíždění nástroje		
9	UX	<_FALX>	REAL	Přídavek rozměru pro opracování načisto ve směru X		
10	UZ	<_FALZ>	REAL	Přídavek rozměru pro opracování načisto ve směru Z		
11		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:	Směr oddělování třísky (podélné nebo příčné) v souřadném systému	
					1 =	podélný
					2 =	příčný
				DESÍTKY:		
					1 =	Obrábění nahrubo až na přídavek rozměru pro obrábění načisto
					2 =	Obrábění načisto
				STOVKY:	rezervováno	
				TISÍCE:	rezervováno	
				DESÍTKY TISÍC:	rezervováno	
12	R1/FS1	<_RF1>	REAL	Rádius zaoblení nebo šířka fasety 1, viz <_AMODE> (DESÍTKY TISÍC)		
13	R2/FS2	<_RF2>	REAL	Rádius zaoblení nebo šířka fasety 2, viz <_AMODE> (STOVKY TISÍC)		
14	R3/FS3	<_RF3>	REAL	Rádius zaoblení nebo šířka fasety 3, viz <_AMODE> (JEDNOTKY MILIONŮ)		
15	SC	<_SDIS>	REAL	Bezpečnostní vzdálenost		
16	F	<_FF1>	REAL	Posuv pro obrábění nahrubo/načisto		
17		<_NR>	INT	Identifikace způsobu oddělování třísky (odpovídá programovým tlačítkům ve svislém pruhu na obrazovce)		
					0 =	oddělování třísky 1, roh 90 stupňů bez faset/zaoblení
					1 =	oddělování třísky 2, roh 90 stupňů s fasetami/zaobleními
					2 =	oddělování třísky 3, libovolný roh s fasetami/zaobleními

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
18		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	Způsob zadávání parametrů <_ZPD>/<_ZPL>	
					0 =	Xm/Zm
					1 =	Xm/α1
					2 =	Xm/α2
					3 =	α1/Zm
					4 =	α2/Zm
					5 =	α1/α2
21		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Vnitřní bod, souřadnice X	
					0 =	absolutně, hodnota příčné osy se zadává jako průměr
					1 =	inkrementálně, hodnota příčné osy se zadává jako rádius
				DESÍTKY:	Vnitřní bod, souřadnice Z	
					0 =	absolutní
					1 =	inkrementální
				STOVKY:	Koncový bod, souřadnice X	
					0 =	absolutně, hodnota příčné osy se zadává jako průměr
					1 =	inkrementálně, hodnota příčné osy se zadává jako rádius
				TISÍCE:	koncový bod v ose Z	
					0 =	absolutní
					1 =	inkrementální
				DESÍTKY TISÍC:	Rádius / faseta 1	
					0 =	Rádius
					1 =	Faseta
				STOVKY TISÍC:	Rádius / faseta 2	
					0 =	Rádius
					1 =	Faseta
				JEDNOTKY MILIONŮ:	Rádius / faseta 3	
					0 =	Rádius
					1 =	Faseta

4.26.1.45 CYCLE952 - Oddělování třísky / odstraňování zbytkového materiálu oddělováním třísky / zapichování / zapichování zbytkového materiálu / soustružení zápichu / odstraňování zbytkového materiálu soustružením zápichu

Syntaxe

```
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>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
1	PRG	<_PRG>	STRING[100]	Název programu pro oddělování třísky
2	CON	<_CON>	STRING[100]	Název programu, z něhož se načítá aktualizovaná kontura surového obrobku (při obrábění zbytkového materiálu)
3	CONR	<_CONR>	STRING[100]	Název programu, ve kterém je popsána aktualizovaná kontura surového obrobku (viz <_AMODE>, DESÍTKY TISÍC)

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
4		<_VARI>	INT	Způsob opracování		
				JEDNOTKY:	Způsob opracování	
					1 =	podélný
					2 =	příčný
					3 =	rovnoběžně s konturou
				DESÍTKY:	Technologie opracování (viz parametr <_GMODE> (STOVKY)	
					1 =	Obrábění nahrubo
					2 =	Obrábění načisto
					3 =	rezervováno
					4 =	Obrábění nahrubo ve dvou knaálech
					5 =	Obrábění načisto ve dvou kanálech
				STOVKY:	Směr obrábění	
					1 =	Směr obrábění X -
					2 =	Směr obrábění X +
					3 =	Směr obrábění Z-
					4 =	směr obrábění Z +
				TISÍCE:	Směr přísuvu	
					1 =	Zvnějšku X -
					2 =	Zevnitř X +
					3 =	Čelní plocha Z -
					4 =	Zadní strana Z +
				DESÍTKY TISÍC:	Definice, platí jako přídavek rozměru pro opracování načisto	
					0 =	Přídavek rozměru pro opracování načisto UX a UZ odděleně
					1 =	Přídavek rozměru pro opracování načisto U rovnoběžně s konturou
				STOVKY TISÍC:	Vyjíždění podél kontury	
					0 =	kompatibilita, automatické vyjíždění podél kontury
					1 =	s vyjížděním podél kontury
					2 =	bez vyjíždění podél kontury
					3 =	automatické vyjíždění podél kontury
				JEDNOTKY MILIONŮ:	Podříznutí	
					0 =	místo se nevyhodnocuje při zapichování, obrábění zbytkového materiálu a soustružení zápichů, při obrábění zbytkového materiálu
					1 =	Podříznutí opracovat
					2 =	podříznutí neopracovat

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
				DESÍTKY MILIONŮ:	Před/za osou otáčení	
					0 =	obrábění před středem otáčení
					1 =	rezervováno
5	F	<_F>	REAL	Posuv pro obrábění nahrubo/načisto		
	FZ			Posuv ve směru abscisy při soustružení zápichů		
6	FR	<_FR>	REAL	Posuv pro zajíždění nástroje při obrábění podříznutí nahrubo		
	FX			Posuv ve směru ordináty při soustružení zápichů		
7	RP	<_RP>	REAL	Návratová rovina při vnitřním obrábění (abs, vždycky průměr)		
8	D	<_D>	REAL	Přísuv při obrábění nahrubo, (viz JEDNOTKY parametru <_AMODE>)		
9	DX	<_DX>	REAL	Přísuv ve směru X (viz JEDNOTKY parametru <_AMODE>)		
10	DZ	<_DZ>	REAL	Přísuv ve směru Z (viz JEDNOTKY parametru <_AMODE>)		
11	UX	<_UX>	REAL	Přídavek rozměru pro opracování načisto ve směru X (viz <_VARI> - DESÍTKY TISÍC)		
12	UZ	<_UZ>	REAL	Přídavek rozměru pro opracování načisto ve směru Z (viz <_VARI> - DESÍTKY TISÍC)		
13	U	<_U>	REAL	Přídavek rozměru pro obrábění načisto rovnoběžně s konturou (viz <_VARI> - DESÍTKY TISÍC)		
14	U1	<_U1>	REAL	Dodatečný přídavek rozměru pro opracování načisto (viz <_AMODE> - TISÍCE)		
15	BL	<_BL>	INT	Definice surového obrobku	1 =	válec s přídavkem rozměru
					2 =	přídavek rozměru na kontuře hotového obrobku
					3 =	kontura surového obrobku je zadána
16	XD	<_XD>	REAL	Definice surového obrobku ve směru X (viz <_AMODE> - STOVKY TISÍC)		
17	ZD	<_ZD>	REAL	Definice surového obrobku ve směru Z (viz <_AMODE> (MILIONY))		
18	XA	<_XA>	REAL	Hranice 1 ve směru X (abs, vždycky průměr)		
19	ZA	<_ZA>	REAL	Hranice 1 ve směru Z (abs)		
20	XB	<_XB>	REAL	Hranice 2 ve směru X (viz <_AMODE> - DESÍTKY MILIONŮ)		
21	ZB	<_ZB>	REAL	Hranice 2 ve směru Z (viz <_AMODE> - STOVKY MILIONŮ)		
22	XDA	<_XDA>	REAL	Hranice zápichu 1 pro pozici 1. zápichu na čelní straně (abs, vždy průměr)		
23	XDB	<_XDB>	REAL	Hranice zápichu 2 pro pozici 1. zápichu na čelní straně (abs, vždy průměr)		
24	N	<_N>	INT	Počet zápichů		
25	DP	<_DP>	REAL	Vzdálenost mezi zápichy	Podélný zápich: rovnoběžně s osou Z	
					Příčný zápich: rovnoběžně s osou X	
26	DI	<_DI>	REAL	Vzdálenost pro přerušení posuvu	0 =	Žádné přerušení
					> 0 =	s přerušením
27	SC	<_SC>	REAL	Bezpečnostní vzdálenost pro objíždění překážky, inkrementálně		
28	D2	<_DN>	INT	D-číslo pro 2. břit nástroje, pokud není naprogramováno ⇒ D+1		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
29		<_GMODE>	INT	Geometrický režim (vyhodnocování naprogramované geometrické hodnoty)		
				JEDNOTKY:	rezervováno	
				DESÍTKY:	rezervováno	
				STOVKY:	Volba opracování/pouze výpočet počátečního bodu	
					0 =	normální opracování (není zapotřebí žádný režim kompatibility)
					1 =	normální opracování
					2 =	výpočet počáteční polohy - žádné obrábění (jen pro vyvolání ze systémů ShopMill/ShopTurn)
				TISÍCE:	Ohraničení	
					0 =	ne
					1 =	ano
				DESÍTKY TISÍC:	Zadat hranici 1 ve směru X	
					0 =	ne
					1 =	ano
				STOVKY TISÍC:	Zadat hranici 2 ve směru X	
					0 =	ne
					1 =	ano
				JEDNOTKY MILIONŮ:	Zadat hranici 1 ve směru Z	
					0 =	ne
					1 =	ano
				DESÍTKY MILIONŮ:	Zadat hranici 2 ve směru Z	
					0 =	ne
					1 =	ano

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
30		<_DMODE>	INT	Režim zobrazování		
				JEDNOTKY:	Rovina obrábění G17/G18/G19	
					0 =	kompatibilita, zůstává aktivní rovina nastavená před voláním cyklu
					1 =	G17 (aktivní jen v cyklu)
					2 =	G18 (aktivní jen v cyklu)
					3 =	G19 (aktivní jen v cyklu)
				DESÍTKY:	technologický režim	
					1 =	Obrábění kontury
					2 =	Zápichy na kontuře
					3 =	Soustružení zápichů
				STOVKY:	Odstraňování zbytkového materiálu	
					0 =	ne
					1 =	ano
				TISÍCE:	---	rezervováno
				DESÍTKY TISÍC:	Odstupňování technologie na obrazovkách cyklů (Strana 1169)	
					0 =	Zadání: kompletní
					1 =	Zadání: jednoduché
				STOVKY TISÍC:	Název programu automaticky	
					0 =	ne
					1 =	ano

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
31		<_AMODE>	INT	Alternativní režim		
				JEDNOTKY:	Volba přísuvu	
					0 =	přísuv DX a DZ při druhu obrábění rovnoběžně s konturou
					1 =	Přísuv D
				DESÍTKY:	Strategie pro přísuv	
					0 =	proměnná hloubka řezu (90 ... 100%)
					1 =	konstantní hloubka průchodu nástroje
				STOVKY:	Rozdělení řezů	
					0 =	rovnoměrné
					1 =	optimalizace na hranách
				TISÍCE:	Volba přídavku rozměru na kontuře U1, dvojí obrábění načisto	
					0 =	ne
					1 =	ano
				DESÍTKY TISÍC:	Aktualizace volby surového obrobku	
					0 =	ne
					1 =	ano
				STOVKY TISÍC:	Volba přídavku rozměru na surovém obrobku XD	
					0 =	absolutně, hodnota příčné osy se zadává jako průměr
					1 =	inkrementálně, hodnota příčné osy se zadává jako rádius
				JEDNOTKY MILIONŮ:	Volba přídavku rozměru na surovém obrobku ZD	
					0 =	absolutní
					1 =	inkrementální
				DESÍTKY MILIONŮ:	Volba hranice 2 XB	
					0 =	absolutně, hodnota příčné osy se zadává jako průměr
					1 =	inkrementálně, hodnota příčné osy se zadává jako rádius
				STOVKY MILIONŮ:	Volba hranice 2 ZB	
					0 =	absolutní
					1 =	inkrementální
				JEDNOTKY MILIARD:		
					0 =	Řídící kanál
					1 =	Vlečný kanál
32		<_PK>	INT	Číslo partnerského kanálu, jestliže jsou na stroji k dispozici více než 2 kanály		

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam
33	DCH	<_DCH>	REAL	Kanálové posunutí
34	FS	<_FS>	REAL	Posuv při obrábění načisto v rámci kompletního opracování

4.26.1.46 CYCLE4071 - Podélné broušení s přísuvem v bodu obratu

Syntaxe

```
CYCLE4071(<S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>, <S_H>, <S_A1>,
<S_A2>)
```

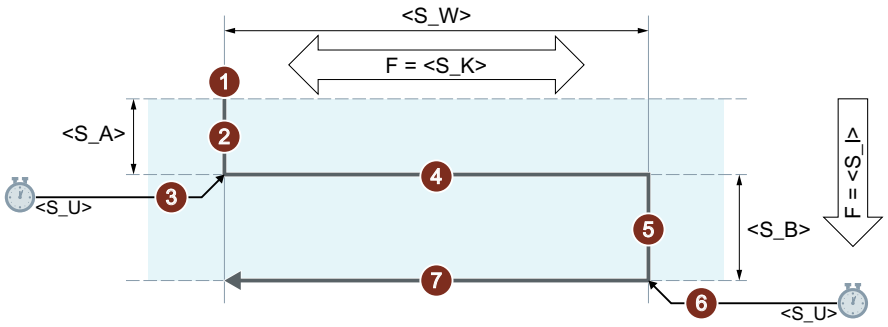
Parametry

Č.	Parametry	Datový typ	Význam
1	<S_A>	REAL	Přísuvná hloubka na začátku
2	<S_B>	REAL	Přísuvná hloubka na konci
3	<S_W>	REAL	Broušená šířka
4	<S_U>	REAL	Doba vyjiskřování
5	<S_I>	REAL	Posuv pro přísuv
6	<S_K>	REAL	Posuv pro přísuv v příčném směru
7	<S_H>	INT	Počet opakování
8	<S_A1>	AXIS	Přísuvná osa (nepovinné), příp. 1. geometrická osa
9	<S_A2>	AXIS	Oscilační osa (nepovinné), příp. 2. geometrická osa

Funkce

Tento cyklus slouží pro zpracování opakujících se přísuvů. Přísuvná hloubka na začátku a na konci se přitom mohou lišit. Mezi přísuvy se uskutečňuje tangenciální pohyb.

Postup



① Spuštění cyklu na momentální pozici oscilační osy.
② Pohyb přísuvné osy na přísuvnou hloubku na začátku <S_A> s posuvem pro přísuv <S_I>.
③ Vyjiskřování s dobou vyjiskřování <S_U>.
④ Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce <S_W> s posuvem pro přísuv v příčném směru <S_K>.
⑤ Pohyb přísuvné osy na přísuvnou hloubku na konci <S_B> s posuvem pro přísuv <S_I>.
⑥ Vyjiskřování s dobou vyjiskřování <S_U>.
⑦ Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce <S_W> na počáteční bod a s posuvem pro přísuv v příčném směru <S_K>.

Označení kroků pracovního postupu, které se opakují.
Operace se opakuje tak dlouho, dokud není dosaženo naprogramovaného počtu opakování <S_H>.

Poznámka

Tento postup při zpracování blok po bloku není možné přerušit.

Příklad

Uskutečněte dva oscilační pohyby s následujícími parametry cyklu:

- Přísuvná hloubka na začátku: 0,02 mm
- Přísuvná hloubka na konci: 0,01 mm
- Zdvih: 100 mm
- Doba vyjiskřování: 1 s
- Posuv pro přísuv: 1 mm/min
- Posuv v příčném směru: 1000 mm/min
- Opakování: 2
- Oscilační a přísuvná osa: Standardní geometrické osy

Programový kód

```
N10 T1 D1
```

Programový kód

```
N20 CYCLE4071(0.02,0.01,100,1,1,1000,2)
N30 M30
```

4.26.1.47 CYCLE4072 - Podélné broušení s přísuvem v bodě obratu a se signálem pro přerušení

Syntaxe

```
CYCLE4072(<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>,
<S_H>, <S_A1>, <S_A2>)
```

Parametry

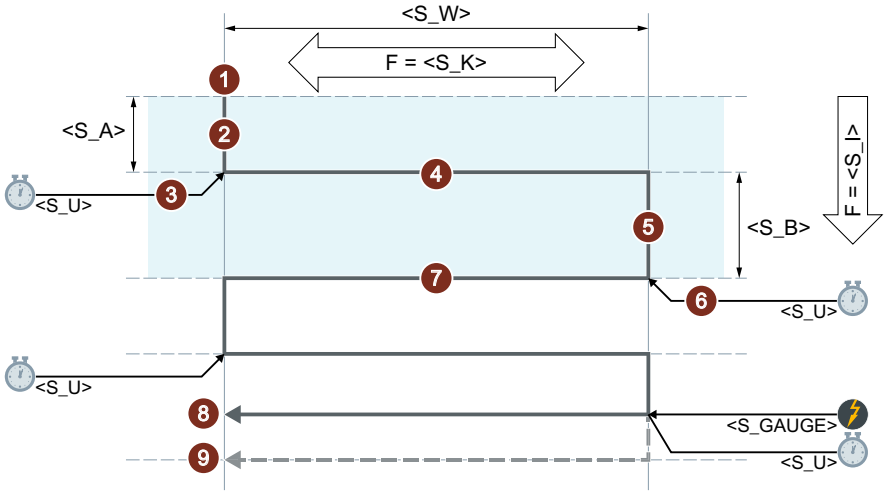
Č.	Parametry	Datový typ	Význam
1	<S_GAUGE>	STRING	Podmínky přerušení pro přísuv: 1. Číslo rychlého vstupu 2. Logický výraz
2	<S_A>	REAL	Přísuvná hloubka na začátku
3	<S_B>	REAL	Přísuvná hloubka na konci
4	<S_W>	REAL	Broušená šířka
5	<S_U>	REAL	Doba vyjiskřování
6	<S_I>	REAL	Posuv pro přísuv
7	<S_K>	REAL	Posuv pro přísuv v příčném směru
8	<S_H>	INT	Počet opakování
9	<S_A1>	AXIS	Přísuvná osa (nepovinné), příp. 1. geometrická osa
10	<S_A2>	AXIS	Oscilační osa (nepovinné), příp. 2. geometrická osa

Funkce

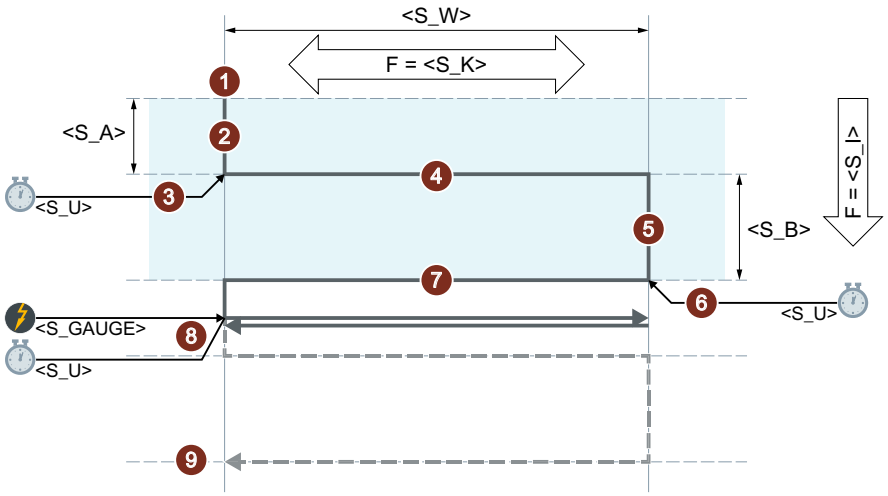
Tento cyklus slouží pro zpracování opakujících se přísuvů, přičemž se vyhodnocuje externí přerušovací signál. Přísuvné hloubky na začátku a na konci se mohou lišit. Mezi přísuvy se uskutečňuje tangenciální pohyb. Když je podmínka pro přerušení splněna, přísuv do hloubky se přeruší. Po přerušení přísuvu do hloubky se vždy uskuteční jeden kompletní zdvih nástroje.

Postup

Přerušení přísuvu na konci



Přerušení přísuvu na začátku



① Spuštění cyklu na momentální pozici oscilační osy.
② Pohyb přísuvné osy na přísuvnou hloubku na začátku <S_A> s posuvem pro přísuv <S_I>.
③ Vyjiskřování s dobou vyjiskřování <S_U>.
④ Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce <S_W> s posuvem pro přísuv v příčném směru <S_K>.
⑤ Pohyb přísuvné osy na přísuvnou hloubku na konci <S_B> s posuvem pro přísuv <S_I>.
⑥ Vyjiskřování s dobou vyjiskřování <S_U>.
⑦ Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce <S_W> na počáteční bod a s posuvem pro přísuv v příčném směru <S_K>.
⑧ Přerušovací signál: Po dosažení následujícího počátečního bodu se opracování ukončí.
⑨ Bez přerušovacího signálu: Operace se opakuje tak dlouho, dokud není dosaženo naprogramovaného počtu opakování <S_H>.

Označení kroků pracovního postupu, které se opakují.

Poznámka

Tento postup při zpracování blok po bloku není možné přerušit.

Zdroje

Jako zdroje cyklus používá synchronní akci překračující hranice bloků a proměnnou synchronní akce. Synchronní akce se dynamicky určuje z volné oblasti ve svazku synchronních akcí (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR – 1199 ...). Jako proměnná synchronní akce se používá SYG_IS[1].

Příklady

Příklad 1: Oscilační pohyb se dvěma zdvihy:

Parametr cyklu:

- Přísuvná hloubka na začátku: 0,02 mm
- Přísuvná hloubka na konci: 0,01 mm
- Zdvih: 100 mm
- Doba vyjiskřování: 1 s
- Posuv pro přísuv: 1 mm/min
- Posuv v příčném směru: 1000 mm/min
- Opakování: 2
- Oscilační a přísuvná osa: Standardní geometrické osy

Přerušovací signál: rychlý vstup 1 ($A_IN[1])

Programový kód

```
N10 T1 D1
N20 CYCLE4072("1",0.02,0.01,100,1,1,1000,2)
N30 M30
```

Příklad 2: Oscilační pohyb se dvěma zdvihy:

Parametr cyklu:

- Přísuvná hloubka na začátku: 0,02 mm
- Přísuvná hloubka na konci: 0,01 mm
- Zdvih: 100 mm
- Doba vyjiskřování: 1 s
- Posuv pro přísuv: 1 mm/min
- Posuv v příčném směru: 1000 mm/min
- Opakování: 2
- Oscilační a přísuvná osa: Standardní geometrické osy

Přerušovací signál: Proměnná $A_DBR[20] < 0,01

Programový kód

```
N10 T1 D1
N20 CYCLE4072("($A_DBR[20]<0.01)",0.02,0.01,100,1,1,1000,2)
N30 M30
```

4.26.1.48 CYCLE4073 - Podélné broušení se spojitým přísuvem

Syntaxe

```
CYCLE4073(<S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>, <S_A1>, <S_A2>)
```

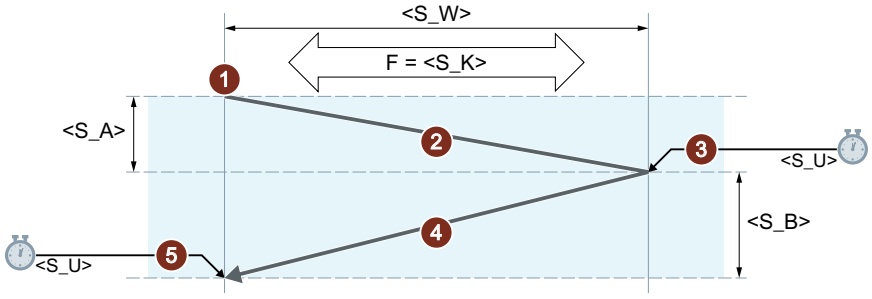
Parametry

Č.	Parametry	Datový typ	Význam
1	<S_A>	REAL	Přísuvná hloubka na začátku
2	<S_B>	REAL	Přísuvná hloubka na konci
3	<S_W>	REAL	Broušená šířka
4	<S_U>	REAL	Doba vyjiskřování
5	<S_K>	REAL	Posuv pro přísuv v příčném směru
6	<S_H>	INT	Počet opakování
7	<S_A1>	AXIS	Přísuvná osa (nepovinné), příp. 1. geometrická osa
8	<S_A2>	AXIS	Oscilační osa (nepovinné), příp. 2. geometrická osa

Funkce

Tento cyklus slouží pro zpracování opakujících se přísuvů. Přísuv od začátku ke konci a od konce na začátek se mohou lišit.

Postup



① Spuštění cyklu na momentální pozici oscilační osy s přísuvnou hloubkou 0

② Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce <S_W> a s posuvem pro přísuv v příčném směru <S_K> se spojitým zvětšováním přísuvné hloubky až po přísuvnou hloubku na začátku <S_A>.

③ Vyjiskřování s dobou vyjiskřování <S_U>.

④ Pohyb oscilační osy na počáteční bod s dráhou posuvu odpovídající broušené šířce <S_W> a s posuvem pro přísuv v příčném směru <S_K> se spojitým zvětšováním přísuvné hloubky až po přísuvnou hloubku na konci <S_B>.

⑤ Vyjiskřování s dobou vyjiskřování <S_U>.

Označení kroků pracovního postupu, které se opakují.

Operace se opakuje tak dlouho, dokud není dosaženo naprogramovaného počtu opakování <S_H>.

Poznámka

Tento postup při zpracování blok po bloku není možné přerušit.

Příklad

Oscilační pohyb se dvěma zdvihy:

Parametr cyklu:

- Přísuvná hloubka na začátku: 0,02 mm
- Přísuvná hloubka na konci: 0,01 mm
- Zdvih: 100 mm
- Doba vyjiskřování: 1 s
- Posuv v příčném směru: 1000 mm/min
- Opakování: 2
- Oscilační a přísuvná osa: Standardní geometrické osy

Programový kód

```
N10 T1 D1
N20 CYCLE4073(0.02,0.01,100,1,1000,2)
N30 M30
```

4.26.1.49 CYCLE4074 - Podélné broušení se spojitým přísuvem a se signálem pro přerušení

Syntaxe

```
CYCLE4074(<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>,
<S_A1>, <S_A2>)
```

Parametry

Č.	Parametry	Datový typ	Význam
1	<S_GAUGE>	STRING	Podmínky přerušení pro přísuv: 1. Číslo rychlého vstupu 2. Logický výraz
2	<S_A>	REAL	Přísuvná hloubka na začátku
3	<S_B>	REAL	Přísuvná hloubka na konci
4	<S_W>	REAL	Broušená šířka
5	<S_U>	REAL	Doba vyjiskřování
6	<S_K>	REAL	Posuv pro přísuv v příčném směru
7	<S_H>	INT	Počet opakování

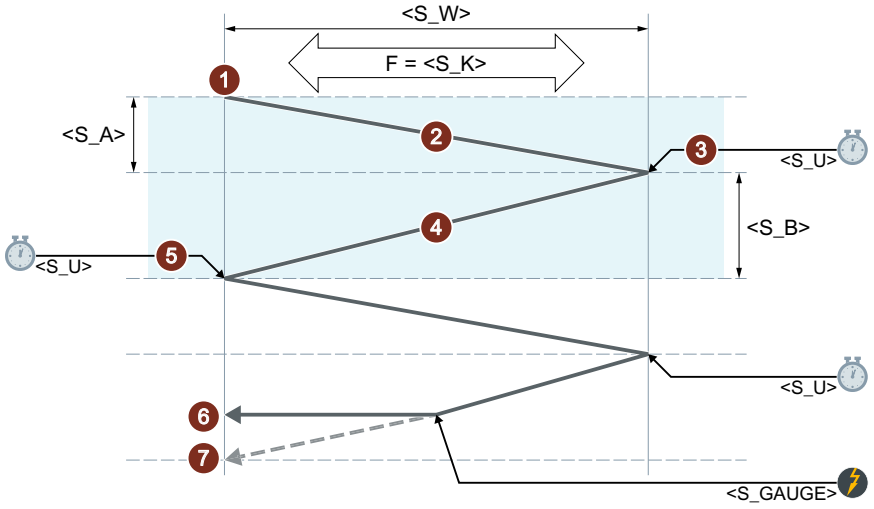
Č.	Parametry	Datový typ	Význam
8	<S_A1>	AXIS	Přísuvná osa (nepovinné), příp. 1. geometrická osa
9	<S_A2>	AXIS	Oscilační osa (nepovinné), příp. 2. geometrická osa

Funkce

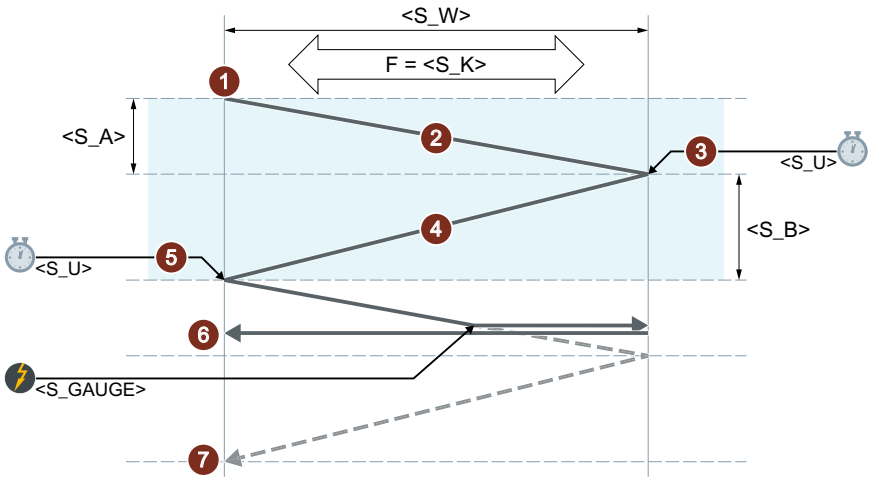
Tento cyklus slouží pro zpracování opakujících se přísuvů, přičemž se vyhodnocuje např. externí přerušovací signál. Přísuvné hloubky na začátku a na konci se mohou lišit. Když je podmínka pro přerušení splněna, přísuv do hloubky se přeruší. Po přerušení přísuvu do hloubky se vždy uskuteční jeden kompletní zdvih nástroje.

Postup

Přerušení přísuvu od konce k začátku



Přerušení přísuvu od začátku ke konci



① Spuštění cyklu na momentální pozici oscilační osy s přísuvnou hloubkou 0

② Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce `<S_W>` a s posuvem pro přísuv v příčném směru `<S_K>` se spojitým zvětšováním přísuvné hloubky až po přísuvnou hloubku na začátku `<S_A>`.

③ Vyjiskřování s dobou vyjiskřování `<S_U>`.

④ Pohyb oscilační osy na počáteční bod s dráhou posuvu odpovídající broušené šířce `<S_W>` a s posuvem pro přísuv v příčném směru `<S_K>` se spojitým zvětšováním přísuvné hloubky až po přísuvnou hloubku na konci `<S_B>`.

⑤ Vyjiskřování s dobou vyjiskřování `<S_U>`.

⑥ Přerušovací signál: Přísuv do hloubky se přeruší. Po dosažení následujícího počátečního bodu se opracování ukončí.

⑦ Bez přerušovacího signálu: Operace se opakuje tak dlouho, dokud není dosaženo naprogramovaného počtu opakování `<S_H>`.

Označení kroků pracovního postupu, které se opakují.

Poznámka

Tento postup při zpracování blok po bloku není možné přerušit.

Zdroje

Jako zdroje cyklus používá synchronní akci překračující hranice bloků a proměnnou synchronní akce. Synchronní akce se dynamicky určuje z volné oblasti ve svazku synchronních akcí (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR – 1199 ...). Jako proměnná synchronní akce se používá SYG_IS[1].

Příklady

Příklad 1: Oscilační pohyb se dvěma zdvihy:

Parametr cyklu:

- Přísuvná hloubka na začátku: 0,02 mm
- Přísuvná hloubka na konci: 0,01 mm
- Zdvih: 100 mm
- Doba vyjiskřování: 1 s
- Posuv v příčném směru: 1000 mm/min
- Opakování: 2
- Oscilační a přísuvná osa: Standardní geometrické osy

Přerušovací signál: rychlý vstup 1 ($A_IN[1])

Programový kód

```
N10 T1 D1
N20 CYCLE4074("1",0.02,0.01,100,1,1000,2)
N30 M30
```

Příklad 2: Oscilační pohyb se dvěma zdvihy:

Parametr cyklu:

- Přísuvná hloubka na začátku: 0,02 mm
- Přísuvná hloubka na konci: 0,01 mm
- Zdvih: 100 mm
- Doba vyjiskřování: 1 s
- Posuv v příčném směru: 1000 mm/min
- Opakování: 2
- Oscilační a přísuvná osa: Standardní geometrické osy

Přerušovací signál: Proměnná $A_DBR[20] < 0,01

Programový kód

```
N10 T1 D1
N20 CYCLE4074("($A_DBR[20]<0.01)",0.02,0.01,100,1,1000,2)
N30 M30
```

4.26.1.50 CYCLE4075 - Plošné broušení s přísuvem v bodě obratu

Syntaxe

```
CYCLE4075(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>, <S_A1>,
<S_A2>)
```

Parametry

Č.	Parametry	Datový typ	Význam
1	<S_I>	REAL	Přísuvná hloubka na začátku
2	<S_J>	REAL	Přísuvná hloubka na konci
3	<S_K>	REAL	Celková přísuvná hloubka
4	<S_A>	REAL	Broušená šířka
5	<S_R>	REAL	Posuv pro přísuv
6	<S_F>	REAL	Posuv pro přísuv v příčném směru
7	<S_P>	REAL	Doba vyjiskřování
8	<S_A1>	AXIS	Přísuvná osa (nepovinné)
9	<S_A2>	AXIS	Osa vykonávající pohyb tam a zpět (nepovinné)

Funkce

Tento cyklus slouží pro opracování s celkovou hloubkou přísuvu v několika přísuvných krocích. Přísuvné hloubky na začátku a na konci se mohou lišit. Mezi přísuvy se uskutečňuje tangenciální pohyb.

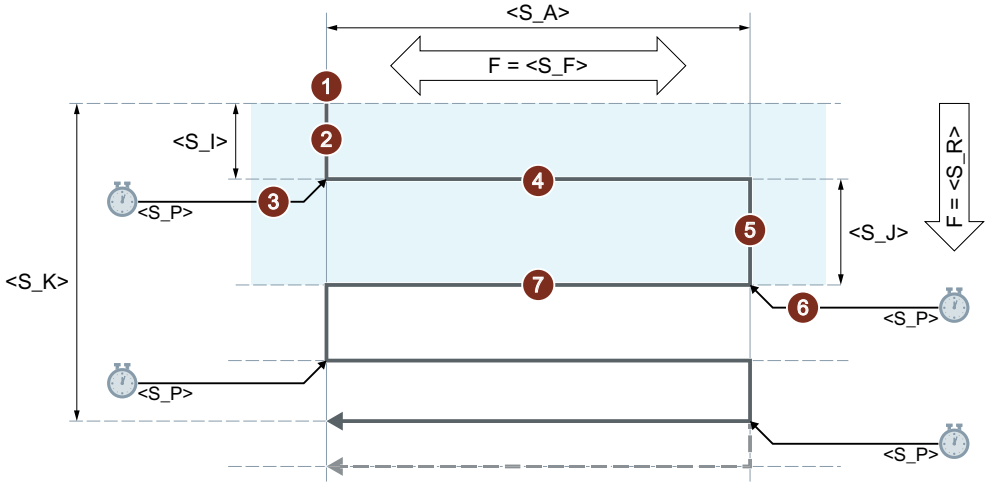
Údaje dráhy P1 až P4 mohou být kladné nebo záporné.

Zadání přísuvné osy a/nebo oscilační osy jsou nepovinné. Jestliže jeden, příp. oba tyto parametry nejsou zadány, cyklus použije první z obou geometrických os kanálu.

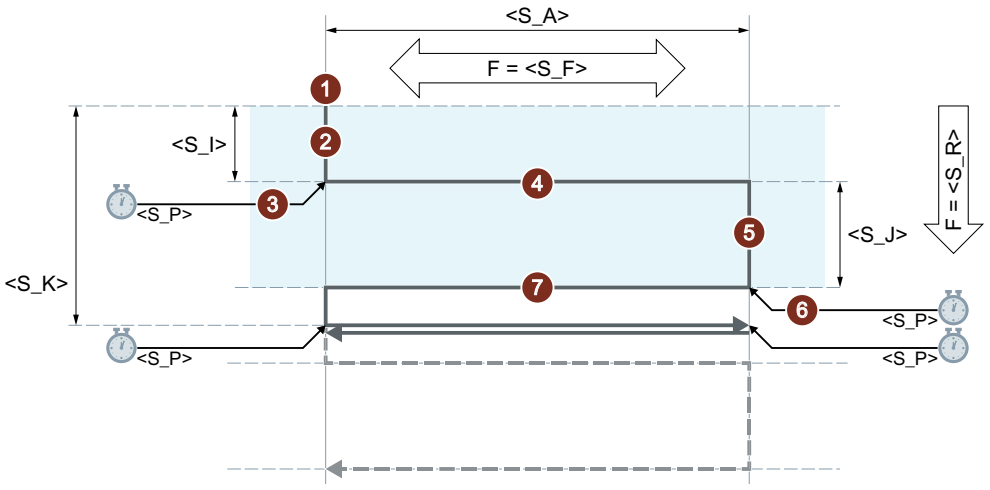
Pokud je součet přísuvné hloubky na začátku a na konci roven 0, příp. pokud se celková přísuvná hloubka rovná 0, uskuteční se pouze jeden vyjiskřovací zdvih.

Postup

Celková přísuvná hloubka dosažená ve druhém bodu obratu



Celková přísuvná hloubka dosažená v prvním bodu obratu



① Spuštění cyklu na momentální pozici oscilační osy.
② Pohyb přísuvné osy na přísuvnou hloubku na začátku <S_I> s posuvem pro přísuv <S_R>.
③ Vyjiskřování s dobou vyjiskřování <S_P>.
④ Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce <S_A> s posuvem pro přísuv v příčném směru <S_F>.
⑤ Pohyb přísuvné osy na přísuvnou hloubku na konci <S_J> s posuvem pro přísuv <S_R>.
⑥ Vyjiskřování s dobou vyjiskřování <S_P>.
⑦ Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce <S_A> na počáteční bod a s posuvem pro přísuv v příčném směru <S_F>.

Označení kroků pracovního postupu, které se opakují.
Tato operace se opakuje tak dlouho, dokud není dosaženo celkové hloubky přísuvu <S_K>. Poslední zdvih je potom rovnoměrně rozdělen.

Poznámka

Tento postup při zpracování blok po bloku není možné přerušit.

Příklad

Oscilační pohyb s těmito parametry:

- Přísuvná hloubka na začátku 0,02 mm
- Přísuvná hloubka na konci 0,01 mm
- Celková přísuvná hloubka 1 mm
- Zdvih 100 mm
- Posuv při přísuvu 1 mm/min
- Přísuv v příčném směru 1000 mm/min
- Doba vyjiskřování 1 sekunda
- Standardní geometrické osy

Programový kód

```
N10 T1 D1
N20 CYCLE4075(0.02,0.01,1,100,1,1000,1)
N30 M30
```

4.26.1.51 CYCLE4077 - Plošné broušení s přísuvem na bod obratu a se signálem pro přerušení

Syntaxe

```
CYCLE4077(<S_GAUGE>, <S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>,
<S_P>, <S_A1>, <S_A2>)
```

Parametry

Č.	Parametry	Datový typ	Význam
1	<S_GAUGE>	STRING	Podmínka pro přerušení přísuvu: • Číslo rychlého vstupu • Logický výraz
2	<S_I>	REAL	Přísuvná hloubka na začátku
3	<S_J>	REAL	Přísuvná hloubka na konci
4	<S_K>	REAL	Celková přísuvná hloubka
5	<S_A>	REAL	Broušená šířka
6	<S_R>	REAL	Posuv pro přísuv
7	<S_F>	REAL	Posuv pro přísuv v příčném směru
8	<S_P>	REAL	Doba vyjiskřování
9	<S_A1>	AXIS	Přísuvná osa (nepovinné)
10	<S_A2>	AXIS	Osa vykonávající pohyb tam a zpět (nepovinné)

Funkce

Tento cyklus slouží pro opracování s celkovou hloubkou přísuvu v několika přísuvných krocích. Přísuvné hloubky na začátku a na konci se mohou lišit. Mezi přísuvy se uskutečňuje tangenciální pohyb. Když se přerušovací signál na rychlém vstupu rovná 1 nebo je-li splněna podmínka pro přerušení, přísuv do hloubky se přeruší. Po přerušení se uskuteční jeden kompletní zdvih nástroje.

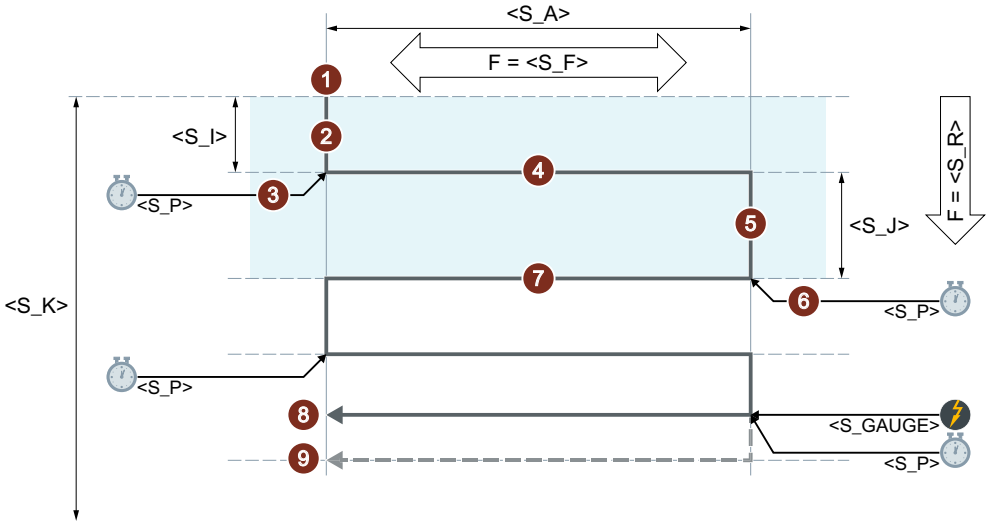
Údaje dráhy P2 až P5 mohou být kladné nebo záporné.

Zadání přísuvné osy a/nebo oscilační osy jsou nepovinné. Jestliže jeden, příp. oba tyto parametry nejsou zadány, cyklus použije první z obou geometrických os kanálu.

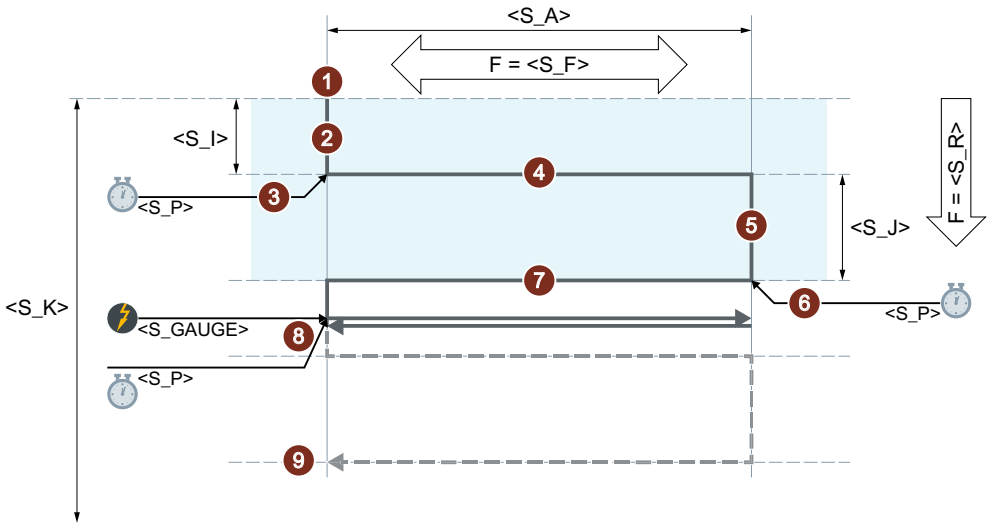
Pokud je součet přísuvné hloubky na začátku a na konci roven 0, příp. pokud se celková přísuvná hloubka rovná 0, uskuteční se pouze jeden vyjiskřovací zdvih.

Postup

Přerušení přísuvu na konci



Přerušení přísuvu na začátku



① Spuštění cyklu na momentální pozici oscilační osy.
② Pohyb přísuvné osy na přísuvnou hloubku na začátku <S_I> s posuvem pro přísuv <S_R>.
③ Vyjiskřování s dobou vyjiskřování <S_P>.
④ Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce <S_A> s posuvem pro přísuv v příčném směru <S_F>.
⑤ Pohyb přísuvné osy na přísuvnou hloubku na konci <S_J> s posuvem pro přísuv <S_R>.
⑥ Vyjiskřování s dobou vyjiskřování <S_P>.
⑦ Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce <S_A> na počáteční bod a s posuvem pro přísuv v příčném směru <S_F>.
⑧ Přerušovací signál: Po dosažení následujícího počátečního bodu se opracování ukončí.
⑨ Bez přerušovacího signálu: Tato operace se opakuje tak dlouho, dokud není dosaženo celkové hloubky přísuvu <S_K>. Poslední zdvih je potom rovnoměrně rozdělen.

Označení kroků pracovního postupu, které se opakují.

Poznámka

Tento postup při zpracování blok po bloku není možné přerušit.

Zdroje

Jako zdroje cyklus používá synchronní akci překračující hranice bloků a proměnnou synchronní akce. Synchronní akce se dynamicky určuje z volné oblasti ve svazku synchronních akcí (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR – 1199 ...). Jako proměnná synchronní akce se používá SYG_IS[1].

Příklady

Příklad 1

Oscilační pohyb s těmito parametry:

- Přísuvná hloubka na začátku 0,02 mm
- Přísuvná hloubka na konci 0,01 mm
- Celková přísuvná hloubka 1 mm
- Zdvih 100 mm
- Posuv při přísuvu 1 mm/min
- Přísuv v příčném směru 1000 mm/min
- Doba vyjiskřování 1 sekunda
- Standardní geometrické osy

Přerušovací signál: rychlý vstup 1 ($A_IN[1])

Programový kód

```
N10 T1 D1
N20 CYCLE4077("1",0.02,0.01,1,100,1,1000,1)
N30 M30
```

Příklad 2

Oscilační pohyb s těmito parametry:

- Přísuvná hloubka na začátku 0,02 mm
- Přísuvná hloubka na konci 0,01 mm
- Celková přísuvná hloubka 1 mm
- Zdvih 100 mm
- Posuv při přísuvu 1 mm/min
- Přísuv v příčném směru 1000 mm/min
- Doba vyjiskřování 1 sekunda
- Standardní geometrické osy

Přerušovací signál: Proměnná z RAM se dvěma porty 20 je menší než 0,01 ($A_DBR[20] < 0,01)

Programový kód

```
N10 T1 D1
N20 CYCLE4077("($A_DBR[20]<0.01)",0.02,0.01,1,100,1,1000,1)
N30 M30
```

4.26.1.52 CYCLE4078 - Plošné broušení se spojitým přísuvem

Syntaxe

```
CYCLE4078(<S_I>, <S_J>, <S_K>, <S_A>, <S_F>, <S_P>, <S_A1>, <S_A2>)
```

Parametry

Č.	Parametry	Datový typ	Význam
1	<S_I>	REAL	Přísuvná hloubka od začátku ke konci
2	<S_J>	REAL	Přísuvná hloubka od konce k začátku
3	<S_K>	REAL	Celková přísuvná hloubka
4	<S_A>	REAL	Broušená šířka
5	<S_F>	REAL	Posuv
6	<S_P>	REAL	Doba vyjiskřování
7	<S_A1>	AXIS	Přísuvná osa (nepovinné)
8	<S_A2>	AXIS	Osa vykonávající pohyb tam a zpět (nepovinné)

Funkce

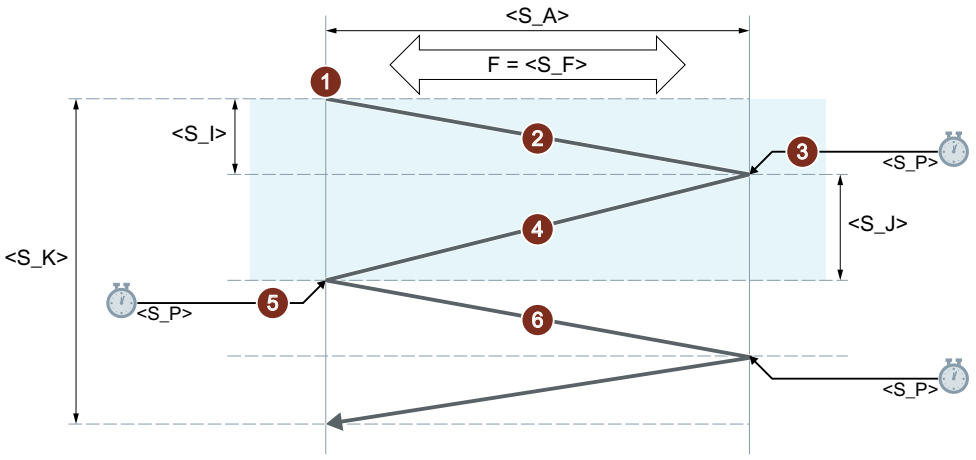
Tento cyklus slouží pro opracování s celkovou hloubkou přísuvu prostřednictvím spojitého přísuvu. Přísuvné hloubky od začátku ke konci a od konce k začátku se mohou lišit.

Údaje dráhy P1 až P4 mohou být kladné nebo záporné.

Zadání přísuvné osy a/nebo oscilační osy jsou nepovinné. Jestliže jeden, příp. oba tyto parametry nejsou zadány, cyklus použije první z obou geometrických os kanálu.

Pokud je součet přísuvných hloubek P1 a P2 roven 0, příp. pokud se celková přísuvná hloubka rovná 0, uskuteční se pouze jeden vyjiskřovací zdvih.

Postup



① Spuštění cyklu na momentální pozici oscilační osy s přísuvnou hloubkou 0

② Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce <S_A> a s posuvem <S_F> se spojitým zvětšováním přísuvné hloubky až po přísuvnou hloubku na začátku <S_I>.

③ Vyjiskřování s dobou vyjiskřování <S_P>.

④ Pohyb oscilační osy na počáteční bod s dráhou posuvu odpovídající broušené šířce <S_A> a s posuvem <S_F> se spojitým zvětšováním přísuvné hloubky až po přísuvnou hloubku na konci <S_J>.

⑤ Vyjiskřování s dobou vyjiskřování <S_P>.

⑥ Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce <S_A> na počáteční bod a s posuvem <S_F>.

Označení kroků pracovního postupu, které se opakují.

Tato operace se opakuje tak dlouho, dokud není dosaženo celkové hloubky přísuvu <S_K>. Poslední zdvih je potom rovnoměrně rozdělen.

Poznámka

Tento postup při zpracování blok po bloku není možné přerušit.

Příklad

Oscilační pohyb s těmito parametry:

- Přísuvná hloubka na začátku 20 mm
- Přísuvná hloubka na konci 10 mm
- Celková přísuvná hloubka 100 mm
- Zdvih 100 mm
- Posuv 1000 mm/min
- Doba vyjiskřování 1 sekunda
- Standardní geometrické osy

Programový kód

```
N10 T1 D1
N20 CYCLE4078(20,10,100,100,1000,1)
N30 M30
```

4.26.1.53 CYCLE4079 - Plošné broušení s přerušovaným přísuvem

Syntaxe

```
CYCLE4079(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>, <S_A1>,
<S_A2>)
```

Parametry

Č.	Parametry	Datový typ	Význam
1	<S_I>	REAL	Přísuvná hloubka na začátku
2	<S_J>	REAL	Přísuvná hloubka na konci
3	<S_K>	REAL	Celková přísuvná hloubka
4	<S_A>	REAL	Broušená šířka
5	<S_R>	REAL	Posuv pro přísuv
6	<S_F>	REAL	Posuv pro přísuv v příčném směru
7	<S_P>	REAL	Doba vyjiskřování
8	<S_A1>	AXIS	Přísuvná osa (nepovinné)
9	<S_A2>	AXIS	Osa vykonávající pohyb tam a zpět (nepovinné)

Funkce

Tento cyklus slouží pro opracování s celkovou hloubkou přísuvu v několika přísuvných krocích. Přísuvné hloubky na začátku a na konci se mohou lišit. Mezi přísuvy se uskutečňuje tangenciální pohyb.

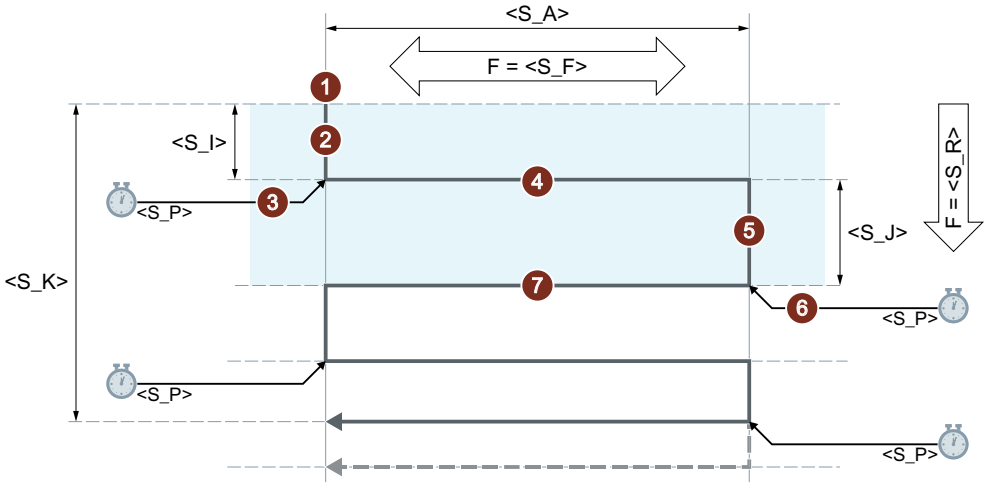
Údaje dráhy P1 až P4 mohou být kladné nebo záporné.

Zadání přísuvné osy a/nebo oscilační osy jsou nepovinné. Jestliže jeden, příp. oba tyto parametry nejsou zadány, cyklus použije první z obou geometrických os kanálu.

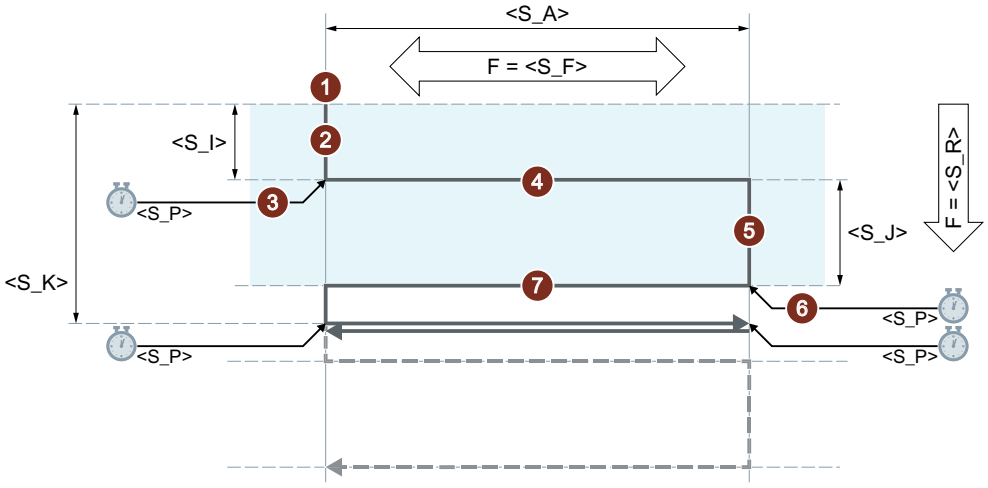
Pokud je součet přísuvné hloubky na začátku a na konci roven 0, příp. pokud se celková přísuvná hloubka rovná 0, uskuteční se pouze jeden vyjiskřovací zdvih.

Postup

Celková přísuvná hloubka dosažená ve druhém bodu obratu



Celková přísuvná hloubka dosažená v prvním bodu obratu



① Spuštění cyklu na momentální pozici oscilační osy.

② Pohyb přísuvné osy na přísuvnou hloubku na začátku <S_I> s posuvem pro přísuv <S_R>.

③ Vyjiskřování s dobou vyjiskřování <S_P>.

④ Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce <S_A> s posuvem pro přísuv v příčném směru <S_F>.

⑤ Pohyb přísuvné osy na přísuvnou hloubku na konci <S_J> s posuvem pro přísuv <S_R>.

⑥ Vyjiskřování s dobou vyjiskřování <S_P>.

⑦ Pohyb oscilační osy s dráhou posuvu odpovídající broušené šířce <S_A> na počáteční bod a s posuvem pro přísuv v příčném směru <S_F>.

Označení kroků pracovního postupu, které se opakují.

Tato operace se opakuje tak dlouho, dokud není dosaženo celkové hloubky přísuvu <S_K>. Poslední zdvih je potom rovnoměrně rozdělen.

Poznámka

Tento postup při zpracování blok po bloku není možné přerušit.

Příklad

Oscilační pohyb s těmito parametry:

- Přísuvná hloubka na začátku 0,02 mm
- Přísuvná hloubka na konci 0,01 mm
- Celková přísuvná hloubka 1 mm
- Zdvih 100 mm
- Posuv při přísuvu 1 mm/min
- Přísuv v příčném směru 1000 mm/min
- Doba vyjiskřování 1 sekunda
- Standardní geometrické osy

Programový kód

```
N10 T1 D1
N20 CYCLE4079(0.02,0.01,1,100,1,1000,1)
N30 M30
```

4.26.1.54 GROUP_BEGIN - Začátek programového bloku

Syntaxe

```
GROUP_BEGIN(<_LEVEL>, <_NAME>, <_SP>, <_MODE>, <S_ICON>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam	
1		<_LEVEL>	INT	Úroveň	
				0 =	Hlavní úroveň
				1 =	1. nižší úroveň
2		<_NAME>	STRING[128]	Název bloku	
3		<_SP>	INT	Vřeteno	
				0 =	žádné vřeteno
				1 =	Hlavní vřeteno
				2 =	Protivřeteno

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam		
4		<_MODE>	INT	Režim		
				Bit 0	= 1	GROUP_ADDEND existuje
				Bit 1	= 1	ShopTurn: automatický zpětný pohyb (najíždění na bod pro výměnu nástroje)
				Bit 12	rezervováno	
				Bit 13	rezervováno	
5		<S_ICON>	STRING[32]	Název ikony (jen pro uživatelské rozhraní)		

4.26.1.55 GROUP_END - konec programového bloku

Syntaxe

```
GROUP_END(<_LEVEL>, <_SP>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam	
1		<_LEVEL>	INT	Úroveň	
				0 =	Hlavní úroveň
				1 =	1. nižší úroveň
2		<_SP>	INT	Vřeteno	
				0 =	žádné vřeteno
				1 =	Hlavní vřeteno
				2 =	Protivřeteno

4.26.1.56 GROUP_ADDEND - konec doplňkového bloku najíždění

Syntaxe

```
GROUP_ADDEND(<_LEVEL>, <_SP>)
```

Parametr

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam	
1		<_LEVEL>	INT	Úroveň	
				0 =	Hlavní úroveň
				1 =	1. nižší úroveň

Č.	Obrazovka parametrů	Interní parametry	Datový typ	Význam	
2		<_SP>	INT	Vřeteno	
				0 =	žádné vřeteno
				1 =	Hlavní vřeteno
				2 =	Protivřeteno

4.26.1.57 Okrajové podmínky

Odstupňování technologie na obrazovkách cyklů

Jestliže je odstupňování technologie aktivní, může být v různých obrazovkách cyklů zvoleno zjednodušené zadání, u kterého se vypisují pouze ty nejdůležitější parametry daného cyklu.

Zjednodušené zadávání může být zvoleno např. u následujících obrazovek cyklů:

Technologie	Obrazovka cyklu
Vrtání	Vrtání hlubokých děr
	Vrtání závitů
Frézování	Pravoúhlá kapsa
	Frézování kontury: Kapsa
Soustružení	Soustružení závitů: Podélný
	Soustružení kontur: Oddělování třísky
	Soustružení kontur: Výroba zápichů
	Soustružení kontur: Soustružení zápichů

U obrazovek cyklů, kterých se to týká, je na ploše uživatelského rozhraní k dispozici možnost volby „Zadání: **jednoduché**" a „Zadání: **kompletní**„.

Nezobrazované parametry cyklu

Parametrům cyklu, které se při zjednodušeném zadávání nezobrazují, jsou dosazeny pevně dané, technologicky smysluplné hodnoty, které ale nelze změnit. Nebo jsou těmto parametrům cyklu přiřazeny hodnoty definované v příslušných nastavovaných parametrech cyklů daného kanálu. Viz níže odstavec "Uvedení do provozu" > "Kanálové nastavované parametry pro cykly"

Přepínání „Zadávání: kompletní" > „Zadávání: jednoduché“

Jestliže je obrazovka cyklu vyplněna s nastavením „Zadávání kompletní“ a potom je přepnuto na nastavení „Zadávání jednoduché“, při generování volání cyklu jsou pro parametry, které se už nezobrazují, použity předdefinované hodnoty nebo hodnoty z nastavovaných parametrů.

Uvádění do provozu

Kanálové konfigurační strojní parametry

Prostřednictvím strojního parametru může být aktivováno odstupňování technologie na obrazovkách cyklů:

MD52210 $MCS_FUNCTION_MASK_DISP, Bit 9 = 1 (volba „Zadávání jednoduché" se zobrazuje)

Kanálové nastavované parametry pro cykly

Jestliže je na obrazovce cyklu aktivováno jednoduché zadávání, mohou být prostřednictvím následujících nastavovaných parametrů dosazeny určitým parametrům cyklů požadované hodnoty:

Číslo	Identifikátor	Význam
SD55300	$SCS_EASY_SAFETY_CLEARANCE	Bezpečnostní vzdálenost
SD55301	$SCS_EASY_DWELL_TIME	Doba prodlevy
SD55305	$SCS_EASY_DRILL_DEEP_FD1	Vrtání hlubokých děr: Procentuální hodnota: 1. Posuv
SD55306	$SCS_EASY_DRILL_DEEP_DF	Vrtání hlubokých děr: Procentuální hodnota: Přísuv
SD55307	$SCS_EASY_DRILL_DEEP_V1	Vrtání hlubokých děr: Minimální přísuvná hloubka
SD55308	$SCS_EASY_DRILL_DEEP_V2	Vrtání hlubokých děr: Vzdálenost zpětného pohybu
SD55309	$SCS_EASY_THREAD_RETURN_DIST	Soustružení závitů: Vzdálenost pro zpětný pohyb

4.26.2 Měřicí cykly

Měřicí cykly jsou speciální podprogramy vytvořené firmou Siemens, které slouží pro řešení určitých měřicích úloh. Jak je u cyklů všeobecně běžné, mohou být i měřicí cykly přizpůsobeny konkrétnímu problému prostřednictvím parametrů.

Měřicí cykly jsou k dispozici pro měření v následujících oblastech a technologiích:

- Měření nástroje při soustružení / frézování
- Měření obrobků při soustružení / frézování

Další informace

Programovací příručka, Měřicí cykly

4.26.2.1 Přehled parametrů měřicích cyklů

Parametry měřicího cyklu 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)
```

Tabulka 4-8 Parametry pro volání cyklu CYCLE973 [1)]

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
1		S_MVAR	Varianta měření (předdefinovaná hodnota=0012103)	
			Hodnoty:	JEDNOTKY: Kalibrace na ploše, hraně nebo v drážce 0 = Délka na ploše/hraně (ve WCS) se známou požadovanou hodnotou 1 = Rádius na ploše (ve WCS) se známou požadovanou hodnotou 2 = Délky v drážce (v MCS), viz S_CALNUM 3 = Rádius v drážce (v MKS), viz S_CALNUM
				DESÍTKY: rezervováno 0 = 0
				STOVKY: rezervováno 0 = 0
				TISÍCE: Volba měřicí osy a směru měření při kalibraci [2)] 0 = Žádný údaj (při kalibraci na ploše na dně drážky se nevybírá ani měřicí osa, ani směr měření) [4)] 1 = Zadání vybrané měřicí osy a směru měření, viz S_MA, S_MD (jeden směr měření v měřicí ose) 2 = Zadání vybrané měřicí osy, viz S_MA (dva směry měření v měřicí ose)
				DESÍTKY TISÍC: Zjišťování odchylky polohy (šikmá poloha měřicí sondy) [2), 3)] 0 = Zjišťování odchylky polohy 1 = Žádné zjišťování odchylky polohy
				STOVKY TISÍC: rezervováno 0 = 0
				JEDNOTKY MILIONŮ: Přizpůsobit délku nástroje [7)] 0 = Délku nástroje nepřizpůsobovat (jen spínací body) 1 = Délku nástroje přizpůsobit
2	ikona+ Počet	S_PRNUM	Počet polí parametrů měřicí sondy (nikoli číslo měřicí sondy) (předdefinovaná hodnota = 1)	
3		S_CALNUM	Číslo kalibrační drážky při kalibraci v drážce (předdefinovaná hodnota = 1) [5)]	
4		S_SETV	Požadovaná hodnota při kalibraci na ploše	
5	X0	S_MA	Měřicí osa (číslo osy) [6)] (předdefinovaná hodnota = 1)	
			Hodnoty:	1 = 1. osa v rovině (v případě G18 Z) 2 = 2. osa v rovině (v případě G18 X) 3 = 3. ose v rovině (v případě G18 Y) [6)]
6	+-	S_MD	Směr měření (předdefinovaná hodnota = 1)	
			Hodnoty:	0 = kladný směr měření 1 = záporný směr měření
7	DFA	S_FA	Měřicí dráha	
8	TSA	S_TSA	Bezpečná oblast	
9	VMS	S_VMS	Proměnná měřicí rychlost při kalibraci [2)]	
10	Měření	S_NMSP	Počet měření na tomtéž místě [2)] (předdefinovaná hodnota = 1)	
11		S_MCBIT	rezervováno	

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
12		_DMODE	Režim zobrazování	
			Hodnoty:	JEDNOTKY: Rovina obrábění G17/G18/G19 0 = Kompatibilita, zůstává aktivní rovina, která byla v platnosti před voláním cyklu 1 = G17 (aktivní pouze uvnitř cyklu) 2 = G18 (aktivní pouze uvnitř cyklu) 3 = G19 (aktivní pouze uvnitř cyklu)
13		_AMODE	Alternativní režim	

1) všechny předdefinované hodnoty = 0 nebo je u nich uvedeno, že se rovnají x

2) Vypisování závisí na všeobecném nastavovaném parametru SD54760 $SNS_MEA_FUNCTION_MASK_PIECE

3) Důležité pouze pro kalibraci ve dvou směrech osy

4) Z polohy břitu (SL) měřicí sondy.se automaticky zjišťuje pouze měřicí osa a směr měření. SL=8 --> X, SL=7 --> Z

5) Číslo kalibrační drážky (n) odkazuje na následující všeobecné nastavované parametry (všechny pozice v MCS):
v případě polohy břitu SL=7:
SD54615 $SNS_MEA_CAL_EDGE_BASE_AX1[n] Poloha dna drážky ve směru 1. osy v rovině (v případě G18 je to osa Z)
SD54621 $SNS_MEA_CAL_EDGE_PLUS_DIR_AX2[n] Poloha stěny drážky v kladném směru 2. osy v rovině (v případě G18 je to osa X)
SD54622 $SNS_MEA_CAL_EDGE_MINUS_DIR_AX2[n] Poloha stěny drážky v záporném směru 2. osy v rovině
v případě polohy břitu SL=8:
SD54619 $SNS_MEA_CAL_EDGE_BASE_AX2[n] Poloha dna drážky ve směru 2. osy v rovině
SD54620 $SNS_MEA_CAL_EDGE_UPPER_AX2[n] Poloha horní hrany drážky ve směru 2. osy v rovině, (pouze pro účely najíždění měřicí sondy do počáteční polohy)
SD54617 $SNS_MEA_CAL_EDGE_PLUS_DIR_AX1[n] Poloha stěny drážky v kladném směru 1. osy v rovině
SD54618 $SNS_MEA_CAL_EDGE_MINUS_DIR_AX1[n] Poloha stěny drážky v záporném směru 1. osy v rovině
Poznámka:
Hodnoty polohy pro stěnu drážky ve směru +/- mohou být určeny jen zhruba.
Šířka drážky z rozdílu hodnot polohy stěny drážky musí být stanovena přesně (jemným měřicím přístrojem).
Při kalibraci v drážce se vychází z toho, že délka nástroje typu „měřicí sonda“ ve směru kalibrované osy = 0.
Hodnoty polohy pro dno drážky na stroji musí být rovněž stanoveny naprosto přesně (žádné rozměry z výkresů).

6) Měřicí osa `S_MA=3` při kalibraci na ploše a na soustruhu s reálnou 3. osou v rovině (v případě G18 je to osa Y).

7) Přizpůsobení délky nástroje při kalibraci délky v drážce nebo délek na ploše.
Obrobková měřicí sonda na soustruzích může být popsána pomocí 2 délek (X Z).
Měřicí sonda pro soustružení, typ 580
Poloha břitu 7: Při kalibraci délky lze zvolit, zda se má provést korekce ve směru Z.
Měřicí sonda pro soustružení, typ 580
Poloha břitu 8: Při kalibraci délky lze zvolit, zda se má provést korekce ve směru X.
U variant měření „Rádius v drážce“ nebo „Rádius na ploše“ se neuskutečňuje žádné přizpůsobení délky nástroje.
Kalibrace se vždy týká pouze odpovídajících spínacích bodů.

Parametry měřicího cyklu 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)
```

Tabulka 4-9 Parametry pro volání cyklu CYCLE974 [1)]

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
1		S_MVAR	Varianta měření	
			Hodnoty:	JEDNOTKY: 0 = měření na čelní ploše 1 = měření zevnitř 2 = měření zvenku
				DESÍTKY: rezervováno
				STOVKY: Cíl korekce 0 = pouze měření (žádná korekce posunutí počátku NV, příp. žádná korekce nástroje) 1 = Měření, výpočet a korekce posunutí počátku (NV) (viz S_KNUM) [3)] 2 = Měření a korekce nástroje (viz S_KNUM1)
				TISÍCE: rezervováno
				DESÍTKY TISÍC: Měření s obratem nebo bez obratu hlavního vřetena (otáčející se vřeteno) 0 = měření bez obratu 1 = měření s obratem
2	Volba	S_KNUM	Korekce posunutí počátku (NV) nebo základního posunutí počátku souřadného systému (NV) nebo základního vztažného bodu [2)]	
			Hodnoty:	JEDNOTKY:
				DESÍTKY: 0 = žádná korekce 1 až max. 99: číslo posunutí počátku (nuly) nebo 1 až max. 16: číslo základního posunutí počátku
				STOVKY: rezervováno
				TISÍCE: Korekce posunutí počátku (NV) nebo základního posunutí počátku souřadného systému NV nebo základního vztažného bodu 0 = korekce nastavitelného posunutí počátku (NV) 1 = korekce kanálového základního posunutí NV 2 = korekce základního vztažného bodu 3 = korekce globálního základního posunutí počátku NV 9 = korekce aktivního posunutí počátku NV nebo v případě G500 posledního aktivního kanálového základního posunutí počátku NV
				DESÍTKY TISÍC: Korekce hrubého nebo jemného posunutí počátku (NV) nebo základního posunutí počátku souřadného systému NV nebo základního vztažného bodu 0 = jemná korekce [6)] 1 = hrubá korekce

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
3	Volba	S_KNUM1	Korekce do korekčních parametrů nástroje [2), 4)]	
			Hodnoty:	JEDNOTKY:
				DESÍTKY:
				STOVKY: 0 = žádná korekce 1 až max. 999: D-číslo (číslo břitu) v případě korekce nástroje; pokud jde o součtové a seřizovací korekce, viz také S_DLNUM
				TISÍCE: 0 nebo jednoznačné D-číslo
				DESÍTKY TISÍC: 0 nebo jednoznačné D-číslo 1 až max. 32000, pokud jsou ve strojních parametrech nastavena jednoznačná D-čísla
				STOVKY TISÍC: Korekční parametry nástroje [2)] 0 = žádný údaj (korekce geometrie nástroje) 1 = korekce délky L1 2 = korekce délky L2 3 = korekce délky L3 4 = korekce rádiusu
				JEDNOTKY MILIONŮ: Korekční parametry nástroje [2)] 0 = žádný údaj (korekce opotřebení délky nástroje) 1 = korekční parametry nástroje - součtová korekce (SK) [5)] Hodnota korekce nástroje se přičítá k již existující hodnotě SK 2 = korekční parametry nástroje - seřizovací korekce (EK) [5)] EK (nová) = EK (stará) + SK (stará) hodnota korekce, SK (nová) = 0 3 = korekční parametry nástroje - seřizovací korekce (EK) [5)] Hodnota korekce nástroje se přičítá k již existující hodnotě EK. 4 = korekční parametry nástroje - geometrie
				DESÍTKY MILIONŮ: Korekční parametry nástroje [2)] 0 = žádný údaj (korekce geometrie nástroje normálně, bez invertování) 1 = korekce invertována
				STOVKY MILIONŮ: Korekční parametry nástroje 0 = Korekční parametry nástroje bez náhradních nástroje 1 = Korekční parametry nástroje do náhradního nástroje (_DP)
4	ikona+ Počet	S_PRNUM	Počet polí parametrů měřicí sondy (nikoli číslo měřicí sondy) (předdefinovaná hodnota = 1)	
5	X0	S_SETV	Požadovaná hodnota	
6	X	S_MA	Měřicí osa (číslo osy) (předdefinovaná hodnota = 1)	
			Hodnoty:	1 = 1. osa v rovině (v případě G18 Z) 2 = 2. osa v rovině (v případě G18 X) 3 = 3. ose v rovině (v případě G18 Y) [5)]
7	DFA	S_FA	Měřicí dráha	
8	TSA	S_TSA	Bezpečná oblast	
9	α	S_STA1	Počáteční úhel při měření s obratem:	
10	Měření	S_NMSP	Počet měření na tomtéž místě [2)] (předdefinovaná hodnota = 1)	
11	T	S_TNAME	Název nástroje [2)]	
12	DL	S_DLNUM	DL-číslo seřizovací součtové korekce [5)]	

Č.	Parametr na obra-zovce	Parametr cyklu	Význam	
13	ST	_DP	Číslo náhradního nástroje (číslo duplo), jehož korekce se mají uskutečnit	
14	TZL	S_TZL	Korekce nuly [2), 4)]	
15	DIF	S_TDIF	Kontrola rozdílu rozměru [2), 4)]	
16	TUL	S_TUL	Horní mezní hodnota tolerance (inkrementálně k požadované hodnotě) [4)]	
17	TLL	S_TLL	Dolní mezní hodnota tolerance (inkrementálně k požadované hodnotě) [4)]	
18	TMV	S_TMV	Oblast korekce při výpočtu střední hodnoty [2)]	
19	FW	S_K	Váhový faktor pro výpočet střední hodnoty [2)]	
20	EVN	S_EVNUM	Číslo paměti empirické střední hodnoty [2), 7)]	
21		S_MCBIT	rezervováno	
22		_DMODE	Režim zobrazování	
			Hodnoty:	JEDNOTKY: Rovina obrábění G17/G18/G19 0 = Kompatibilita, zůstává aktivní rovina, která byla v platnosti před voláním cyklu 1 = G17 (aktivní pouze uvnitř cyklu) 2 = G18 (aktivní pouze uvnitř cyklu) 3 = G19 (aktivní pouze uvnitř cyklu)
23		_AMODE	Alternativní režim	
			Hodnoty:	JEDNOTKY: Tolerance měření ano/ne 0 = ne 1 = ano

1) všechny předdefinované hodnoty = 0 nebo je u nich uvedeno, že se rovnají x

2) Vypisování závisí na všeobecném nastavovaném parametru SD54760 $SNS_MEA_FUNCTION_MASK_PIECE

3) Korekce v posunutí počátku (NV) je možná jen u měření bez obratu.

4) V případě korekce nástroje věnujte pozornost bitu 0 a bitu 1 v kanálovém strojním parametru MD 20360 TOOL_PARAMETER_DEF_MASK

5) Jen když je aktivována funkce "Seřizovací součtová korekce" v obecném strojním parametru MD 18108 $MN_MM_NUM_SUMCORR . Kromě toho musí být nastaven bit 8 = 1 v obecném strojním parametru MD 18080 $MN_MM_TOOL_MANAGEMENT_MASK .

6) Jestliže ve strojních parametrech není aktivováno „jemné“ posunutí počátku, potom se korekce uskuteční po „hrubém“ posunutí počátku NV.

7) Výpočet empirické střední hodnoty je možný pouze v případě korekce nástroje.
Rozsah hodnot paměti empirických středních hodnot:
1 až 20: Číslo (n) paměti empirické hodnoty, viz kanálový nastavovaný parametr SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1]
10000 až 200000: Číslo (n) paměti střední hodnoty, viz kanálový nastavovaný parametr SD55625 $SCS_MEA_AVERAGE_VALUE[n-1]

Parametry měřicího cyklu 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)
```

Tabulka 4-10 Parametry pro volání cyklu CYCLE994 [1)]

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
1		S_MVAR	Varianta měření	
			Hodnoty:	JEDNOTKY: Měření zvnějšku nebo zevnitř (předdefinovaná hodnota = 1) 1 = měření zevnitř 2 = měření zvnějšku
				DESÍTKY: rezervováno
				STOVKY: Cíl korekce 0 = pouze měření (žádná korekce posunutí počátku NV, příp. žádná korekce nástroje) 1 = měření a výpočet a korekce posunutí počátku (NV) (viz S_KNUM) [3)] 2 = měření a korekce nástroje (viz S_KNUM1)
				TISÍCE: Rozsah objíždění 0 = žádný rozsah objíždění 1 = osa pro objíždění 1. ose v rovině (v případě G18 Z). Měřicí osa viz S_MA. 2 = osa pro objíždění 2. osy v rovině (v případě G18 je to osa X). Měřicí osa viz S_MA. 3 = osa pro objíždění 3. osou v rovině (v případě G18 je to osa Y). Měřicí osa viz S_MA. [8)]
2	Volba	S_KNUM	Korekce posunutí počátku (NV) nebo základního posunutí počátku souřadného systému (NV) nebo základního vztažného bodu [2)]	
			Hodnoty:	JEDNOTKY:
				DESÍTKY: 0 = žádná korekce 1 až max. 99: číslo posunutí počátku (nuly) nebo 1 až max. 16: číslo základního posunutí počátku
				STOVKY: rezervováno
				TISÍCE: Korekce posunutí počátku (NV) nebo základního posunutí počátku nebo základního vztažného bodu 0 = korekce nastavitelného posunutí počátku (NV) 1 = korekce kanálového základního posunutí NV 2 = korekce základního vztažného bodu 3 = korekce globálního základního posunutí počátku NV 9 = korekce aktivního posunutí počátku NV nebo v případě G500 posledního aktivního kanálového základního posunutí počátku NV
				DESÍTKY TISÍC: Korekce hrubého nebo jemného posunutí počátku (NV) nebo základního posunutí počátku souřadného systému NV nebo základního vztažného bodu 0 = jemná korekce [6)] 1 = hrubá korekce

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
3	Volba	S_KNUM1	Korekce do korekčních parametrů nástroje [2), 4)]	
			Hodnoty:	JEDNOTKY:
				DESÍTKY:
				STOVKY: 0 = žádná korekce 1 až max. 999: D-číslo (číslo břitu) korekčních parametrů nástroje Pokud jde o součtové a seřizovací korekce, viz také S_DLNUM
				TISÍCE: 0 nebo jednoznačné D-číslo
				DESÍTKY TISÍC: 0 nebo jednoznačné D-číslo 1 až max. 32000, pokud jsou ve strojních parametrech nastavena jednoznačná D-čísla
				STOVKY TISÍC: Korekční parametry nástroje [2)] 0 = žádný údaj (korekce geometrie nástroje) 1 = korekce délky L1 2 = korekce délky L2 3 = korekce délky L3 4 = korekce rádiusu
				JEDNOTKY MILIONŮ: Korekční parametry nástroje [2)] 0 = žádný údaj (korekce opotřebení délky nástroje) 1 = korekční parametry nástroje - součtová korekce (SK) [5)] Hodnota korekce nástroje se přičítá k již existující hodnotě SK 2 = korekční parametry nástroje - seřizovací korekce (EK) [5)] EK (nová) = EK (stará) + SK (stará) hodnota korekce, SK (nová) = 0 3 = korekční parametry nástroje - seřizovací korekce (EK) [5)] Hodnota korekce nástroje se přičítá k již existující hodnotě EK. 4 = korekční parametry nástroje - geometrie
				DESÍTKY MILIONŮ: Korekční parametry nástroje [2)] 0 = žádný údaj (korekce geometrie nástroje normálně, bez invertování) 1 = korekce invertována
				STOVKY MILIONŮ: Korekční parametry nástroje 0 = Korekční parametry nástroje bez náhradních nástroje 1 = Korekční parametry nástroje do náhradního nástroje (_DP)
4	ikona+ Počet	S_PRNUM	Počet polí parametrů měřicí sondy (nikoli číslo měřicí sondy) (předdefinovaná hodnota = 1)	
5	X0	S_SETV	Požadovaná hodnota	
6	X	S_MA	Číslo měřicí osy (předdefinovaná hodnota = 1)	
			Hodnoty:	1 = 1. osa v rovině (v případě G18 Z) 2 = 2. osa v rovině (v případě G18 X) 3 = 3. osa v rovině (v případě G18 Y) [8)]
7	X1	S_SZA	Šířka objíždění v měřicí ose	
8	Y1	S_SZO	Šířka objíždění v ose pro objíždění	
9	DFA	S_FA	Měřicí dráha	
10	TSA	S_TSA	Bezpečná oblast	
11	Měření	S_NMSP	Počet měření na tomtéž místě [2)] (předdefinovaná hodnota = 1)	
12	T	S_TNAME	Název nástroje [2)]	

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
13	DL	S_DLNUM	DL-číslo seřizovací součtové korekce [5]	
14	ST	_DP	Číslo náhradního nástroje (číslo duplo), jehož korekce se mají uskutečnit	
15	TZL	S_TZL	Korekce nuly [2), 4)]	
16	DIF	S_TDIF	Kontrola rozdílu rozměru [2), 4)]	
17	TUL	S_TUL	Horní mezní hodnota tolerance (inkrementálně k požadované hodnotě) [4)]	
18	TLL	S_TLL	Dolní mezní hodnota tolerance (inkrementálně k požadované hodnotě) [4)]	
19	TMV	S_TMV	Oblast korekce při výpočtu střední hodnoty [2)]	
20	FW	S_K	Váhový faktor pro výpočet střední hodnoty [2)]	
21	EVN	S_EVNUM	Číslo paměti empirické hodnoty[2), 7)]	
22		S_MCBIT	rezervováno	
23		_DMODE	Režim zobrazování	
			Hodnoty:	JEDNOTKY: Rovina obrábění G17/G18/G19 0 = Kompatibilita, zůstává aktivní rovina, která byla v platnosti před voláním cyklu 1 = G17 (aktivní pouze uvnitř cyklu) 2 = G18 (aktivní pouze uvnitř cyklu) 3 = G19 (aktivní pouze uvnitř cyklu)
24		_AMODE	Alternativní režim	
			Hodnoty:	JEDNOTKY: Tolerance měření ano/ne 0 = ne 1 = ano

1) všechny předdefinované hodnoty = 0 nebo je u nich uvedeno, že se rovnají x

2) Vypisování závisí na všeobecném nastavovaném parametru SD54760 $SNS_MEA_FUNCTION_MASK_PIECE

3) Korekce v posunutí počátku (NV) je možná jen u měření bez obratu.

4) V případě korekcí nástroje věnujte pozornost kanálovému parametrů-MD 20360 TOOL_PARAMETER_DEF_MASK .

5) Jen když je aktivována funkce "Seřizovací součtová korekce" v obecném strojním parametru MD 18108 $MN_MM_NUM_SUMCORR . Kromě toho musí být nastaven bit 8 = 1 v obecném strojním parametru MD 18080 $MN_MM_TOOL_MANAGEMENT_MASK .

6) Jestliže ve strojních parametrech není aktivováno „jemné“ posunutí počátku, potom se korekce uskuteční po „hrubém“ posunutí počátku NV.

7) Výpočet empirické střední hodnoty je možný pouze v případě korekce nástroje.
Rozsah hodnot paměti empirických středních hodnot:
1 až 20: Číslo (n) paměti empirické hodnoty, viz kanálový nastavovaný parametr SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1]
10000 až 200000: Číslo (n) paměti střední hodnoty, viz kanálový nastavovaný parametr SD55625 $SCS_MEA_AVERAGE_VALUE[n-1]

8) Je-li na stroji k dispozici osa Y.

Parametry měřicího cyklu 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)
```

Tabulka 4-11 Parametry pro volání cyklu CYCLE976 [1)]

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
1		S_MVAR	Varianta měření (předdefinovaná hodnota=1000)	
			Hodnoty:	JEDNOTKY: Kalibrace na ploše, pomocí kalibrační kuličky nebo v kalibračním kruhu [2)] 0 = délka na ploše se známou požadovanou hodnotou 1 = rádius v kalibračním kruhu se známým průměrem (požadovaná hodnota) a se známým středem. 2 = rádius v kalibračním kruhu se známým průměrem (požadovaná hodnota) a s neznámým středem 3 = rádius a délka na kalibrační kuličce 4 = rádius na hraně se známou žádanou hodnotou. Věnujte pozornost volbě měřicí osy a směru měření. [3)] 5 = Rádius mezi dvěma hranami se známou požadovanou hodnotou a vzdáleností mezi hranami. Je třeba vybrat měřicí osu.
				DESÍTKY: rezervováno 0 = 0
				STOVKY: rezervováno 0 = 0
				TISÍCE: Volba měřicí osy a směru měření při kalibraci. 0 = Žádný údaj (není potřeba vybrat ani měřicí osu, ani směr měření) [8)] 1 = Zadání vybrané měřicí osy a směru měření, viz S_MA, S_MD (jeden směr měření v měřicí ose) 2 = Zadání vybrané měřicí osy, viz S_MA (dva směry měření v měřicí ose)
				DESÍTKY TISÍC: Zjišťování odchylky polohy (šikmá poloha měřicí sondy) [2)] 0 = Zjišťování odchylky polohy měřicí sondy[6)] 1 = žádné zjišťování odchylky polohy
				STOVKY TISÍC: Kalibrace rovnoběžně s osami nebo pod určitým úhlem 0 = kalibrace rovnoběžně s osami v aktivním WCS 1 = kalibrace pod určitým úhlem [7)]
				JEDNOTKY MILIONŮ: Stanovení délky nástroje při kalibraci na ploše nebo pomocí kuličky 0 = Žádné zjišťování délky nástroje 1 = Zjišťování délky nástroje [4)] 2 = Kalibrace přísuvné osy podle kuličky, zjišťování délky nástroje, zápis změřeného rozdílu délek nástrojů do kalibračních dat
2	ikona+ Počet	S_PRNUM	Počet polí parametrů měřicí sondy (nikoli číslo měřicí sondy) (předdefinovaná hodnota = 1)	
3		S_SETV	Požadovaná hodnota	
4	Z0	S_SETV0	Požadovaná hodnota vztažného bodu délky při kalibraci pomocí kuličky	
5	X / Y / Z	S_MA	Měřicí osa (číslo osy) [2), 6)] (předdefinovaná hodnota = 1)	
			Hodnoty:	1 = 1. osa v rovině (v případě G17 X) 2 = 2. osa v rovině (v případě G17 Y) 3 = 3. osa v rovině (v případě G17 Z).

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
6	+ -	S_MD	Směr měření [2), 6)]	
			Hodnoty:	0 = kladný 1 = záporný
7	DFA	S_FA	Měřicí dráha	
8	TSA	S_TSA	Bezpečná oblast	
9	VMS	S_VMS	Proměnná měřicí rychlost při kalibraci [2)]	
10	α	S_STA1	Počáteční úhel [2), 5)]	
11	Měření	S_NMSP	Počet měření na tomtéž místě [2)] (předdefinovaná hodnota = 1)	
12	X0	S_SETV1	Vztažný bod hrany při kalibraci mezi 2 hranami [3)]	
13		_DMODE	Režim zobrazování	
			Hodnoty:	JEDNOTKY: Rovina obrábění G17/G18/G19 0 = Kompatibilita, zůstává aktivní rovina, která byla v platnosti před voláním cyklu 1 = G17 (aktivní pouze uvnitř cyklu) 2 = G18 (aktivní pouze uvnitř cyklu) 3 = G19 (aktivní pouze uvnitř cyklu)
14		_AMODE	Alternativní režim	

1) všechny předdefinované hodnoty = 0 nebo je u nich uvedeno, že se rovnají x

2) Vypisování závisí na všeobecném nastavovaném parametru SD54760 $SNS_MEA_FUNCTION_MASK_PIECE

3) Při kalibraci "Rádius v kalibračním kruhu" musí být hodnoty průměru a středu kruhu známy (4 měření).
Při kalibraci typu "Rádius na dvou hranách" musí být vzdálenost hran ve směru měřicí osy známa (2 měření).
Při kalibraci typu "Rádius na hraně" musí být známa požadovaná hodnota plochy.

4) Varianta měření typu "Jen kalibrace na ploše" (délka na ploše) vyplývá korigovaná délka nástroje z parametrů S_MD a S_MA.

5) Jen při variantě měření "Kalibrační kruh, ... a známý střed" (S_MVAR=1xxx02).

6) Měřicí osa jen při variantě měření S_MVAR=0 nebo =xx1x01 nebo =xx2x01 nebo =20000
Varianta měření: „Kalibrace na ploše“ --> volba měřicí osy a směru měření
nebo v případě typu „Kalibrační kruh, ... a známý střed“ --> volba směru osy a volba měřicí osy a směru měření
nebo v případě typu „Kalibrační kruh, ... a známý střed“ --> volba dvou směrů os a volba měřicí osy
nebo „Zjišťování délky měřicí osy“ --> S_MA=3 --> 3. osa v rovině (v případě G17 Z).

7) Pouze varianta měření kalibrace v kalibračním kruhu nebo pomocí kalibrační kuličky
Při "Kalibraci na kalibrační kuličce" se při měření pod určitým úhlem kulička objíždí podél jejího "rovníku".

8) Při kalibraci typu "Rádius v kalibračním kruhu" s neznámým středem čtyři měření v rovině (v případě G17 +-X +-Y).
Při kalibraci "Délka na ploše" v záporném směru osy nástroje (v případě G17 je to směr -Z).

Parametry měřicího cyklu 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)
```

Tabulka 4-12 Parametry pro volání cyklu CYCLE978 [1)]

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
1		S_MVAR	Varianta měření	
			Hodnoty:	JEDNOTKY: Konturový prvek 0 = měření plochy
				DESÍTKY: rezervováno
				STOVKY: Cíl korekce 0 = pouze měření (žádná korekce posunutí počátku NV, příp. žádná korekce nástroje) 1 = měření, výpočet a korekce posunutí počátku (NV) (viz S_KNUM) 2 = měření a korekce nástroje (viz S_KNUM1)
				TISÍCE: rezervováno
				DESÍTKY TISÍC: Měření s obratem/bez obratu vřetena nebo nastavení orientace měřicí sondy ve směru spínání [9)] 0 = měření bez obratu vřetena, bez nastavování orientace měřicí sondy 1 = měření s obratem vřetena 2 = měření s nastavováním orientace měřicí sondy
2	Volba	S_KNUM	Korekce posunutí počátku (NV) nebo základního posunutí počátku souřadného systému (NV) nebo základního vztažného bodu [2)]	
			Hodnoty:	JEDNOTKY:
				DESÍTKY: 0 = žádná korekce 1 až max. 99: číslo posunutí počátku (nuly) nebo 1 až max. 16: číslo základního posunutí počátku
				STOVKY: rezervováno
				TISÍCE: Korekce posunutí počátku (NV) nebo základního posunutí počátku nebo základního vztažného bodu 0 = korekce nastavitelného posunutí počátku (NV) 1 = korekce kanálového základního posunutí NV 2 = korekce základního vztažného bodu 3 = korekce globálního základního posunutí počátku NV 9 = korekce aktivního posunutí počátku NV nebo v případě G500 posledního aktivního kanálového základního posunutí počátku NV
				DESÍTKY TISÍC: Korekce hrubého nebo jemného posunutí počátku (NV) nebo základního posunutí počátku souřadného systému NV nebo základního vztažného bodu 0 = jemná korekce [6)] 1 = hrubá korekce

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
3	Volba	S_KNUM1	Korekce do korekčních parametrů nástroje [2]	
			Hodnoty:	JEDNOTKY:
				DESÍTKY:
				STOVKY: 0 = žádná korekce 1 až max. 999: D-číslo (číslo břitu) v případě korekce nástroje, Pokud jde o součtové a seřizovací korekce, viz také S_DLNUM
				TISÍCE: 0 nebo jednoznačné D-číslo
				DESÍTKY TISÍC: 0 nebo jednoznačné D-číslo 1 až max. 32000, pokud jsou ve strojních parametrech nastavena jednoznačná D-čísla
				STOVKY TISÍC: Korekční parametry nástroje [2] 0 = žádný údaj (korekce geometrie nástroje) 1 = korekce délky L1 2 = korekce délky L2 3 = korekce délky L3 4 = korekce rádiusu
				JEDNOTKY MILIONŮ: Korekční parametry nástroje [2] 0 = žádný údaj (korekce opotřebení rádiusu nástroje) 1 = korekční parametry nástroje - součtová korekce (SK) [5] Hodnota korekce nástroje se přičítá k již existující hodnotě SK 2 = korekční parametry nástroje - seřizovací korekce (EK) [5] EK (nová) = EK (stará) + SK (stará) hodnota korekce, SK (nová) = 0 3 = korekční parametry nástroje - seřizovací korekce (EK) [5] Hodnota korekce nástroje se přičítá k již existující hodnotě EK. 4 = korekční parametry nástroje - geometrie
				DESÍTKY MILIONŮ: Korekční parametry nástroje [2] 0 = žádný údaj (korekce geometrie nástroje normálně, bez invertování) 1 = korekce invertována
				STOVKY MILIONŮ: Korekční parametry nástroje 0 = Korekční parametry nástroje bez náhradních nástroje 1 = Korekční parametry nástroje do náhradního nástroje (_DP)
4	ikona+Počet	S_PRNUM	Číslo pole parametru měřicí sondy (nikoli číslo měřicí sondy) (rozsah hodnot = 1 až 40)	
5	X0	S_SETV	Požadovaná hodnota	
6	DFA	S_FA	Měřicí dráha	
7	TSA	S_TSA	Bezpečná oblast	
8	X	S_MA	Číslo měřicí osy [7] (rozsah hodnot 1 až 3)	
			Hodnoty:	1 = 1. osa v rovině (v případě G17 X) 2 = 2. osa v rovině (v případě G17 Y) 3 = 3. osa v rovině (v případě G17 Z) Měření ve směru nástroje
9		S_MD	Směr měření měřicí osy	
			Hodnoty:	1 = kladný směr měření 2 = záporný směr měření
10	Měření	S_NMSP	Počet měření na tomtéž místě [2] (rozsah hodnot 1 až 9)	
11	TR	S_TNAME	Název nástroje [3]	

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
12	DL	S_DLNUM	DL-číslo seřizovací součtové korekce [5]	
13	ST	_DP	Číslo náhradního nástroje (číslo duplo), jehož korekce se mají uskutečnit	
14	TZL	S_TZL	Korekce nuly [2], [3]	
15	DIF	S_TDIF	Kontrola rozdílu rozměru [2], [3]	
16	TUL	S_TUL	Horní mezní hodnota tolerance (inkrementálně k požadované hodnotě) [3]	
17	TLL	S_TLL	Dolní mezní hodnota tolerance (inkrementálně k požadované hodnotě) [3]	
18	TMV	S_TMV	Oblast korekce při výpočtu střední hodnoty [2]	
19	FW	S_K	Váhový faktor pro výpočet střední hodnoty [2]	
20	EVN	S_EVNUM	Datový blok paměti empirické hodnoty[2], [8]	
21		S_MCBIT	rezervováno	
22		_DMODE	Režim zobrazování	
			Hodnoty:	JEDNOTKY: Rovina obrábění G17/G18/G19 0 = Kompatibilita, zůstává aktivní rovina, která byla v platnosti před voláním cyklu 1 = G17 (aktivní pouze uvnitř cyklu) 2 = G18 (aktivní pouze uvnitř cyklu) 3 = G19 (aktivní pouze uvnitř cyklu)
23		_AMODE	Alternativní režim	
			Hodnoty:	JEDNOTKY: Tolerance měření ano/ne 0 = ne 1 = ano

[1] všechny předdefinované hodnoty = 0 nebo je u nich uvedeno, že mají rozsah hodnot a až b

[2] Vypisování závisí na všeobecném nastavovaném parametru SD54760 $SNS_MEA_FUNCTION_MASK_PIECE

[3] jen v případě korekce nástroje, jinak má parametr hodnotu ""

[4] jen v případě korekce nástroje a je-li pro toleranci rozměru zvoleno "Ano", jinak má parametr hodnotu = 0

[5] Jen když je aktivována funkce "Seřizovací součtová korekce" v obecném strojním parametru MD 18108 $MN_MM_NUM_SUMCORR . Kromě toho musí být nastaven bit 8 = 1 v obecném strojním parametru MD 18080 $MN_MM_TOOL_MANAGEMENT_MASK .

[6] Jestliže ve strojních parametrech není aktivováno „jemné“ posunutí počátku, potom se korekce uskuteční po „hrubém“ posunutí počátku NV.

[7] Korekce geometrie nástroje:
- při měření v rovině (S_MA=1 nebo S_MA=2) Korekce rádiusu nástroje
- při měření ve směru nástroje (S_MA=3) Korekce délky nástroje L1

[8] Výpočet empirické střední hodnoty je možný pouze v případě korekce nástroje a korekce posunutí počátku NV.
Rozsah hodnot paměti empirických středních hodnot:
1 až 20: Číslo (n) paměti empirické hodnoty, viz kanálový nastavovaný parametr SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1]
10000 až 200000: Číslo (n) paměti střední hodnoty, viz kanálový nastavovaný parametr SD55625 $SCS_MEA_AVERAGE_VALUE[n-1]

[9] Při měření s obratem vřetena musí být rádius/průměr měřicí sondy přesně stanoven. To by mělo být uskutečněno pomocí varianty kalibrace CYCLE976 Rádius na kruhu nebo na hraně nebo na kuličce. Jinak je výsledek měření zkreslený.

Parametry měřicího cyklu 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)
```

Tabulka 4-13 Parametry pro volání cyklu CYCLE998 [1)]

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
1		S_MVAR	Varianta měření (předdefinovaná hodnota=5)	
			Hodnoty:	JEDNOTKY: Konturový prvek 5 = měření hrany (úhel) 6 = měření roviny (dva úhly)
				DESÍTKY: rezervováno
				STOVKY: Cíl korekce 0 = pouze měření a žádná korekce posunutí NV 1 = měření a výpočet a korekce posunutí počátku NV (viz S_KNUM)
				TISÍCE: Chráněná oblast 0 = žádné zohledňování chráněné oblasti 1 = zohledňování chráněné oblasti
				DESÍTKY TISÍC: Měření s obratem vřetena (diferenční měření) 0 = měření bez obratu vřetena 1 = měření s obratem vřetena
				STOVKY TISÍC: Měření pod určitým úhlem nebo rovnoběžně s osou 0 = měření pod určitým úhlem 1 = měření rovnoběžně s osou
2	Volba	S_KNUM	Korekce posunutí počátku (NV) nebo základního posunutí počátku souřadného systému (NV) nebo základního vztažného bodu [2)]	
			Hodnoty:	JEDNOTKY:
				DESÍTKY: 0 = žádná korekce 1 až max. 99: číslo posunutí počátku (nuly) nebo 1 až max. 16: číslo základního posunutí počátku
				STOVKY: rezervováno
				TISÍCE: Korekce posunutí počátku (NV) nebo základního posunutí počátku nebo základního vztažného bodu 0 = korekce nastavitelného posunutí počátku (NV) 1 = korekce kanálového základního posunutí NV 2 = korekce základního vztažného bodu 9 = korekce aktivního posunutí počátku NV nebo v případě G500 do posledního aktivního kanálového základního posunutí počátku NV
				DESÍTKY TISÍC: Jemná nebo hrubá korekce posunutí počátku (NV) nebo základního posunutí počátku NV nebo základního vztažného bodu [3)] 0 = jemná korekce 1 = hrubá korekce

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
3	A, B, C	S_RA	Cílové korekční pootočení souřadnic nebo kruhové osy [14]	
			Hodnoty:	0 = Cílem korekce je pootočení souřadnic okolo osy, která je dána parametrem S_MA [4] >0 = Cílem korekce je kruhová osa. Hodnota udává číslo kanálové osy, která je kruhovou osou (nejlépe otočný stůl). Úhlová korekce se uskutečňuje v translační složce posunutí počátku NV kruhové osy.
4	ikona+ Počet	S_PRNUM	Počet polí parametrů měřicí sondy (předdefinovaná hodnota = 1)	
5	DX / DY / DZ	S_SETV	Dráha (inkrementálně) od počáteční pozice po měřicí bod P1 měřicí osy (S_MA) [5]	
6	α	S_STA1	Požadovaná hodnota úhlu v případě funkce „Srovnat podle hrany“, příp. „Srovnat podle roviny“ okolo 1. osy v rovině (v případě G17 X) [9]	
7	β	S_INCA	Požadovaná hodnota úhlu v případě funkce „Srovnat podle roviny“ okolo 2. osy v rovině (v případě G17 Y) [9]	
8	DFA	S_FA	Měřicí dráha	
9	TSA	S_TSA	Bezpečná oblast Kontrola úhlového rozdílu vzhledem k požadované hodnotě úhlu [stupně] [6]	
10	X / Y / Z	S_MA	Měřicí osa, osa posuvu [7] (předdefinovaná hodnota=201)	
			Hodnoty:	JEDNOTKY: Číslo měřicí osy 1 = 1. osa v rovině (v případě G17 X) 2 = 2. osa v rovině (v případě G17 Y) 3 = 3. osa v rovině (v případě G17 Z).
				DESÍTKY: rezervováno
				STOVKY: Číslo osy posuvu 1 = 1. osa v rovině (v případě G17 X) 2 = 2. osa v rovině (v případě G17 Y) 3 = 3. osa v rovině (v případě G17 Z).
11	+-	S_MD	Směr měření měřicí osy [8]	
			Hodnoty:	0 = Směr měření se zjišťuje z požadované hodnoty a ze skutečné polohy měřicí osy (kompatibilita) 1 = kladný směr měření 2 = záporný směr měření
12	L2	S_ID	V případě varianty měření "Srovnat podle hrany": Vzdálenost (inkrementálně) mezi měřicími body P1 a P2 na ose posuvu (hodnota > 0) V případě varianty měření "Srovnat podle roviny“ platí parametry uvedené dále.	
13	L2	S_SETV0	Vzdálenost mezi měřicími body P1 a P2 v 1. ose v rovině [10]	
14		S_SETV1	Vzdálenost mezi měřicími body P1 a P2 ve 2. ose v rovině [11], [12]	
15	L3x	S_SETV2	Vzdálenost mezi měřicími body P1 a P3 v 1. ose v rovině [11]	
16	L3y	S_SETV3	Vzdálenost mezi měřicími body P1 a P3 v 2. ose v rovině [10]	
17	Měření	S_NMSP	Počet měření na tomtéž místě [2] (předdefinovaná hodnota = 1)	
18		S_EVNUM	Datový blok paměti empirické hodnoty [2], [13]	

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
19		_DMODE	Režim zobrazování	
			Hodnoty:	JEDNOTKY: Rovina obrábění G17/G18/G19 0 = Kompatibilita, zůstává aktivní rovina, která byla v platnosti před voláním cyklu 1 = G17 (aktivní pouze uvnitř cyklu) 2 = G18 (aktivní pouze uvnitř cyklu) 3 = G19 (aktivní pouze uvnitř cyklu)
20		_AMODE	Rezervováno (alternativní režim)	

1) všechny předdefinované hodnoty = 0 nebo je u nich uvedeno, že se rovnají x

2) Vypisování závisí na všeobecném nastavovaném parametru SD54760 $SNS_MEA_FUNCTION_MASK_PIECE

3) „Jemné“ posunutí počátku jen tehdy, je-li cílem korekce kruhová osa a MD 52207 $MCS_AXIS_USAGE_ATTRIB[n] bit 6 = 1. Jestliže ve strojních parametrech není aktivováno posunutí počátku NV, potom se korekce uskuteční v „hrubém“ posunutí počátku NV.

4) Příklad pro korekci v pootočení souřadnic: S_MA=102 Měřicí osa Y, osa posuvu X, z toho vyplývá otáčení souřadnic okolo osy Z (v případě G17)

5) Hodnota má smysl jen tehdy, je-li pro chráněnou oblast nastaveno "ano" (S_MVAR místo TISÍCŮ = 1)

6) V případě nastavování polohy měřicího bodu P1 k měřicímu bodu P2 na ose posuvu se úhel v parametrech S_STA1 a S_TSA sečítá.

7) Číslo měřicí osy se nesmí rovnat číslu osy posuvu (např. 101 je nepřípustné)

8) Směr měření jen u variant "Srovnat podle hrany" a "Měření rovnoběžně s osami" (S_MVAR=10x105)

9) Rozsah úhlů S_STA1 ±45 stupňů v případě varianty "Srovnat podle hrany"
Rozsah úhlů S_STA1 0 až +60 stupňů a S_INCA ±30 stupňů v případě varianty "Srovnat podle roviny"

10) V případě varianty měření "Srovnat podle roviny" a "Srovnat podle hrany"

11) V případě varianty měření "Měření roviny" a "Měření rovnoběžně s osami"

12) nikoli pro verzi měřicích cyklů SW 04.04.

13) Výpočet empirické střední hodnoty v případě korekce posunutí počátku NV, rozsah hodnot pro paměť empirických středních hodnot:
1 až 20, číslo (n) paměti empirických středních hodnot viz kanálový nastavovaný parametr SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1]

14) Parametr S_RA je relevantní pouze pro variantu měření „Srovnání podle hrany“ (S_MVAR xxxxx5).

Parametry měřicího cyklu 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)
```

Tabulka 4-14 Parametry pro volání cyklu CYCLE977 [1)]

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
1		S_MVAR	Varianta měření	
			Hodnoty:	JEDNOTKY: Konturový prvek (rozsah hodnot 1 až 6) 1 = měření díry 2 = měření čepu (hřídele 3 = měření drážky 4 = měření žebra 5 = měření vnitřního pravého rohu 6 = měření vnějšího pravého rohu
				DESÍTKY: rezervováno
				STOVKY: Cíl korekce 0 = pouze měření (žádná korekce posunutí počátku NV, příp. žádná korekce nástroje) 1 = měření a výpočet a korekce posunutí počátku (NV) (viz S_KNUM) 2 = měření a korekce nástroje (viz S_KNUM1)
				TISÍCE: Chráněná oblast 0 = žádné zohledňování chráněné oblasti 1 = zohledňování chráněné oblasti
				DESÍTKY TISÍC: Měření s obratem/bez obratu vřetena (diferenční měření) nebo nastavení orientace měřicí sondy ve směru spínání 0 = měření bez obratu vřetena, orientace měřicí sondy není nastavována 1 = měření s obratem vřetena 2 = měření s nastavováním orientace měřicí sondy
2	Volba	S_KNUM	Korekce posunutí počátku (NV) nebo základního posunutí počátku souřadného systému (NV) nebo základního vztažného bodu [2)]	
			Hodnoty:	JEDNOTKY:
				DESÍTKY: 0 = žádná korekce 1 až max. 99: číslo posunutí počátku (nuly) nebo 1 až max. 16: číslo základního posunutí počátku
				STOVKY: rezervováno
				TISÍCE: Korekce posunutí počátku (NV) nebo základního posunutí počátku nebo základního vztažného bodu 0 = korekce nastavitelného posunutí počátku (NV) 1 = korekce kanálového základního posunutí NV 2 = korekce základního vztažného bodu 3 = korekce globálního základního posunutí počátku NV 9 = korekce aktivního posunutí počátku NV nebo v případě G500 posledního aktivního kanálového základního posunutí počátku NV
				DESÍTKY TISÍC: Korekce hrubého nebo jemného posunutí počátku (NV) nebo základního posunutí počátku souřadného systému NV nebo základního vztažného bodu 0 = jemná korekce [6)] 1 = hrubá korekce

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
3	Volba	S_KNUM1	Korekce do korekčních parametrů nástroje [2]	
			Hodnoty:	JEDNOTKY:
				DESÍTKY:
				STOVKY: 0 = žádná korekce 1 až max. 999: D-číslo (číslo břitu) v případě korekce nástroje; pokud jde o součtové a seřizovací korekce, viz také S_DLNUM
				TISÍCE: 0 nebo jednoznačné D-číslo
				DESÍTKY TISÍC: 0 nebo jednoznačné D-číslo 1 až max. 32000, pokud jsou ve strojních parametrech nastavena jednoznačná D-čísla
				STOVKY TISÍC: Korekční parametry nástroje [2] 0 = žádný údaj (korekce rádiusu nástroje) 1 = korekce délky L1 2 = korekce délky L2 3 = korekce délky L3 4 = korekce rádiusu
				JEDNOTKY MILIONŮ: Korekční parametry nástroje [2] 0 = žádný údaj (korekce opotřebení rádiusu nástroje) 1 = korekční parametry nástroje - součtová korekce (SK) [5] Hodnota korekce nástroje se přičítá k již existující hodnotě SK 2 = korekční parametry nástroje - seřizovací korekce (EK) [5] EK (nová) = EK (stará) + SK (stará) hodnota korekce, SK (nová) = 0 3 = korekční parametry nástroje - seřizovací korekce (EK) [5] Hodnota korekce nástroje se přičítá k již existující hodnotě EK. 4 = korekční parametry nástroje - geometrie
				DESÍTKY MILIONŮ: Korekční parametry nástroje [2] 0 = žádný údaj (korekce geometrie nástroje normálně, bez invertování) 1 = korekce invertována
				STOVKY MILIONŮ: Korekční parametry nástroje 0 = Korekční parametry nástroje bez náhradních nástroje 1 = Korekční parametry nástroje do náhradního nástroje (_DP)
4	ikona+ Počet	S_PRNUM	Číslo pole parametru měřicí sondy (nikoli číslo měřicí sondy) (rozsah hodnot = 1 až 40)	
5	X0	S_SETV	Požadovaná hodnota	
6	X0	S_SETV0	Požadovaná hodnota v případě pravého úhlu v 1. ose v rovině (v případě G17 je to osa X)	
7	Y0	S_SETV1	Požadovaná hodnota v případě pravého úhlu v 2. ose v rovině (v případě G17 je to osa Y)	
8	XM	S_XM	Zadání požadované hodnoty středu, geometrická osa X	
9	YM	S_YM	Zadání požadované hodnoty středu, geometrická osa Y	
10	DFA	S_FA	Měřicí dráha	
11	TSA	S_TSA	Bezpečná oblast	
12	α 0	S_STA1	Počáteční úhel	

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
13	DZ	S_ID	Inkrementální hodnota 1. Inkrementální přísuv 3. osy v rovině (v případě G17 je to osa Z) Směr přísuvu určuje znaménko parametru S_ID Při měření čepu, žebra a vnějšího pravého úhlu je parametrem S_ID definováno spuštění na měřicí výšku. 2. Zohledňování chráněné oblasti Při měření díry, drážky a vnitřního pravého úhlu a s chráněnou oblastí je pomocí parametru S_ID definována výška pro přejíždění.	
14	X1	S_SZA	Průměr nebo délka (šířka) chráněné oblasti [7]	
15	Y1	S_SZO	V případě měření pravého úhlu: Šířka chráněné oblasti ve 2. ose v rovině	
16	X	S_MA	Číslo měřicí osy [7] (jen v případě měření drážky nebo žebra)	
			Hodnoty:	1 = 1. osa v rovině (v případě G17 X) 2 = 2. osy v rovině (v případě G17 je to osa Y)
17	ST	_DP	Číslo náhradního nástroje (číslo duplo), jehož korekce se mají uskutečnit	
18	Měření	S_NMSP	Počet měření na tomtéž místě [2] (rozsah hodnot 1 až 9)	
19	TR	S_TNAME	Název nástroje [2]	
20	DL	S_DLNUM	DL-číslo seřizovací součtové korekce [5]	
21	TZL	S_TZL	Korekce nuly [2, 4]	
22	DIF	S_TDIF	Kontrola rozdílu rozměru [2, 4]	
23	TUL	S_TUL	Horní mezní hodnota tolerance (inkrementálně k požadované hodnotě) [4]	
24	TLL	S_TLL	Dolní mezní hodnota tolerance (inkrementálně k požadované hodnotě) [4]	
25	TMV	S_TMV	Oblast korekce při výpočtu střední hodnoty [2]	
26	FW	S_K	Váhový faktor pro výpočet střední hodnoty [2]	
27		S_EVNUM	Datový blok paměti empirické střední hodnoty [2, 8]	
28		S_MCBIT	rezervováno	
29		_DMODE	Režim zobrazování	
			Hodnoty:	JEDNOTKY: Rovina obrábění G17/G18/G19 0 = Kompatibilita, zůstává aktivní rovina, která byla v platnosti před voláním cyklu 1 = G17 (aktivní pouze uvnitř cyklu) 2 = G18 (aktivní pouze uvnitř cyklu) 3 = G19 (aktivní pouze uvnitř cyklu)
30		_AMODE	Alternativní režim	
			Hodnoty:	JEDNOTKY: Tolerance měření ano/ne 0 = ne 1 = ano

1) všechny předdefinované hodnoty = 0 nebo je u nich uvedeno, že mají rozsah hodnot a až b

2) Vypisování závisí na všeobecném nastavovaném parametru SD54760 $SNS_MEA_FUNCTION_MASK_PIECE

3) jen v případě korekce nástroje, jinak má parametr hodnotu ""

4) jen v případě korekce nástroje a je-li pro toleranci rozměru zvoleno "Ano", jinak má parametr hodnotu = 0

5) Jen když je aktivována funkce "Seřizovací součtová korekce" v obecném strojním parametru MD 18108 $MN_MM_NUM_SUMCORR . Kromě toho musí být nastaven bit 8 = 1 v obecném strojním parametru MD 18080 $MN_MM_TOOL_MANAGEMENT_MASK .

6) Jestliže ve strojních parametrech není aktivováno „jemné“ posunutí počátku, potom se korekce uskuteční po „hrubém“ posunutí počátku NV.

[7] Průměr, příp. šířka chráněné oblasti uvnitř díry nebo drážky.
Průměr, příp. šířka chráněné oblasti vně čepu nebo žebra

[8] Výpočet empirické střední hodnoty je možný v případě korekce nástroje.
Rozsah hodnot paměti empirických středních hodnot:
1 až 20: Číslo (n) paměti empirické hodnoty, viz kanálový nastavovaný parametr SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1]
10000 až 200000: Číslo (n) paměti střední hodnoty, viz kanálový nastavovaný parametr SD55625 $SCS_MEA_AVERAGE_VALUE[n-1]

Parametry měřicího cyklu 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)
```

Tabulka 4-15 Parametry pro volání cyklu CYCLE961 [1]

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
1		S_MVAR	Varianta měření (předdefinovaná hodnota ≥ 6)	
			Hodnoty:	JEDNOTKY: Konturový prvek 5 = seřizování polohy podle pravoúhlého vnitřního rohu, zadávané požadované hodnoty jsou úhel a vzdálenosti A1 až A3 6 = seřizování polohy podle pravoúhlého vnějšího rohu, zadávané požadované hodnoty jsou úhel a vzdálenosti A1 až A3 7 = seřizování polohy podle vnitřního rohu, zadávané požadované hodnoty jsou úhel a vzdálenosti A1 až A4 8 = seřizování polohy podle vnějšího rohu, zadávané požadované hodnoty jsou úhel a vzdálenosti A1 až A4
				DESÍTKY: Zadávání požadované hodnoty jako vzdálenost nebo pomocí čtyř bodů 0 = zadávání požadované hodnoty jako vzdálenost (polární) 1 = zadávání požadované hodnoty pomocí čtyř bodů (měřicí body P1 až P4)
				STOVKY: Cíl korekce 0 = pouze měření (žádná korekce posunutí počátku NV, příp. žádná korekce nástroje) 1 = měření a výpočet a korekce posunutí počátku NV, viz S_KNUM
				TISÍCE: Chráněná oblast 0 = žádné zohledňování chráněné oblasti (překážka) 1 = zohledňování chráněné oblasti (překážky), viz S_ID
				DESÍTKY TISÍC: Poloha rohu ve WCS 0 = poloha rohu se určuje prostřednictvím parametru S_STA1 (kompatibilita) 1 = poloha 1 rohu nastavená v počátečním bodě měření, [6] 2 = poloha 2 rohu, vzdálenosti v 1. ose roviny (v případě G17 je to osa X) jsou záporné (viz S_SETV0, S_SETV1) 3 = poloha 3 rohu, vzdálenosti v 1. a ve 2. ose roviny (v případě G17 je to rovina XY) jsou záporné (viz S_SETV0 až S_SETV3) 4 = poloha 4 rohu, vzdálenosti ve 2. ose roviny (v případě G17 je to osa Y) jsou záporné (viz S_SETV2, S_SETV3)

Č.	Parametr na obra-zovce	Parametr cyklu	Význam	
2	Volba	S_KNUM	Korekce posunutí počátku (NV) nebo základního posunutí počátku souřadného systému (NV) nebo základního vztažného bodu [2]	
			Hodno-ty:	JEDNOTKY:
				DESÍTKY: 0 = žádná korekce 1 až max. 99: číslo posunutí počátku (nuly) nebo 1 až max. 16: číslo základního posunutí počátku
				STOVKY: rezervováno
				TISÍCE: Korekce posunutí počátku (NV) nebo základního posunutí počátku nebo základního vztažného bodu 0 = korekce nastavitelného posunutí počátku (NV) 1 = korekce kanálového základního posunutí NV 2 = korekce základního vztažného bodu 9 = korekce aktivního posunutí počátku NV nebo v případě G500 do posledního aktivního kanálového základního posunutí počátku NV
				DESÍTKY TISÍC: Korekce hrubého nebo jemného posunutí počátku (NV) nebo základního posunutí počátku souřadného systému NV nebo základního vztažného bodu 0 = jemná korekce [5] 1 = hrubá korekce
3	ikona+ Počet	S_PRNUM	Číslo pole parametru měřicí sondy (nikoli číslo měřicí sondy) (rozsah hodnot = 1 až 40)	
4	L1/X1	S_SETV0	Vzdálenost L1 mezi pólem a měřicím bodem P1 ve směru 1. osy v rovině (v případě G17 je to osa X) [3] (Je-li vzdálenost L1=0, automaticky se vypočítává L1 = M_SETV1 / 2) nebo Počáteční bod P1x na 1. ose v rovině (v případě G17 je to osa X) [4]	
5	L2/Y1	S_SETV1	Vzdálenost L2 mezi pólem a měřicím bodem P2 ve směru 1. osy v rovině [3] nebo počáteční bod P1y na 2. ose v rovině (v případě G17 je to osa Y) [4]	
6	L3/X2	S_SETV2	Vzdálenost L3 mezi pólem a měřicím bodem P3 ve směru 2. osy v rovině [3] (Je-li vzdálenost L3=0, automaticky se vypočítává v případě nepravoúhlého rohu L3 = M_SETV3 / 2) nebo počáteční bod P2x na 1. ose v rovině [4]	
7	L4/Y2	S_SETV3	Vzdálenost L4 mezi pólem a měřicím bodem P3 ve směru 2. osy v rovině v případě nepravoúhlého rohu [3] nebo počáteční bod P2y na 2. ose v rovině [4]	
8	XP/X3	S_SETV4	Poloha pólu na 1. ose v rovině [3] nebo počáteční bod P3x 1. osy v rovině [4]	
9	XP/Y3	S_SETV5	Poloha pólu na 2. ose v rovině [3] nebo počáteční bod P3y 2. osy v rovině [4]	
10	X4	S_SETV6	Počáteční bod P4x na 1. ose v rovině [4]	
11	Y4	S_SETV7	Počáteční bod P4y na 2. ose v rovině [4]	
12	X0	S_SETV8	Požadovaná hodnota změřeného rohu na 1. ose v rovině v případě korekce posunutí počátku (NV)	
13	Y0	S_SETV9	Požadovaná hodnota změřeného rohu na 2. ose v rovině v případě korekce posunutí počátku (NV)	
14	α0	S_STA1	Počáteční úhel vzhledem ke kladnému směru 1. osy v rovině vůči vztažné hraně obrobku v MCS (+/-270 stupňů)	

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
15	α1	S_INCA	Úhel mezi vztažnými hranami obrobku v případě měření nepravoúhlého rohu [7)]	
16	DZ	S_ID	Hodnota přísuvu na měřicí výšku u každého měřicího bodu, když je aktivní chráněná oblast (viz S_MVAR).	
17	DFA	S_FA	Měřicí dráha	
18	TSA	S_TSA	Bezpečná oblast Kontrola úhlového rozdílu vzhledem k požadované hodnotě úhlu [stupně]	
19	Měření	S_NMSP	Počet měření na tomtéž místě [2)] (rozsah hodnot 1 až 9) [2)]	
20		S_MCBIT	rezervováno	
21		_DMODE	Režim zobrazování	
			Hodnoty:	JEDNOTKY: Rovina obrábění G17/G18/G19 0 = Kompatibilita, zůstává aktivní rovina, která byla v platnosti před voláním cyklu 1 = G17 (aktivní pouze uvnitř cyklu) 2 = G18 (aktivní pouze uvnitř cyklu) 3 = G19 (aktivní pouze uvnitř cyklu)
22		_AMODE	Alternativní režim	

1) všechny předdefinované hodnoty = 0 nebo je u nich uvedeno, že mají rozsah hodnot a až b

2) Vypisování závisí na všeobecném nastavovaném parametru SD54760 $SNS_MEA_FUNCTION_MASK_PIECE

3) Zadávání měřicích bodů v polárních souřadnicích, přičemž se bere v úvahu počáteční úhel S_STA1 u měřicího bodu 3 nebo 4 úhlový krok S_INCA.

4) Zadávání měřicích bodů v pravoúhlém souřadném systému (zadávání pomocí 4 bodů)

5) Jestliže ve strojních parametrech není aktivováno „jemné“ posunutí počátku NV, potom se korekce uskuteční po „hrubém“ posunutí počátku NV.

6) Rozsah hodnot úhlu S_INCA: -180 až +180 stupňů

7) Počáteční úhel S_STA1, rozsah hodnot: pravoúhlý roh: +- 90 stupňů, libovolný roh: +- 45 stupňů

Parametry měřicího cyklu 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)
```

Tabulka 4-16 Parametry pro volání cyklu CYCLE979 [0)]

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
1		S_MVAR	Varianta měření	
			Hodnoty:	JEDNOTKY: Konturový prvek 1 = měření díry 2 = měření čepu (hřídele)
				DESÍTKY: rezervováno
				STOVKY: Cíl korekce 0 = pouze měření (žádná korekce posunutí počátku NV, příp. žádná korekce nástroje) 1 = měření a výpočet a korekce posunutí počátku (NV) (viz S_KNUM) 2 = měření a korekce nástroje (viz S_KNUM1)
				TISÍCE: Počet měřicích bodů 0 = 3 měřicí body 1 = 4 měřicí body
				DESÍTKY TISÍC: Měření s obratem/bez obratu vřetena (diferenční měření) nebo nastavení orientace měřicí sondy ve směru spínání 0 = měření bez obratu vřetena, bez nastavování orientace měřicí sondy 1 = měření s obratem vřetena 2 = měření s nastavováním orientace měřicí sondy
2	Volba	S_KNUM	Korekce posunutí počátku (NV) nebo základního posunutí počátku souřadného systému (NV) nebo základního vztažného bodu [2)]	
			Hodnoty:	JEDNOTKY:
				DESÍTKY: 0 = žádná korekce 1 až max. 99: číslo posunutí počátku (nuly) nebo 1 až max. 16: číslo základního posunutí počátku
				STOVKY: rezervováno
				TISÍCE: Korekce posunutí počátku (NV) nebo základního posunutí počátku nebo základního vztažného bodu 0 = korekce nastavitelného posunutí počátku (NV) 1 = korekce kanálového základního posunutí NV 2 = korekce základního vztažného bodu 3 = korekce globálního základního posunutí počátku NV 9 = korekce aktivního posunutí počátku NV nebo v případě G500 posledního aktivního kanálového základního posunutí počátku NV
				DESÍTKY TISÍC: Korekce hrubého nebo jemného posunutí počátku (NV) nebo základního posunutí počátku souřadného systému NV nebo základního vztažného bodu 0 = jemná korekce [6)] 1 = hrubá korekce

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
3	Volba	S_KNUM1	Korekce do korekčních parametrů nástroje [2]	
			Hodnoty:	JEDNOTKY:
				DESÍTKY:
				STOVKY: 0 = žádná korekce 1 až max. 999: D-číslo (číslo břitu) v případě korekce nástroje; pokud jde o součtové a seřizovací korekce, viz také S_DLNUM
				TISÍCE: 0 nebo jednoznačné D-číslo
				DESÍTKY TISÍC: 0 nebo jednoznačné D-číslo 1 až max. 32000, pokud jsou ve strojních parametrech nastavena jednoznačná D-čísla
				STOVKY TISÍC: Korekční parametry nástroje [2] 0 = žádný údaj (korekce rádiusu nástroje) 1 = korekce délky L1 2 = korekce délky L2 3 = korekce délky L3 4 = korekce rádiusu
				JEDNOTKY MILIONŮ: Korekční parametry nástroje [2] 0 = žádný údaj (korekce opotřebení rádiusu nástroje) 1 = korekční parametry nástroje - součtová korekce (SK) [5] Hodnota korekce nástroje se přičítá k již existující hodnotě SK 2 = korekční parametry nástroje - seřizovací korekce (EK) [5] EK (nová) = EK (stará) + SK (stará) hodnota korekce, SK (nová) = 0 3 = korekční parametry nástroje - seřizovací korekce (EK) [5] Hodnota korekce nástroje se přičítá k již existující hodnotě EK. 4 = korekční parametry nástroje - geometrie
				DESÍTKY MILIONŮ: Korekční parametry nástroje [2] 0 = žádný údaj (korekce geometrie nástroje normálně, bez invertování) 1 = korekce invertována
				STOVKY MILIONŮ: Korekční parametry nástroje 0 = Korekční parametry nástroje bez náhradních nástroje 1 = Korekční parametry nástroje do náhradního nástroje (_DP)
4	ikona+ Počet	S_PRNUM	Číslo pole parametru měřicí sondy (nikoli číslo měřicí sondy) (rozsah hodnot = 1 až 40)	
5	X0	S_SETV	Požadovaná hodnota	
6	DFA	S_FA	Měřicí dráha	
7	TSA	S_TSA	Bezpečná oblast	
8	X0	S_CPA	Střed na 1. ose v rovině (v případě G17 je to osa X).	
9	Y0	S_CPO	Střed na 2. ose v rovině (v případě G17 je to osa Y).	
10	XM	S_XM	Zadání požadované hodnoty středu, geometrická osa X	
11	YM	S_YM	Zadání požadované hodnoty středu, geometrická osa Y	
12	alfa 0	S_STA1	Počáteční úhel [7]	
13	alfa 1	S_INCA	Úhlový krok [8]	
14	Měření	S_NMSP	Počet měření na tomtéž místě [1] (rozsah hodnot 1 až 9)	
15	ST	_DP	Číslo náhradního nástroje (číslo duplo), jehož korekce se mají uskutečnit	

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
16	T	S_TNAME	Název nástroje [2)]	
17	DL	S_DLNUM	DL-číslo seřizovací součtové korekce [1), 4)]	
18	TZL	S_TZL	Korekce nuly [1), 2)]	
19	DIF	S_TDIF	Kontrola rozdílu rozměru [1), 2)]	
20	TUL	S_TUL	Horní mezní hodnota tolerance (inkrementálně k požadované hodnotě) [2)]	
21	TLL	S_TLL	Dolní mezní hodnota tolerance (inkrementálně k požadované hodnotě) [2)]	
22	TMV	S_TMV	Oblast korekce při výpočtu střední hodnoty [1)]	
23	FW	S_K	Váhový faktor pro výpočet střední hodnoty [1)]	
24		S_EVNUM	Datový blok paměti empirické hodnoty[1), 6)]	
25		S_MCBIT	rezervováno	
26		_DMODE	Režim zobrazování	
			Hodnoty:	JEDNOTKY: Rovina obrábění G17/G18/G19 0 = Kompatibilita, zůstává aktivní rovina, která byla v platnosti před voláním cyklu 1 = G17 (aktivní pouze uvnitř cyklu) 2 = G18 (aktivní pouze uvnitř cyklu) 3 = G19 (aktivní pouze uvnitř cyklu)
27		_AMODE	Alternativní režim	
			Hodnoty:	JEDNOTKY: Tolerance měření ano/ne 0 = ne 1 = ano

0) Všechny předdefinované hodnoty = 0 nebo je u nich uvedeno, že mají rozsah hodnot a až b.

1) Vypisování závisí na všeobecném nastavovaném parametru SD54760 $SNS_MEA_FUNCTION_MASK_PIECE

2) jen v případě korekce nástroje, jinak má parametr hodnotu ""

3) jen v případě korekce nástroje a je-li pro toleranci rozměru zvoleno "Ano", jinak má parametr hodnotu = 0

4) Jen když je aktivována funkce "Seřizovací součtová korekce" v obecném strojním parametru MD 18108 $MN_MM_NUM_SUMCORR .

5) Jestliže ve strojních parametrech není aktivováno „jemné“ posunutí počátku NV, potom se korekce uskuteční po „hrubém“ posunutí počátku NV.

6) Výpočet empirické střední hodnoty je možný pouze v případě korekce nástroje.
Rozsah hodnot paměti empirických středních hodnot:
1 až 20: Číslo (n) paměti empirické hodnoty, viz kanálový nastavovaný parametr SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1]
10000 až 200000: Číslo (n) paměti střední hodnoty, viz kanálový nastavovaný parametr SD55625 $SCS_MEA_AVERAGE_VALUE[n-1]

7) Rozsah hodnot počátečního úhlu -360 až +360 stupňů

8) Rozsah hodnot úhlového kroku > 0 až ≤90 stupňů v případě 4 měřicích bodů, příp. >0 až ≤120 stupňů v případě 3 měřicích bodů.

Parametry měřicího cyklu 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)
```

Tabulka 4-17 Parametry pro volání cyklu CYCLE997 [1), 2)]

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
1		S_MVAR	Varianta měření (předdefinovaná hodnota = 9)	
			Hodnoty:	JEDNOTKY: Konturový prvek 9 = Měření koule
				DESÍTKY: Opakování měření 0 = bez opakování měření 1 = s opakováním měření
				STOVKY: Cíl korekce 0 = pouze měření (žádná korekce posunutí počátku NV) 1 = měření a výpočet a korekce posunutí počátku NV (viz S_KNUM)
				TISÍCE: Strategie měření 0 = měření rovnoběžně s osami, bez počátečního úhlu, orientace měřicí sondy podle parametru SD55740, bit 1 1 = měření po kruhové dráze, s počátečním úhlem, orientace měřicí sondy podle parametru SD55740, bit 1 2 = měření po kruhové dráze, s počátečním úhlem, měřicí sondu orientovat ve směru spínání 3 = měření rovnoběžně s osami, s počátečním úhlem, orientace měřicí sondy podle parametru SD55740, Bit 1 4 = měření rovnoběžně s osami, s počátečním úhlem, měřicí sondu orientovat ve směru spínání
				DESÍTKY TISÍC: Počet koulí, které mají být změřeny 0 = měření jedné koule 1 = měření tří koulí
				STOVKY TISÍC: Počet měřicích bodů, pouze při měření pod určitým úhlem (mějte na paměti parametr strategie měření: místo TISÍCŮ > 0) 0 = tři měřicí body při měření pod určitým úhlem (při objíždění koule) 1 = čtyři měřicí body při měření pod určitým úhlem (při objíždění koule)
				JEDNOTKY MILIONŮ: Stanovení požadované hodnoty průměru koule 0 = požadovaná hodnota průměru koule není stanovena 1 = stanovení požadované hodnoty průměru koule

Č.	Parametr na obra-zovce	Parametr cyklu	Význam	
2	Volba	S_KNUM	Korekce posunutí počátku (NV) nebo základního posunutí počátku souřadného systému nebo základního vztažného bodu [3]	
			Hodno-ty:	JEDNOTKY:
				DESÍTKY: 0 = žádná korekce 1 až max. 99: číslo posunutí počátku (nuly) nebo 1 až max. 16: číslo základního posunutí počátku
				STOVKY: rezervováno
				TISÍCE: Korekce posunutí počátku (NV) nebo základního posunutí počátku souřad-ného systému NV nebo základního vztažného bodu 0 = korekce nastavitelného posunutí počátku (NV) 1 = korekce kanálového základního posunutí NV 2 = korekce základního vztažného bodu NV 3 = korekce globálního základního posunutí počátku NV [7] 9 = korekce aktivního posunutí počátku NV nebo v případě G500 posledního aktiv-ního kanálového základního posunutí počátku NV
				DESÍTKY TISÍC: Korekce posunutí počátku (NV) nebo základního posunutí počátku souřadného systému NV nebo základního vztažného bodu hrubého nebo jemného 0 = jemná korekce [6] 1 = hrubá korekce
3	ikona+ Počet	S_PRNUM	Číslo pole parametru měřicí sondy (nikoli číslo měřicí sondy) (rozsah hodnot = 1 až 40)	
4		S_SETV	Průměr koule nebo koulí [4]	
5	DFA	S_FA	Měřicí dráha	
6	TSA	S_TSA	Bezpečná oblast	
7	alfa 0	S_STA1	Počáteční úhel v případě "Měření pod úhlem"	
8	alfa 1	S_INCA	Úhlový krok v případě "Měření pod úhlem"	
9	X1	S_SETV0	Požadovaná hodnota pozice 1. koule na 1. ose v rovině (v případě G17 je to osa X) při měření 3 koulí	
10	Y1	S_SETV1	Požadovaná hodnota pozice 1. koule na 2. ose v rovině (v případě G17 je to osa Y) při měření 3 koulí	
11	Z1	S_SETV2	Požadovaná hodnota pozice 1. koule na 3. ose v rovině (v případě G17 je to osa Z) při měření 3 koulí	
12	X2	S_SETV3	Požadovaná hodnota pozice 2. koule na 1. ose v rovině při měření 3 koulí	
13	Y2	S_SETV4	Požadovaná hodnota pozice 2. koule na 2. ose v rovině při měření 3 koulí	
14	Z2	S_SETV5	Požadovaná hodnota pozice 2. koule na 3. ose v rovině při měření 3 koulí	
15	X3	S_SETV6	Požadovaná hodnota pozice 3. koule na 1. ose v rovině při měření 3 koulí	
16	Y3	S_SETV7	Požadovaná hodnota pozice 3. koule na 2. ose v rovině při měření 3 koulí	
17	Z3	S_SETV8	Požadovaná hodnota pozice 3. koule na 3. ose v rovině při měření 3 koulí	
18	TVL	S_TNVL	Mezní hodnota pro zkreslení trojúhelníku (součet odchylek) při měření 3 koulí [5]	
19	Měření	S_NMSP	Počet měření na tomtéž místě [2] (rozsah hodnot 1 až 9)	
20		S_MCBIT	rezervováno	

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
21		_DMODE	Režim zobrazování	
			Hodnoty:	JEDNOTKY: Rovina obrábění G17/G18/G19 0 = Kompatibilita, zůstává aktivní rovina, která byla v platnosti před voláním cyklu 1 = G17 (aktivní pouze uvnitř cyklu) 2 = G18 (aktivní pouze uvnitř cyklu) 3 = G19 (aktivní pouze uvnitř cyklu)
22		_AMODE	Alternativní režim	

1) všechny předdefinované hodnoty = 0 nebo je u nich uvedeno, že mají rozsah hodnot a až b

2) Vypisování závisí na všeobecném nastavovaném parametru SD54760 $SNS_MEA_FUNCTION_MASK_PIECE

3) Najíždění do pomocných poloh při objíždění koule na jejím "rovníku".

4) Měření 3 koulí: Pro všechny koule platí stejná požadovaná hodnota průměru (_SETV).

5) Předdefinovaná hodnota pro parametr S_TNVL=1.2
Korekce posunutí počátku (NV): Korekce posunutí počátku NV se uskuteční jen tehdy, pokud zjištěné zkreslení nepřesahuje mezní hodnotu S_TNVL .

6) Jestliže ve strojních parametrech není aktivováno „jemné“ posunutí počátku, potom se korekce uskuteční po „hrubém“ posunutí počátku NV.

7) V případě varianty měření "Měření tří koulí" není možné korekce zadat do globálního základního framu (S_KNUM = 3001 až 3016), protože tento frame neobsahuje žádné rotační složky.

Parametry měřicího cyklu 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)
```

Tabulka 4-18 Parametry pro volání cyklu CYCLE995 [1)]

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
1		S_MVAR	Varianta měření (předdefinovaná hodnota=5)	
			Hodnoty:	JEDNOTKY: Konturový prvek 5 = geometrie vřetena (rovnoběžnost s osou nástroje)
				DESÍTKY: Opakování měření 1 = s opakováním měření
				STOVKY: žádný cíl korekce 0 = pouze měření
				TISÍCE: Strategie měření 2 = měření pod určitým úhlem, orientace měřicí sondy nastavena ve směru spínání
				DESÍTKY TISÍC: Počet koulí, které mají být změřeny 0 = měření jedné koule
				STOVKY TISÍC: Počet měřicích bodů 1 = 4 měřicí body při měření pod určitým úhlem (objíždění koule)
				JEDNOTKY MILIONŮ: Stanovení požadované hodnoty průměru koule 0 = požadovaná hodnota průměru koule není stanovena 1 = stanovení požadované hodnoty průměru koule
2	Volba	S_KNUM	Cíl korekce 0 = 0	
3	ikona+ Počet	S_PRNUM	Číslo pole parametru měřicí sondy (nikoli číslo měřicí sondy) (rozsah hodnot = 1 až 40)	
4	DM	S_SETV	Průměr kalibrační koule [4)]	
5	DFA	S_FA	Měřicí dráha	
6	TSA	S_TSA	Bezpečná oblast [5)]	
7	alfa 0	S_STA1	Počáteční úhel v případě "Měření pod úhlem" [3)]	
8		S_INCA	Úhlový krok v případě "Měření pod úhlem" [2)]	
9	DZ	S_DZ	Vzdálenost 1. měřicího bodu P1 od 2. měřicího bodu P2 na stopce měřicí sondy	
10		S_SETV0	Požadovaná hodnota pozice koule na 1. ose v rovině (v případě G17 je to osa X) [2)]	
11		S_SETV1	Požadovaná hodnota pozice koule na 2. ose v rovině (v případě G17 je to osa Y) [2)]	
12		S_SETV2	Požadovaná hodnota pozice koule na 3. ose v rovině (v případě G17 je to osa Z) [2)]	
13	TUL	S_TUL	Horní mezní hodnota tolerance úhlové odchylky	
14	TZL	S_TZL	Oblast nulové korekce [1), 4)]	
15	Počet	S_NMSP	Počet měření na tomtéž místě [2)] (rozsah hodnot 1 až 9)	
16		S_MCBIT	rezervováno [2)]	
17		_DMODE	Režim zobrazování	
			Hodnoty:	JEDNOTKY: Rovina obrábění G17/G18/G19 0 = Kompatibilita, zůstává aktivní rovina, která byla v platnosti před voláním cyklu 1 = G17 (aktivní pouze uvnitř cyklu) 2 = G18 (aktivní pouze uvnitř cyklu) 3 = G19 (aktivní pouze uvnitř cyklu)

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
18		_AMODE	Alternativní režim	
			Hodnoty:	JEDNOTKY: Tolerance měření ano/ne 0 = ne 1 = ano

všechny předdefinované hodnoty = 0 nebo je u nich uvedeno, že mají rozsah hodnot a až b

1) Vypisování závisí na všeobecném nastavovaném parametru SD54760 $SNS_MEA_FUNCTION_MASK_PIECE.

2) Tyto parametry nejsou v této době využívány a nezobrazují se ani ve vstupní obrazovce.
Parametry pro úhlový krok S_INCA je napevno nastaven na 90 stupňů.

3) Rozsah hodnot počátečního úhlu -360 až +360 stupňů

4) Jestliže je pro toleranci měření nastaveno "ano":
Jestliže je změřený úhel menší než hodnota oblasti nulové korekce TZL, je výsledným parametrům pro úhel (_OVR[2], _OVR[3]) a pro odchylky (_OVR[7], _OVR[8]) dosazena nulová hodnota.
Vypisování parametru TZL se uskutečňuje prostřednictvím nastavovaného parametru SD54760 $SNS_MEA_FUNCTION_MASK_PIECE bit 25=1.
(odblokování volby nulové korekce při měření úhlové chyby vřetena)

5) Parametr TSA se vztahuje na 1. měření kalibrační kuličky.

Parametry měřicího cyklu 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)
```

Tabulka 4-19 Parametry pro volání cyklu CYCLE996 [1)]

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
1		S_MVAR	Varianta měření (předdefinovaná hodnota=1)	
			Hodnoty:	JEDNOTKY: Posloupnost při měření 0 = výpočet kinematiky (je možno vybrat: Výpis výsledků, protokol, změna datového bloku naklápění, příp. potvrzení obsluhou), viz _AMODE 1 = 1. měření 2 = 2. měření 3 = 3. měření
				DESÍTKY: rezervováno 0 = 0
				STOVKY: Varianta měření při 1. až 3. měření 0 = měření kalibrační kuličky rovnoběžně s osami 1 = měření kalibrační kuličky pod určitým úhlem a žádné udržování polohy vřetena v jedné poloze [3)] 2 = měření kalibrační kuličky a udržování polohy vřetena ve směru spínání měřicí sondy [3)] 3 = měření rovnoběžně s osami, s počátečním úhlem [8)] 4 = měření rovnoběžně s osami, s počátečním úhlem, udržování polohy vřetena ve směru spínání měřicí sondy [8)]
				TISÍCE: Započítání cílové korekce do kinematiky [4)] 0 = pouze měření. Datové bloky naklápění jsou vypočítány, zůstávají ale nezměněny 1 = Vypočítat datový blok naklápění. Datové bloky naklápění jsou, v případě potřeby po potvrzení obsluhou, změněny.[4)]
				DESÍTKY TISÍC: Vypočítat měřicí osu (kruhová osa 1 nebo 2) nebo otevřený nebo zavřený vektorový řetězec kinematiky 0 = zavřený vektorový řetězec (jen při výpočtu kinematiky) 1 = kruhová osa 1 (jen v případě 1. až 3. měření) 2 = kruhová osa 2 (jen v případě 1. až 3. měření) [5)] 3 = otevřený vektorový řetězec (jen při výpočtu kinematiky)
				STOVKY TISÍC: Normování kruhové osy 1 při výpočtu kinematiky 0 = žádné normování kruhové osy 1 1 = normování ve směru 1. osy v rovině (v případě G17 je to osa X) 2 = normování ve směru 2. osy v rovině (v případě G17 je to osa Y) 3 = normování ve směru 3. osy v rovině (v případě G17 Z).
				JEDNOTKY MILIONŮ: Normování kruhové osy 2 při výpočtu kinematiky [5)] 0 = žádné normování kruhové osy 2 1 = normování ve směru 1. osy v rovině (v případě G17 je to osa X) 2 = normování ve směru 2. osy v rovině (v případě G17 je to osa Y) 3 = normování ve směru 3. osy v rovině (v případě G17 Z).
				DESÍTKY MILIONŮ: Protokolový soubor 0 = žádný protokolový soubor 1 = protokolový soubor s vypočítanými vektory (Toolcarrier) a s 1. dynamickou 5-osou transformací (TRAORI(1)), pokud jsou odpovídajícím způsobem nastaveny příslušné strojní parametry.
2		S_TC	Číslo datového bloku naklápění (Toolcarrier)	

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
3	ikona+ Počet	S_PRNUM	Počet polí parametrů měřicí sondy (nikoli číslo měřicí sondy) (předdefinovaná hodnota = 1)	
4		S_SETV	Průměr kalibrační koule	
5	alfa 0	S_STA1	Počáteční úhel v případě "Měření pod úhlem"	
6	alfa 0	S_SETV0	Hodnota polohy kruhové osy 1 (pokud je kruhová osa manuální nebo poloautomatická)	
7	alfa 1	S_SETV1	Hodnota polohy kruhové osy 2 (pokud je kruhová osa manuální nebo poloautomatická) [6]	
8	XN	S_SETV2	Hodnota polohy pro normování kruhové osy 1	
9	XN	S_SETV3	Hodnota polohy pro normování kruhové osy 2 [6]	
10	delta	S_SETV4	Hodnota tolerance offsetových vektorů I1 až I4	
11	delta	S_SETV5	Hodnota tolerance pro vektory kruhových os V1 a V2	
12	TVL	S_TNVL	Mezní hodnota úhlového segmentu kruhové osy (rozsah hodnot1 až 60 stupňů), (předdefinovaná hodnota = 20) [7]	
13	DFA	S_FA	Měřicí dráha	
14	TSA	S_TSA	Bezpečná oblast	
15	Měření	S_NMSP	Počet měření na tomtéž místě [2] (předdefinovaná hodnota = 1)	
16		S_MCBIT	rezervováno	
17		_DMODE	Režim zobrazování	
			Hodnoty:	JEDNOTKY: Rovina obrábění G17/G18/G19 0 = Kompatibilita, zůstává aktivní rovina, která byla v platnosti před voláním cyklu 1 = G17 (aktivní pouze uvnitř cyklu) 2 = G18 (aktivní pouze uvnitř cyklu) 3 = G19 (aktivní pouze uvnitř cyklu)
18		_AMODE	Alternativní režim	
			Hodnoty:	JEDNOTKY: Kontrola tolerance ano/ne 0 = ne 1 = ano: Vyhodnocování tolerančních hodnot vektorů S_SETV4, S_SETV5
				DESÍTKY: Potvrzování obsluhou při přenášení vypočítaných vektorů do datového bloku naklápění [4] 0 = ano: Obsluha musí změnu potvrdit. 1 = ne: Vypočítané vektory jsou okamžitě přenášeny (v platnosti jen tehdy, pokud je na místě STOVEK a TICÍCŮ hodnota 0).
				STOVKY: Vypisování výsledků měření [5] 0 = ne 1 = ano
				TISÍCE: Výpis výsledků měření může být editován. 0 = ne 1 = ano, a je editovatelný (v platnosti jen tehdy, pokud je na místě STOVEK hodnota 1)

[1] všechny předdefinované hodnoty = 0 nebo je u nich uvedeno, že se rovnají x

[2] Vypisování závisí na všeobecném nastavovaném parametru SD54760 $SNS_MEA_FUNCTION_MASK_PIECE.

[3] Pomocí této varianty je možné na kalibrační kuličce uskutečňovat měření například v polohách kinematiky lišících se od sebe o 90 stupňů, aniž by došlo ke kolizi s upevňovacím držákem kalibrační kuličky. Předem může být zadán počáteční úhel S_STA1 (0 až 360 stupňů. Úhlový krok při objíždění kuličky se rovná 90 stupňům.
Jako hodnota posuvu po kruhové dráze se používá kanálový nastavovaný parametr SD55634 $SCS_MEA_FEED_PLANE_VALUE .

[4] Před přenesením se vypisuje kontrolní dotaz pro obsluhu s příkazem M0. Vektory jsou přeneseny až se stisknutím tlačítka NC-Start.
Pokud je měřicí program přerušen pomocí tlačítka RESET , žádný z vypočítaných vektorů se nepřenese.
Vektory jsou přeneseny jen tehdy, pokud tolerance offsetových vektorů není při výpočtu překročena.

[5] Vypisování výsledků měření se uskutečňuje pouze při variantě měření "Výpočet kinematiky".
Pokud se má výsledek měření vypisovat i po 1. až 3. měření, můžete toho dosáhnout nastavením kanálového parametru SD 55613 $SCS_MEA_RESULT_DISPLAY.

[6] Kruhová osa 2 pouze u kinematik se dvěma kruhovými osami

[7] Mezní hodnota úhlového segmentu kruhové osy. Rozsah hodnot parametru `S_TNVL` je od 20 do 60 stupňů. V případě hodnot parametru `S_TNVL` < 20 stupňů je nutno počítat s nepřesnostmi, které jsou podmíněny nepřesnostmi při měření měřicí sondy v oblasti mikrometrů. Pokud je hodnota nižší, než tato mezní hodnota, aktivuje se chybové hlášení 61430 s údajem minima pro danou mezní hodnotu.

[8] Udržování polohy vřetena ve směru spínání měřicí sondy, pokud je v parametru SD54760 nastaven bit 17 = 1

Parametry měřicího cyklu 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)
```

Tabulka 4-20 Parametry pro volání cyklu CYCLE9960 [1)]

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
1	Volba	S_MVAR	Varianta měření (předdefinovaná hodnota=1)	
			Hodnoty:	JEDNOTKY: Varianta měření 0 = Měření a výpočet kinematiky (je možno vybrat: Výpis výsledků, protokol, změna datového bloku naklápění (viz místo tisíců S_MVAR)) 1 = měření referenční hlavičky 2 = přizpůsobení hlavičky referenční hlavičce 3 = měření a výpočet uzlových bodů (E996)
				DESÍTKY: rezervováno
				STOVKY: Varianta měření při měření koule 2 = měření kalibrační kuličky a udržování polohy vřetena ve směru spínání měřicí sondy 4 = měření rovnoběžně s osami, s počátečním úhlem, udržování polohy vřetena ve směru spínání měřicí sondy [2)]
				TISÍCE: Započítání cílové korekce do kinematiky [3)] 0 = Měření a výpočet. Datové bloky se vypočítají a zůstanou nezměněny 1 = vypočítané datové bloky, v případě potřeby po potvrzení obsluhou se provede změna 2 = již změřená kinematika stroje se vypočítá a v případě potřeby po potvrzení obsluhou se změní
				MÍSTO DESÍTEK TISÍC: měřicí osa (kruhová osa 1 nebo 2) 1 = měření a výpočet se provádí pro všechny existující kruhové osy 4 = měření a výpočet se provádí jen pro kruhovou osu 1 5 = měření a výpočet se provádí jen pro kruhovou osu 2
2		S_TNAME	Název transformace (datový blok naklápění nebo transformace na základě kinematických řetězců)	
3	ikona+ Počet	S_PRNUM	Počet polí parametrů měřicí sondy (nikoli číslo měřicí sondy) (předdefinovaná hodnota = 1)	
4		S_SETV	Průměr kalibrační koule	
5	alfa 1	S_SETV1	Počáteční úhel v případě „Měření pod úhlem“ pro 1. kruhovou osu	
6			S_START_RA1 : Počáteční úhel 1. kruhové osy	
7			S_END_RA1 : Koncový úhel 1. kruhové osy	
8			S_CMEA_RA1 : Počet měření 1. kruhové osy, 3 pro měření a výpočet kinematiky. Pro uzlové body je možných až 12 měření.	
9			S_POS_RA2 : Pozice 2. kruhové osy při měření 1. kruhové osy [4)]	
10	alfa 2	S_SETV2	Počáteční úhel v případě „Měření pod úhlem“ pro 2. kruhovou osu [4)]	
11			S_START_RA2 : Počáteční úhel 2. kruhové osy [4)]	
12			S_END_RA2 : Koncový úhel 2. kruhové osy [4)]	
13			S_CMEA_RA2 : Počet měření 2. kruhové osy, 3 pro měření a výpočet kinematiky. Pro uzlové body je možných až 12 měření.[4)]	
14			S_POS_RA1 : Pozice 1. kruhové osy při měření 2. kruhové osy	
15	delta	S_SETV4	Hodnota tolerance ofsetových vektorů	
16	DFA	S_FA	Měřicí dráha	

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
17	TSA	S_TSA	Bezpečná oblast	
18	Počet	S_NMSP	Počet měření na tomtéž místě [5] (předdefinovaná hodnota = 1)	
19		_DMODE	Režim zobrazování	
			Hodnoty:	JEDNOTKY: Rovina obrábění G17/G18/G19 0 = Kompatibilita, zůstává aktivní rovina, která byla v platnosti před voláním cyklu 1 = G17 (aktivní pouze uvnitř cyklu) 2 = G18 (aktivní pouze uvnitř cyklu) 3 = G19 (aktivní pouze uvnitř cyklu)
20		_AMODE	Alternativní režim	
			Hodnoty:	JEDNOTKY: Kontrola tolerance ano/ne 0 = ne, 1 = ano: Vyhodnocování tolerančních hodnot vektorů -> S_SETV4, S_SETV5
				DESÍTKY: Automatická kalibrace 0 = neprovádět automatickou kalibraci 1 = provádět automatickou kalibraci
				STOVKY: Automatické stanovení počátečního úhlu 0 = použije se nastavený počáteční úhel 1 = automatický výpočet počátečního úhlu, u každého měřicího bodu
21		S_KNUM	Číslo posunutí počátku pro uložení hodnoty korekce při měření referenční hlavičky	
			Hodnoty	JEDNOTKY: Korekce se vždy uskutečňuje v hrubém posunutí, jemné posunutí je vymazáno
				DESÍTKY: 1 ... 99, číslo nastavitelného posunutí počátku (1 = G54)
				TISÍCE: 9 Korekce aktivního nastavitelného posunutí počátku

[1] všechny předdefinované hodnoty = 0 nebo je u nich uvedeno, že se rovnají xx

[2] Udržování polohy vřetena ve směru spínání měřicí sondy, pokud je v parametru SD54760 nastaven bit 17 = 1

[3] Před přenesením se vypisuje kontrolní dotaz pro obsluhu s příkazem M0. Vektory se uloží, až když je stisknuto tlačítko NC-Start. Jestliže je měřicí program přerušen stisknutím tlačítka RESET, žádné vypočítané vektory se neukládají. Vektory jsou ukládány jen tehdy, pokud není při výpočtu překročena tolerance offsetových vektorů.

[4] Kruhová osa 2 pouze u kinematik se dvěma kruhovými osami

[5] Vypisování závisí na všeobecném nastavovaném parametru SD54760 $SNS_MEA_FUNCTION_MASK_PIECE.

Parametry měřicího cyklu 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)
```

Tabulka 4-21 Parametry pro volání cyklu CYCLE982 [1)]

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
1		S_MVAR	Varianta měření	
			Hodnoty:	JEDNOTKY: Kalibrace / měření 0 = kalibrace nástrojové měřicí sondy 1 = jednotlivé měření nástroje [3)] 2 = několikanásobné měření nástroje, stanovení délek a rádiusu nástroje (u frézovacích nástrojů)
				DESÍTKY: Kalibrace nebo měření v MCS nebo ve WCS 0 = vztaženo k souřadnému systému stroje [4)] 1 = vztaženo k souřadnému systému obrobku
				STOVKY: Měření frézovacího nástroje s obratem nebo bez obratu 0 = měření s obratem 1 = měření bez obratu
				TISÍCE: Cíl korekce u frézovacích nástrojů 0 = stanovení délky nebo délky a rádiusu (viz S_MVAR , místo jednotek) 1 = stanovení rádiusu, pokud místo jednotek parametru S_MVAR = 1 2 = stanovení délky a rádiusu (čelní strana), pokud místo jednotek parametru S_MVAR = 1 nebo 2 3 = stanovení horního břitu (zadní strana) kotoučové frézy a délky a rádiusu [5)]
				DESÍTKY TISÍC: Poloha frézovacího nástroje nebo vrtáku 0 = axiální poloha frézovacího nástroje nebo vrtáku, rádius ve 2. ose v rovině (v případě G18 je to osa X) [7)] 1 = radiální poloha frézovacího nástroje nebo vrtáku, rádius v 1. ose v rovině (v případě G18 je to osa Z) [7)]
				STOVKY TISÍC: Inkrementální kalibrace nebo měření 0 = žádný údaj 1 = inkrementální kalibrace nebo měření
				JEDNOTKY MILIONŮ: Nastavení polohy vřetena na počáteční úhel daný parametrem S_STA1 (jen při měření frézovacího nástroje) 0 = polohu vřetena nenastavovat 1 = vřeteno nastavit na počáteční úhel daný parametrem S_STA1
2	Volba	S_KNUM	Varianta korekce [2)]	
			Hodnoty:	JEDNOTKY: Korekční parametry nástroje 0 = žádný údaj (korekce nástroje se ukládá do geometrie) 1 = korekce nástroje se ukládají do opotřebení
3	ikona+ Počet	S_PRNUM	Počet polí parametrů měřicí sondy (nikoli číslo měřicí sondy) (předdefinovaná hodnota = 1)	
4	X0	S_MA	Měřicí osa	
			Hodnoty:	1 = 1. osa v rovině (v případě G18 Z) 2 = 2. osa v rovině (v případě G18 je to osa X)
5	+-	S_MD	Směr měření	
			Hodnoty:	0 = žádná volba (směr měření se zjišťuje ze skutečné hodnoty) 1 = kladný 2 = záporný

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
6	Z2	S_ID	Přesazení	
7	DFA	S_FA	Měřicí dráha	
8	TSA	S_TSA	Bezpečná oblast	
9	VMS	S_VMS	Proměnná měřicí rychlost při kalibraci [2]	
10	alfa1	S_STA1	Počáteční úhel při měření frézovacích nástrojů	
11	alfa2	S_CORA	Korekční úhel při měření frézovacího nástroje s obratem [8]	
12	TZL	S_TZL	Nulová korekce při měření nástrojů Při kalibraci je S_TZL = 0	
13	DIF	S_TDIF	Kontrola rozdílu rozměru	
14	Měření	S_NMSP	Počet měření na tomtéž místě [2] (předdefinovaná hodnota = 1)	
15	EVN	S_EVNUM	Číslo paměti empirické střední hodnoty [2, 9]	
16		S_MCBIT	rezervováno	
17		_DMODE	Režim zobrazování	
			Hodnoty:	JEDNOTKY: Rovina obrábění G17/G18/G19 0 = Kompatibilita, zůstává aktivní rovina, která byla v platnosti před voláním cyklu 1 = G17 (aktivní pouze uvnitř cyklu) 2 = G18 (aktivní pouze uvnitř cyklu) 3 = G19 (aktivní pouze uvnitř cyklu)
				DESÍTKY: Poloha břitu u soustružnických a frézovacích nástrojů (jen pro zobrazování ve vstupních obrazovkách, 1 - 9)
				STOVKY: Typ nástroje 0 = soustružnický nůž 1 = fréza 2 = vrták
				TISÍCE: Strategie najíždění vzhledem k nástrojové měřicí sondě 0 = PLUS [X/Z]; X, je-li nástroj v axiální poloze, Z, je-li nástroj v radiální poloze 1 = MINUS [X/Z]; X, je-li nástroj v axiální poloze, Z, je-li nástroj v radiální poloze
18		_AMODE	Alternativní režim	
			Hodnoty:	JEDNOTKY: rezervováno
				DESÍTKY: rezervováno
				STOVKY: rezervováno
				TISÍCE: Najíždění na počáteční pozici po dokončení operace měření při kalibraci a jednotlivém měření (viz S_MVAR - JEDNOTKY) 0 = nástroj se nachází posunutí o vzdálenost DFA oproti hraně měřicí sondy 1 = najíždění na počáteční pozici

[1] všechny předdefinované hodnoty = 0 nebo je u nich uvedeno, že se rovnají x

[2] Vypisování závisí na všeobecném nastavovaném parametru SD54762 _MEA_FUNCTION_MASK_TOOL

[3] Měření soustružnického nebo frézovacího nástroje nebo vrtáku. Měřicí osa v parametru S_MA
Specifikace u soustružnických nástrojů pomocí polohy břitu 1...8, u frézovacích nástrojů prostřednictvím míst STOVEK a TISÍCŮ v parametru S_MVAR.

[4] Měření a kalibrace se uskutečňují v základním souřadném systému (MCS s deaktivovanou kinematickou transformací).

[5] nikoli pro inkrementální měření

[6] jen pro několikanásobná měření S_MVAR=x2x02 nebo x3x02 (například v případě kotoučové nebo drážkovací frézy)

7) Pokud je nastaven kanálový parametr SD42950 $SC_TOOL_LENGTH_TYPE = 2, uskutečňuje se přiřazení délkových složek nástroje stejně jako u soustružnických nástrojů.

8) jen při měření s obratem `S_MVAR=xx1x1`

9) Výpočet empirické hodnoty
Rozsah hodnot paměti empirické hodnoty: 1 až 20, číslo (n) paměti empirických středních hodnot viz kanálový nastavovaný parametr SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1].

Parametry měřicího cyklu 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)
```

Tabulka 4-22 Parametry pro volání cyklu CYCLE971 1)

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
1		`S_MVAR`	Varianta měření	
			Hodnoty:	JEDNOTKY: 0 = kalibrace nástrojové měřicí sondy 1 = měření nástroje se zastaveným vřetenem (délka, příp. rádius) 2 = měření nástroje s otáčejícím se vřetenem (délka, příp. rádius), viz parametry `S_F1` až `S_S4`
				DESÍTKY: Měření v MCS nebo ve WCS 0 = měření v MCS (vztaženo ke stroji), měření nástroje nebo kalibrace nástrojové měřicí sondy 1 = měření ve WCS (vztaženo k obrobku), měření nástroje nebo kalibrace nástrojové měřicí sondy
				STOVKY: Kontrola jednotlivých zubů 0 = ne 1 = ano
				TISÍCE: 0 = 0
				DESÍTKY TISÍC: Inkrementální kalibrace nebo měření 0 = žádný údaj 1 = inkrementální kalibrace nebo měření
				STOVKY TISÍC: Automatická kalibrace nástrojové měřicí sondy 0 = neprovádět automatickou kalibraci nástrojové měřicí sondy 1 = provádět automatickou kalibraci nástrojové měřicí sondy
				JEDNOTKY MILIONŮ: Kalibrace v rovině s obratem vřetena 0 = kalibrace v rovině bez obratu vřetena 1 = kalibrace v rovině s obratem vřetena

Č.	Parametr na obra-zovce	Parametr cyklu	Význam	
2	Volba	S_KNUM	Varianta korekce [2]	
			Hodno-ty:	JEDNOTKY: Korekční parametry nástroje 0 = žádný údaj (korekce nástroje se ukládá do geometrie) 1 = korekce nástroje se ukládají do opotřebení
3	ikona+ Počet	S_PRNUM	Počet polí parametrů měřicí sondy (nikoli číslo měřicí sondy)	
4	X0	S_MA	Měřicí osa, osa posuvu [4]	
			Hodno-ty:	JEDNOTKY: Číslo měřicí osy 1 = 1. osa v rovině (v případě G17 X) 2 = 2. osa v rovině (v případě G17 Y) 3 = 3. osa v rovině (v případě G17 Z).
				DESÍTKY: 0 = 0
				STOVKY: Číslo osy posuvu 0 = žádná osa pro předsazení 1 = 1. osa v rovině (v případě G17 X) 2 = 2. osy v rovině (v případě G17 je to osa Y)
5	+-	S_MD	Směr měření	
			Hodno-ty:	0 = žádná volba (směr měření se zjišťuje ze skutečné hodnoty) 1 = kladný 2 = záporný
6	V	S_ID	Přesazení	
			Hodno-ty:	0 = u nástrojů bez přesazení >0 = • kalibrace: Přesazení se uplatňuje na 3. osu v rovině (v případě G17 je to osa Z), pokud je průměr kalibračního nástroje větší než je průměr měřicí sondy u jejího horního okraje. Nástroj je přesazený o vzdálenost rovnající se jeho rádiusu od středu měřicí sondy mínus hodnota v parametru S_ID. Osa, v níž je přesazení realizováno, se zadává v parametru S_MA . • Měření: V případě nástrojů s více břity musí být specifikovány přesazení délky nástroje a nejvyšší bod břitu při měření rádiusu a přesazení rádiusu nástroje oproti nejvyššímu bodu břitu při měření délky nástroje.
7	DFA	S_FA	Měřicí dráha	
8	TSA	S_TSA	Bezpečná oblast	
9	VMS	S_VMS	Proměnná měřicí rychlost při kalibraci [2]	
10	TZL	S_TZL	Nulová korekce (jen při měření nástroje)	
11	DIF	S_TDIF	Kontrola rozdílu rozměru při měření nástroje (S_MVAR=xx1 nebo S_MVAR=xx2)	
12	Měření	S_NMSP	Počet měření na tomtéž místě [2]	
13	F1	S_F1	1. posuv pro snímání s otáčejícím se vřetenem [2]	
14	S1	S_S1	1. otáčky pro snímání s otáčejícím se vřetenem [2]	
15	F2	S_F2	2. posuv pro snímání s otáčejícím se vřetenem [2]	
16	S2	S_S2	2. otáčky pro snímání s otáčejícím se vřetenem [2]	
17	F3	S_F3	3. posuv pro snímání s otáčejícím se vřetenem [2]	
18	S3	S_S3	3. otáčky pro snímání s otáčejícím se vřetenem [2]	

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
19	EVN	S_EVNUM	Číslo paměti empirické hodnoty[2)]	
20		S_MCBIT	Maska _CBIT příp. _CHBIT	
21		_DMODE	Režim zobrazování	
			Hodnoty:	JEDNOTKY: Rovina obrábění G17/G18/G19 0 = Kompatibilita, zůstává aktivní rovina, která byla v platnosti před voláním cyklu 1 = G17 (aktivní pouze uvnitř cyklu) 2 = G18 (aktivní pouze uvnitř cyklu) 3 = G19 (aktivní pouze uvnitř cyklu)
22		_AMODE	Alternativní režim	
			Hodnoty:	JEDNOTKY: Přesazení nástroje při měření rádiusu 1 = ne 2 = ano
				DESÍTKY: Směr přesazení nástroje při měření rádiusu ve 3. ose v rovině (v případě G17 je to osa Z) 1 = kladný 2 = záporný
				STOVKY: Přesazení nástroje při měření délky, příp. při kalibraci měřicí sondy ve směru 3. osy 0 = kompatibilita, automaticky 1 = ne 2 = ano
				TISÍCE: Směr přesazení nástroje při měření délky ve směru osy, která provádí přesazení (viz S_MA, místo STOVEK) 1 = kladný 2 = záporný

1) všechny předdefinované hodnoty = 0 nebo je u nich uvedeno, že se rovnají x

2) Vypisování závisí na všeobecném nastavovaném parametru SD54762 MEA_FUNCTION_MASK_TOOL

3) jen v případě korekce nástroje a je-li pro toleranci rozměru zvoleno "Ano", jinak má parametr hodnotu = 0

4) při automatickém měření (S_MVAR=1x00xx) se měřicí osa nevypisuje, osa posuvu => S_MA=0.

Parametry měřicího cyklu CYCLE150

```
PROC CYCLE150(INT S_PICT,INT S_PROT,STRING[160] S_PATH) SAVE
ACTBLOCNO DISPLOF
```

Tabulka 4-23 Parametry pro volání cyklu CYCLE150

Č.	Parametr na obrazovce	Parametr cyklu	Význam	
1	Obrazovka s výsledky měření	S_PICT	Volba zobrazování výsledků (předdefinovaná hodnota = 0)	
			Hodnoty:	JEDNOTKY: 0 = obrazovku s výsledky měření nezobrazovat 1 = obrazovku s výsledky měření zobrazovat
				DESÍTKY: Volba režimu zobrazování (hodnoty stejné jako u parametru SD55613) 1 = obrazovku s výsledky měření zobrazovat - automaticky zrušit po 8 s 3 = obrazovku s výsledky zobrazovat - potvrdit tlačítkem NC-Start 4 = obrazovku s výsledky měření zobrazovat - jen v případě alarmů (61303 ... 61306)
2		S_PROT	Volba ukládání dat do protokolu (předdefinovaná hodnota = 0)	
	Protokol		Hodnoty:	JEDNOTKY: Volba protokolu zap. / vyp. / poslední měření 0 = protokol deaktivován 1 = protokol aktivován 2 = do protokolu ukládat poslední měření
	Typ protokolu			DESÍTKY: Volba typu protokolu 0 = standardní protokol 1 = uživatelský protokol (volně definovatelný)
	Formát protokolu			STOVKY: Volba formátu protokolu 0 = textový formát 1 = tabulkový formát (pro import do Excelu)
	Data protokolu			TISÍCE: Volba zápisu do nového protokolu nebo připojení k již existujícímu 0 = nový 1 = připojit na konec
	Protokol Ukládání			DESÍTKY TISÍC: Volba místa pro uložení protokolu 0 = stejné jako výrobní program 1 = adresář
3		S_PATH	Cesta do adresáře pro protokolový soubor v souladu se zvoleným místem pro uložení protokolu (kompletní údaj cesty nebo jen název souboru, např.: "//NC:/WKS.DIR/NAME.WPD nebo "MESSPROTOKOLL.TXT"	

Tabulky

5

5.1 Příkazy

Poznámka

Cykly

Tento seznam příkazů obsahuje všechny cykly, které se mohou v NC programu (G-kódy) vyskytnout, tzn. které mohou být v editoru programu naprogramovány pomocí obrazovek, příp. které musí být v případě broušení naprogramovány bez podpory programování. Není brát ohled na cykly, které jsou z důvodů kompatibility v řídícím systému stále ještě k dispozici, ale prostřednictvím programového editoru systému SINUMERIK Operate už je není možné editovat („kompatibilní cykly“).

Příkazy A ... C

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
:	O	Číslo hlavního NC-bloku, ukončení návěští skoku, operátor zřetězení		+		PM-NC
*	O	Operátor pro násobení		+		PM-NC
+	O	Operátor pro sečítání		+		PM-NC
-	O	Operátor pro odečítání		+		PM-NC
<	O	Porovnávací operátor, je menší než		+		PM-NC
<<	O	Operátor zřetězení pro řetězce		+		PM-NC
<=	O	Porovnávací operátor, je menší nebo rovno než		+		PM-NC
=	O	Operátor přiřazení		+		PM-NC
>=	O	Porovnávací operátor, je větší nebo rovno než		+		PM-NC
/	O	Operátor pro dělení		+		PM-NC
/0 ... ... /7		Blok se bude přeskakovat (1. přeskakovaná úroveň)° ... ... blok se bude přeskakovat (8. přeskakovaná úroveň)		+		PM-NC
A	A	Název osy	m/s	+		PM-NC
A2	A	Orientace nástroje: Úhel RPY nebo Eulerův úhel	s	+		PM-NC
A3	A	Orientace nástroje: 1. Složka směrového vektoru	s	+		PM-NC
A4	A	Orientace nástroje: 1. Složka normálového vektoru plochy na začátku bloku	s	+		PM-NC
A5	A	Orientace nástroje: 1. Složka normálového vektoru plochy na konci bloku	s	+		PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
A6	A	Orientace nástroje: 1. Složka směrového vektoru pro osu kužele	s	+		PM-NC
A7	A	Orientace nástroje: 1. Vektorová složka orientace ve vnitřním bodě na plášťové ploše kužele	s	+		PM-NC
ABS	F	Absolutní hodnota		+	+	PM-NC
AC	K	Absolutní zadání rozměru pro souřadnice/polohy	s	+		PM-NC
ACC	K	Ovlivňování aktuálního zrychlení osy	m	+	+	PM-NC
ACCLIMA	K	Ovlivňování aktuálního maximálního zrychlení osy	m	+	+	PM-NC
ACN	K	Zadání absolutního rozměru pro kruhovou osu, na pozici se najíždí v záporném směru	s	+		PM-NC
ACOS	F	arkus kosinus (goniometr. funkce)		+	+	PM-NC
ACP	K	Zadání absolutního rozměru pro kruhovou osu, na pozici se najíždí v kladném směru	s	+		PM-NC
ACTBLOCNO	P	Výstup aktuální čísla alarmového bloku, a to i když je aktivní příkaz "Potlačit výpis aktuálního bloku" (DISPLOF)!		+		PM-NC
ADDFRAME	F	Započítání a případné aktivování změřeného framu		+	-	PM-NC, FM-B
ADIS	A	Vzdálenost zaoblení pro funkce pohybu po dráze G1, G2, G3, ...	m	+		PM-NC
ADISPOS	A	Vzdálenost zaoblení pro rychlý posuv G0	m	+		PM-NC
ADISPOSA	P	Velikost tolerančního okna pro IPOBRKA	m	+	+	PM-NC
AFISOF	P	Deaktivování automatického přepínání filtračního řetězce	m	+	-	PM-NC
AFISON	P	Aktivování automatického přepínání filtračního řetězce	m	+	-	PM-NC
ALF	A	Úhel rychlého zvedání	m	+		PM-NC
AMIRROR	G	Programovatelné zrcadlové převrácení	s	+		PM-NC
AND	K	Logické A		+		PM-NC
ANG	A	Úhel konturové křivky	s	+		PM-NC
AP	A	Polární úhel	m/s	+		PM-NC
APR	K	Načtení / výpis přístupové ochrany		+		PM-NC
APRB	K	Načtení přístupové ochrany, BTSS		+		PM-NC
APRP	K	Načtení přístupové ochrany, výrobní program		+		PM-NC
APW	K	Zápis přístupové ochrany		+		PM-NC
APWB	K	Zápis přístupové ochrany, BTSS		+		PM-NC
APWP	K	Zápis přístupové ochrany, výrobní program		+		PM-NC
APX	K	Definice ochrany proti přístupu pro provedení uvedeného prvku jazyka		+		PM-NC
AR	A	Úhel kruhové výseče	m/s	+		PM-NC
AROT	G	Programovatelné otočení	s	+		PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
AROTS	G	Programovatelné otáčení framu o prostorový úhel	s	+		PM-NC
AS	K	Definice makra		+		PM-NC
ASCALE	G	Programovatelná změna měřítka	s	+		PM-NC
ASIN	F	Matematická funkce, arkus sinus		+	+	PM-NC
ASPLINE	G	Akimovy spliny	m	+		PM-NC
ATAN2	F	arkus tangens2		+	+	PM-NC
ATOL	A	Tolerance pro specifickou osu pro funkce kompresoru, vyhlazení orientace a druhy zaoblování konturových přechodů	m	+		PM-NC
ATRANS	G	Aditivní programovatelné posunutí počátku	s	+		PM-NC
AUXFUDEL	P	Vymazání pomocné kanálové funkce z globálního seznamu		+	-	FM-B
AUXFUDELG	P	Vymazání všech pomocných kanálových funkcí z dané skupiny z globálního seznamu		+	-	FM-B
AUXFUMSEQ	P	Zjištění posloupnosti při výstupu pomocných M-funkcí		+	-	FM-B
AUXFUSYNC	P	Vygenerování řetězce, který reprezentuje kompletní blok výrobního programu pro kanálový příkaz SERUPRO-konec-ASUP, z globálního seznamu pomocných funkcí		+	-	FM-B
AX	K	Proměnný identifikátor osy	m/s	+		PM-NC
AXCTSWE	P	Otočení osového zásobníku		+	-	PM-NC
AXCTSWEC	P	Uvolnění pro reset otočení osového zásobníku		+	+	PM-NC
AXCTSWED	P	Otočení osového zásobníku (varianta příkazu pro uvádění do provozu!)		+	-	PM-NC
AXIS	K	Identifikátor osy, adresa osy		+		PM-NC
AXNAME	F	Konvertovaný vstupní řetězec v identifikátoru osy		+	-	PM-NC
AXSTRING	F	Konvertuje řetězec čísla vřetena		+	-	PM-NC
AXTOCHAN	P	Vyžádání osy pro určitý kanál Je možné z NC programu a ze synchronních akcí.		+	+	PM-NC
AXTOSPI	F	Konvertuje identifikátor osy na index vřetena		+	-	PM-NC
B	A	Název osy	m/s	+		PM-NC
B2	A	Orientace nástroje: Úhel RPY nebo Eulerův úhel	s	+		PM-NC
B3	A	Orientace nástroje: Složka směrového vektoru	s	+		PM-NC
B4	A	Orientace nástroje: 2. Složka normálového vektoru plochy na začátku bloku	s	+		PM-NC
B5	A	Orientace nástroje: 2. Složka normálového vektoru plochy na konci bloku	s	+		PM-NC
B6	A	Orientace nástroje: 2. Složka směrového vektoru pro osu kužele	s	+		PM-NC
B7	A	Orientace nástroje: 2. Vektorová složka orientace ve vnitřním bodě na plášťové ploše kužele	s	+		PM-NC
B_AND	O	Bitové logické A		+		PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
B_OR	O	Bitové logické NEBO		+		PM-NC
B_NOT	O	Bitová negace		+		PM-NC
B_XOR	O	Bitové logické XOR		+		PM-NC
BAUTO	G	Definice prvního splinového úseku prostřednictvím následujících tří bodů	m	+		PM-NC
BLOCK	K	Spolu s klíčovým slovem TO definuje úsek programu v nepřímo volaném podprogramu, který se má zpracovat.		+		PM-NC
BLSYNC	K	Zpracování rutiny přerušení má být zahájeno teprve s následující změnou bloku.		+		PM-NC
BNAT [6]	G	Přirozený přechod na první splinový blok	m	+		PM-NC
BOOL	K	Datový typ: Logické hodnoty TRUE/FALSE, příp. 1/0		+		PM-NC
BOUND	F	Zkontroluje, zda hodnota leží v rámci definovaného rozsahu hodnot. V případě rovnosti se vrací zpět zkoušená hodnota.		+	+	PM-NC
BRISK [6]	G	Skokové změny zrychlení po dráze	m	+		PM-NC
BRISKA	P	Aktivování skokových změn zrychlení při pohybu po dráze pro naprogramované osy		+	-	PM-NC
BSPLINE	G	B-Spline	m	+		PM-NC
BTAN	G	Tangenciální přechod na první splinový blok	m	+		PM-NC
C	A	Název osy	m/s	+		PM-NC
C2	A	Orientace nástroje: Úhel RPY nebo Eulerův úhel	s	+		PM-NC
C3	A	Orientace nástroje: 3. Složka směrového vektoru	s	+		PM-NC
C4	A	Orientace nástroje: 3. Složka normálového vektoru plochy na začátku bloku	s	+		PM-NC
C5	A	Orientace nástroje: 3. Složka normálového vektoru plochy na konci bloku	s	+		PM-NC
C6	A	Orientace nástroje: 3. Složka směrového vektoru pro osu kužele	s	+		PM-NC
C7	A	Orientace nástroje: 3. Vektorová složka orientace ve vnitřním bodě na plášťové ploše kužele	s	+		PM-NC
CAC	K	Absolutní najíždění na pozici		+		PM-NC
CACN	K	Na hodnotu uloženou v tabulce se bude najíždět v záporném směru a hodnota bude považována za absolutní.		+		PM-NC
CACP	K	Na hodnotu uloženou v tabulce se bude najíždět v kladném směru a hodnota bude považována za absolutní.		+		PM-NC
CADAPTOF	P	Deaktivování vazby přizpůsobování zátěže		+	-	PM-NC
CADAPTON	P	Aktivování přizpůsobování zátěže		+	-	PM-NC
CALCDAT	F	Tato funkce vypočítá rádius a střed kruhu ne základě 3 nebo 4 bodů.		+	-	PM-NC
CALCFIR	P	Přizpůsobení filtru ryvu FIR režimu dynamiky		+	-	PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
CALCPOSI	F	Kontrola narušení chráněné oblasti, ohraničení pracovního pole a softwarové mezní hodnoty		+	-	PM-NC
CALL	K	Nepřímé volání podprogramu		+		PM-NC
CALLPATH	P	Programovatelné vyhledávání cesty při volání podprogramu		+	-	PM-NC
CANCEL	P	Přerušení modální synchronní akce		+	-	FM-SA
CANCELSUB	P	Přerušení aktuální úrovně podprogramu		+	+	FM-SA
CASE	K	Podmíněné větvení programu		+		PM-NC
CDC	K	Přímé najíždění na pozici		+		PM-NC
CDOF [6]	G	Deaktivování monitorování kolize	m	+		PM-NC
CDOF2	G	Deaktivování monitorování kolize, při 3D obvodovém frézování	m	+		PM-NC
CDON	G	Aktivování monitorování kolize	m	+		PM-NC
CFC [6]	G	Konstantní posuv na kontuře	m	+		PM-NC
CFIN	G	Konstantní posuv, jen na vnitřních zakřiveních, ne na vnějších zakřiveních	m	+		PM-NC
CFINE	F	Přiřazení jemného posunutí proměnné typu FRAME		+	-	PM-NC
CFTCP	G	Konstantní posuv na vztažném bodu břitu nástroje, dráha středu nástroje	m	+		PM-NC
CHAN	K	Specifikace rozsahu platnosti dat		+		PM-NC
CHANDATA	P	Nastavení čísla kanálu pro přístup ke kanálovým datům		+	-	PM-NC
CHAR	K	Datový typ: Znaky ASCII		+		PM-NC
CHF	A	Faseta; hodnota = délka fasety	s	+		PM-NC
CHKDM	F	Zkouška jednoznačnosti v rámci zásobníku		+	-	FM-TM
CHKDNO	F	Kontrola jednoznačnosti D-čísel		+	-	PM-NC
CHR	A	Faseta; Hodnota = délka fasety ve směru pohybu		+		PM-NC
CIC	K	Inkrementální najíždění na pozici		+		PM-NC
CIP	G	Kruhová interpolace přes vnitřní bod	m	+		PM-NC
CLEARM	P	Vymazání jedné/více značek pro přiřazení kanálu		+	+	PM-NC
CLRINT	P	Deaktivování přerušení		+	-	PM-NC
CMIRROR	F	Zrcadlové převrácení souřadné osy.		+	-	PM-NC
COARSEA	K	Konec pohybu při dosažení hrubého okna přesného najetí	m	+		PM-NC
COLLPAIR	F	Kontrola příslušnosti k protikolizní dvojici		+		PM-NC
COMPCAD	G	Aktivování funkce kompresoru COMPCAD	m	+		PM-NC
COMPCURV	G	Aktivování funkce kompresoru COMPCURV	m	+		PM-NC
COMPLETE		Řídící příkaz pro odesílání a příjem dat		+		PM-NC
COMPOF [6]	G	Deaktivování komprese NC bloků	m	+		PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
COMPON	G	Aktivování funkce kompresoru COMPON	m	+		PM-NC
COMPSURF	G	Aktivování funkce kompresoru COMPSURF	m	+		PM-NC
CONTDCON	P	Aktivování dekódování kontury v tabulkové formě		+	-	PM-NC
CONTPRON	P	Aktivování referenční přípravy		+	-	PM-NC
CORROF	P	Všechny aktivní superponované pohyby jsou deaktivovány.		+	-	PM-NC
CORRTC	F	Modifikace vektorů offsetů nebo směrových vektorů orientovatelného držáku nástroje po měření stroje.		+	-	PM-NC
CORRTRAFO	F	Změna offsetových vektorů nebo směrových vektorů orientačních os v kinematickém modelu stroje		+	-	PM-NC
COS	F	kosinus (goniometr. funkce)		+	+	PM-NC
COUPDEF	P	Definice vazby ELG / vazby synchron. vřeten		+	-	PM-NC
COUPDEL	P	Zrušení vazby ELG		+	-	PM-NC
COUPOF	P	Deaktivování vazby ELG / dvojice synchronních vřeten		+	-	PM-NC
COUPOFS	P	Vypnutí vazby ELG / dvojice synchronních vřeten se zastavením vlečného vřetena		+	-	PM-NC
COUPON	P	Aktivování vazby ELG / dvojice synchronních vřeten		+	-	PM-NC
COUPONC	P	Zapnutí vazby ELG / dvojice synchronních vřeten, převzít předešlé naprogramování		+	-	PM-NC
COUPRES	P	Reset vazby ELG		+	-	PM-NC
CP [6]	G	Pohyb po dráze	m	+		PM-NC
CPBC	K	Generická vazba: Kritérium přechodu na další blok		+	+	FM-A
CPDEF	K	Generická vazba: Definice režimu vazby		+	+	FM-A
CPDEL	K	Generická vazba: Vymazání režimu vazby		+	+	FM-A
CPFMOF	K	Generická vazba: Chování vlečné osy při úplném deaktivování		+	+	FM-A
CPFMON	K	Generická vazba: Chování vlečné vazby při aktivování		+	+	FM-A
CPFMSON	K	Generická vazba: Režim synchronní akce		+	+	FM-A
CPFPOS	K	Generická vazba: Synchronní pozice vlečné osy		+	+	FM-A
CPFRS	K	Generická vazba: Vztažný souřadný systém		+	+	FM-A
CPLA	K	Generická vazba: Definice řídící osy		+	-	FM-A
CPLCTID	K	Generická vazba: Číslo tabulky křivky		+	+	FM-A
CPLDEF	K	Generická vazba: Definice řídící osy a definice režimu osy		+	+	FM-A
CPLDEL	K	Generická vazba: Vymazání řídící osy a režimu vazby		+	+	FM-A
CPLDEN	K	Generická vazba: Jmenovatel faktoru vazby		+	+	FM-A

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
CPLINSC	K	Generická vazba: Faktor měřítka pro vstupní hodnotu řídící osy		+	+	FM-A
CPLINTR	K	Generická vazba: Hodnota posunutí pro vstupní hodnotu řídící osy		+	+	FM-A
CPLNUM	K	Generická vazba: Čitatel faktoru vazby		+	+	FM-A
CPLOF	K	Generická vazba: Deaktivování řídící osy režimu vazby		+	+	FM-A
CPLON	K	Generická vazba: Aktivování řídící osy režimu vazby		+	+	FM-A
CPLOUTSC	K	Generická vazba: Faktor měřítka pro výstupní hodnotu vazby		+	+	FM-A
CPLOUTTR	K	Generická vazba: Hodnota posunutí pro výstupní hodnotu vazby		+	+	FM-A
CPLPOS	K	Generická vazba: Synchronní pozice řídící osy		+	+	FM-A
CPLSETVAL	K	Generická vazba: Vztažný objekt vazby		+	+	FM-A
CPMALARM	K	Generická vazba: Potlačení výstupu speciálních alarmů vztahujících se k vazbě		+	+	FM-A
CPMBRAKE	K	Generická vazba: Chování vlečné osy při určitých signálech a příkazech pro zastavení		+	-	FM-A
CPMPRT	K	Generická vazba: Chování vazby při spuštění výrobního programu pomocí hledání bloku prostřednictvím zkušebního zpracování programu		+	+	FM-A
CPMRESET	K	Generická vazba: Chování vazby při resetu		+	+	FM-A
CPMSTART	K	Generická vazba: Chování vazby při spuštění výrobního programu		+	+	FM-A
CPMVDI	K	Generická vazba: Chování vlečné osy při určitých signálech rozhraní NC/PLC		+	+	FM-A
CPOF	K	Generická vazba: Deaktivování režimu vazby		+	+	FM-A
CPON	K	Generická vazba: Aktivování režimu vazby		+	+	FM-A
CPRECOF [6]	G	Deaktivování programovatelné přesnosti kontury	m	+		PM-NC
CPRECON	G	Aktivování programovatelné přesnosti kontury	m	+		PM-NC
CPRES	K	Generická vazba: Aktivování dat vazby synchronních vřeten nastavených v konfiguraci		+	-	FM-A
CPROT	P	Aktivování/deaktivování chráněné oblasti pro specifický kanál		+	-	PM-NC
CPROTDEF	P	Definice chráněné oblasti pro specifický kanál		+	-	PM-NC
CPSETTYPE	K	Generická vazba: Typ vazby		+	+	FM-A
CPSYNCOP	K	Generická vazba: Prahová hodnota pro "hrubé" synchronní najíždění na určitou pozici		+	+	FM-A
CPSYNCOP2	K	Generická vazba: Prahová hodnota pro "hrubé" synchronní najíždění 2 na určitou pozici		+	+	FM-A
CPSYNCOV	K	Generická vazba: Mezní hodnota pro "hrubé" synchronní najíždění na určitou rychlost		+	+	FM-A

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
CPSYNFIP	K	Generická vazba: Prahová hodnota pro "jemné" synchronní najíždění na určitou pozici		+	+	FM-A
CPSYNFIP2	K	Generická vazba: Prahová hodnota pro "jemné" synchronní najíždění 2 na určitou pozici		+	+	FM-A
CPSYNFIV	K	Generická vazba: Mezní hodnota pro "jemné" synchronní najíždění na určitou rychlost		+	+	FM-A
CR	A	Rádius kruhu	s	+		PM-NC
CROT	F	Otočení aktuálního souřadného systému		+	-	PM-NC
CROTS	F	Programovatelné otočení framu o prostorový úhel (otáčení v uvedených osách)	s	+	-	PM-NC
CRPL	F	Otočení framu v libovolné rovině		+	-	FM-B
CSCALE	F	Faktor změny měřítka pro více os		+	-	PM-NC
CSPLINE	F	Kubické spliny	m	+		PM-NC
CT	G	Kruh s tangenciálním přechodem	m	+		PM-NC
CTAB	F	Zjistit polohu vlečné osy z tabulky křivek na základě polohy řídící osy		+	+	PM-NC
CTABDEF	P	Aktivování definice tabulky		+	-	PM-NC
CTABDEL	P	Vymazání tabulky křivek		+	-	PM-NC
CTABEND	P	Deaktivování definice tabulky		+	-	PM-NC
CTABEXISTS	F	Kontrola tabulky křivek s číslem n		+	+	PM-NC
CTABFNO	F	Počet ještě možných tabulek křivek v paměti		+	+	PM-NC
CTABFPOL	F	Počet ještě možných polynomů v paměti		+	+	PM-NC
CTABFSEG	F	Počet ještě možných křivkových segmentů v paměti		+	+	PM-NC
CTABID	F	Zjištění čísla tabulky n-té tabulky křivek		+	+	PM-NC
CTABINV	F	Zjistit polohu řídící osy z tabulky křivek na základě polohy vlečné osy		+	+	PM-NC
CTABISLOCK	F	Deaktivování zablokování tabulky křivek s číslem n		+	+	PM-NC
CTABLOCK	P	Zablokování mazání a přepisování		+	+	PM-NC
CTABMEMTYP	F	Zjištění paměti, ve které je uložena tabulka křivek s číslem n.		+	+	PM-NC
CTABMPOL	F	Maximální možný počet polynomů v paměti		+	+	PM-NC
CTABMSEG	F	Maximální možný počet křivkových segmentů v paměti		+	+	PM-NC
CTABNO	F	Počet definovaných křivkových tabulek v paměti SRAM nebo DRAM		+	+	FM-A
CTABNOMEM	F	Počet definovaných křivkových tabulek v paměti SRAM nebo DRAM		+	+	PM-NC
CTABPERIOD	F	Zjištění periodicity tabulky křivek s číslem n		+	+	PM-NC
CTABPOL	F	Počet již používaných polynomů v paměti		+	+	PM-NC
CTABPOLID	F	Počet křivkových polynomů používaných tabulkou křivek s číslem n		+	+	PM-NC
CTABSEG	F	Počet již používaných křivkových segmentů v paměti		+	+	PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
CTABSEGID	F	Počet křivkových segmentů používaných tabulkou křivek s číslem n		+	+	PM-NC
CTABSEV	F	Zjištění koncové hodnoty vlečné osy segmentu křivkové tabulky		+	+	PM-NC
CTABSSV	F	Zjištění počáteční hodnoty vlečné osy segmentu křivkové tabulky		+	+	PM-NC
CTABTEP	F	Zjištění hodnoty řídící osy na konci křivkové tabulky		+	+	PM-NC
CTABTEV	F	Zjištění hodnoty vlečné osy na konci křivkové tabulky		+	+	PM-NC
CTABTMAX	F	Zjištění maximální hodnoty vlečné osy křivkové tabulky		+	+	PM-NC
CTABTMIN	F	Zjištění minimální hodnoty vlečné osy křivkové tabulky		+	+	PM-NC
CTABTSP	F	Zjištění hodnoty řídící osy na začátku křivkové tabulky		+	+	PM-NC
CTABTSV	F	Zjištění hodnoty vlečné osy na začátku křivkové tabulky		+	+	PM-NC
CTABUNLOCK	P	Odstranění blokování mazání a přepisování		+	+	PM-NC
CTOL	A	Tolerance kontury pro funkce kompresoru, vyhlazení orientace a druhy zaoblování konturových přechodů	m	+	-	PM-NC
CTOLG0	A	Tolerance kontury při pohybech s rychlým posuvem	m	+	-	PM-NC
CTRANS	F	Posunutí počátku pro více os.		+	-	PM-NC
CUT2D [6]	G	2D korekce rádiusu nástroje	m	+		PM-NC
CUT2DD	G	2½-D-korekce rádiusu nástroje vztažená na diferenční nástroj	m	+		PM-NC
CUT2DF	G	2D korekce rádiusu nástroje, vzhledem k aktuálnímu framu (šikmá rovina)	m	+		PM-NC
CUT2DFD	G	2½-D-korekce rádiusu nástroje vztažená na diferenční nástroj, vzhledem k aktuálnímu framu (šikmá rovina)	m	+		PM-NC
CUT3DC	G	3D korekce rádiusu nástroje pro obvodové frézování	m	+		PM-NC
CUT3DCC	G	3D korekce rádiusu nástroje pro obvodové frézování se zohledňováním omezovací plochy s 3D korekcí rádiusu: Kontura na obráběné ploše	m	+		PM-NC
CUT3DCCD	G	3D korekce rádiusu nástroje pro obvodové frézování se zohledňováním omezovací plochy s diferenčním nástrojem na dráze středu nástroje: Přísuv k omezující ploše.	m	+		PM-NC
CUT3DCD	G	3D korekce rádiusu nástroje pro obvodové frézování vztažená na diferenční nástroj	m	+		PM-NC
CUT3DF	G	3D korekce rádiusu nástroje pro frézování na čelní ploše se změnou orientace	m	+		PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
CUT3DFD	G	3D korekce rádiusu nástroje pro frézování na čelní ploše se změnou orientace vztažená na diferenční nástroj	m	+		PM-NC
CUT3DFF	G	3D korekce rádiusu nástroje pro frézování na čelní ploše s konstantní orientací. Orientace nástroje je definována příkazy G17 - G19 a v případě potřeby může být pomocí framu po-otočena do jiného směru.	m	+		PM-NC
CUT3DFS	G	3D korekce rádiusu nástroje pro frézování na čelní ploše s konstantní orientací. Orientace nástroje je definována příkazy G17 - G19 a nemůže být ovlivňována pomocí framů.	m	+		PM-NC
CUTCONOF [6]	G	Deaktivování konstantní korekce rádiusu	m	+		PM-NC
CUTCONON	G	Aktivování konstantní korekce rádiusu	m	+		PM-NC
CUTMOD	A	Aktivování funkce "Modifikace korekčních parametrů u točivých nástrojů“ (ve spojení s orientovatelnými držáky nástrojů)	m	+		PM-NC
CUTMODK	A	Aktivování funkce "Modifikace korekčních parametrů u točivých nástrojů“ (ve spojení s transformacemi orientace, které byly definovány prostřednictvím kinematického řetězce)	m	+		PM-NC
CYCLE60	C (T)	Gravírovací cyklus		+		PM-NC
CYCLE61	C (T)	Rovinné frézování		+		PM-NC
CYCLE62	C (T)	Volání kontury		+		PM-NC
CYCLE63	C (T)	Frézování konturové kapsy		+		PM-NC
CYCLE64	C (T)	Předvrtání konturové kapsy		+		PM-NC
CYCLE70	C (T)	Frézování závitu		+		PM-NC
CYCLE72	C (T)	Frézování po dráze		+		PM-NC
CYCLE76	C (T)	Frézování pravoúhlého čepu		+		PM-NC
CYCLE77	C (T)	Frézování kruhového čepu		+		PM-NC
CYCLE78	C (T)	Frézování vrtaných závitů		+		PM-NC
CYCLE79	C (T)	Mnohohran		+		PM-NC
CYCLE81	C (T)	Vrtání, navrtávání středicích důlků		+		PM-NC
CYCLE82	C (T)	Vrtání, čelní zahlubování		+		PM-NC
CYCLE83	C (T)	Vrtání hlubokých děr		+		PM-NC
CYCLE84	C (T)	Vrtání závitů bez vyrovnávací hlavičky		+		PM-NC
CYCLE85	C (T)	Vystružování		+		PM-NC
CYCLE86	C (T)	Vyvrtávání		+		PM-NC
CYCLE92	C (T)	Upichování		+		PM-NC
CYCLE95	C (T)	Obrábění kontury		+		PM-NC
CYCLE98	C (T)	Řetězec závitů		+		PM-NC
CYCLE99	C (T)	Soustružení závitů		+		PM-NC
CYCLE116	C (M)	Výpočet středu a rádiusu kruhu		+		PM-MC
CYCLE119	C (M)	Stanovení polohy v prostoru		+		PM-MC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
CYCLE150	C (M)	Vypisování/ukládání do protokolu výsledků měření		+		PM-MC
CYCLE435	C (T)	Výpočet polohy orovnávače		+		PM-NC
CYCLE495	C (T)	Profilování		+		PM-NC
CYCLE750	C (A)	Interní pracovní cyklus pro cykly CYCLE751 ... CYCLE759 (obsahuje příkaz MMC pro vlastní volání funkce)		-		FM-A
CYCLE751	C (A)	Zahájení / uskutečnění / ukončení optimalizační procedury		M		FM-A
CYCLE752	C (A)	Připojení osy do optimalizační procedury		M		FM-A
CYCLE753	C (A)	Aktivování režimu optimalizace		M		FM-A
CYCLE754	C (A)	Vložení / odstranění datového bloku		M		FM-A
CYCLE755	C (A)	Uložit / opětovné načtení datového bloku		M		FM-A
CYCLE756	C (A)	Aktivování výsledků optimalizace		M		FM-A
CYCLE757	C (A)	Uložení optimalizačních dat		M		FM-A
CYCLE758	C (A)	Změna hodnoty parametru		M		FM-A
CYCLE759	C (A)	Načtení hodnoty parametru		M		FM-A
CYCLE782	C (T)	Přizpůsobení zatížení		+		PM-NC
CYCLE800	C (T)	Naklápění		+		PM-NC
CYCLE801	C (T)	Mřížka nebo obdélník		+		PM-NC
CYCLE802	C (T)	Libovolné polohy		+		PM-NC
CYCLE806	C (T)	Interpolační soustružení		+		PM-NC
CYCLE830	C (T)	Vrtání hlubokých děr 2		+		PM-NC
CYCLE832	C (T)	Parametry pro vysokorychlostní obrábění		+		PM-NC
CYCLE840	C (T)	Vrtání závitů s vyrovnávací hlavičkou		+		PM-NC
CYCLE899	C (T)	Frézování otevřené drážky		+		PM-NC
CYCLE930	C (T)	Zápich		+		PM-NC
CYCLE940	C (T)	Odlehčovací zápich určitého tvaru		+		PM-NC
CYCLE951	C (T)	Oddělování třísky		+		PM-NC
CYCLE952	C (T)	Zápichy na kontuře		+		PM-NC
CYCLE961	C (M)	Stanovení polohy rohu obrobku (vnitřního nebo vnějšího) a její dosazení jako posunutí počátku.		+		PM-MC
CYCLE971	C (M)	Kalibrace nástrojové měřicí sondy, měření délky a/nebo rádiusu nástroje (jen pro technologii frézování)		+		PM-MC
CYCLE973	C (M)	Kalibrace obrobkové měřicí sondy na ploše na obrobku nebo v drážce (pouze pro technologii soustružení)		+		PM-MC
CYCLE974	C (M)	Stanovení počátku souřadného systému obrobku ve zvolené měřicí ose, stanovení korekce nástroje pomocí měření 1 bodu (pouze pro technologii soustružení)		+		PM-MC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
CYCLE976	C (M)	Uskutečňování kalibrace obrobkové měřicí sondy v kalibračním kruhu, příp. na kalibrační kuličce, zcela v pracovní rovině nebo na hraně pro určitou osu a směr		+		PM-MC
CYCLE977	C (M)	Stanovení středu v rovině, jakož i šířky, příp. průměru		+		PM-MC
CYCLE978	C (M)	Měření polohy hrany v souřadném systému obrobku		+		PM-MC
CYCLE979	C (M)	Stanovení středu v rovině a měření rádiusu kruhového segmentu		+		PM-MC
CYCLE982	C (M)	Kalibrace nástrojové měřicí sondy, měření soustružnických, vrtacích a frézovacích nástrojů (jen pro technologii soustružení)		+		PM-MC
CYCLE994	C (M)	Stanovení počátku souřadného systému obrobku ve zvolené měřicí ose pomocí měření 2 bodů (pouze pro technologii soustružení)		+		PM-MC
CYCLE995	C (M)	Měření úhlové chyby vřetena na obráběcím stroji		+		PM-MC
CYCLE996	C (M)	Stanovení důležitých parametrů pro kinematické transformace s kruhovými osami		+		PM-MC
CYCLE997	C (M)	Určování středu a průměru koule, měření středů tří libovolně umístěných koulí		+		PM-MC
CYCLE998	C (M)	Určování úhlové polohy plochy (roviny) vzhledem k pracovní rovině, stanovení úhlu hran v souřadném systému obrobku		+		PM-MC
CYCLE4071	C (T)	Podélné broušení s přísuvem na bod obratu		+		PM-NC
CYCLE4072	C (T)	Podélné broušení s přísuvem na bod obratu a se signálem pro přerušení		+		PM-NC
CYCLE4073	C (T)	Podélné broušení se spojitým přísuvem		+		PM-NC
CYCLE4074	C (T)	Podélné broušení se spojitým přísuvem a se signálem pro přerušení		+		PM-NC
CYCLE4075	C (T)	Plošné broušení s přísuvem na bod obratu		+		PM-NC
CYCLE4077	C (T)	Plošné broušení s přísuvem na bod obratu a se signálem pro přerušení		+		PM-NC
CYCLE4078	C (T)	Plošné broušení se spojitým přísuvem		+		PM-NC
CYCLE4079	C (T)	Plošné broušení s přerušovaným přísuvem		+		PM-NC
CYCLE9960	C (M)	Kompletní měření kinematiky		+		PM-MC

Příkazy D ... F

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
D	A	Číslo korekčních parametrů nástroje		+		PM-NC
D0	A	Je-li použito D0, jsou korekční parametry daného nástroje deaktivovány.		+		PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1][2][3][4][5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
DAC	K	Absolutní blokové programování průměrů pro specifickou osu	s	+		PM-NC
DC	K	Údaj absolutního rozměru pro kruhové osy, na pozici se najíždí přímo	s	+		PM-NC
DCI	K	Přiřazení datové třídy I (= Individuální) (jen SINUMERIK 828D!)		+		PM-NC
DCM	K	Přiřazení datové třídy M (= Výrobce) (jen SINUMERIK 828D!)		+		PM-NC
DCU	K	Přiřazení datové třídy U (= Uživatel) (jen SINUMERIK 828D!)		+		PM-NC
DEF	K	Definice proměnných		+		PM-NC
DEFAULT	K	Větev v příkazu větvení CASE		+		PM-NC
DEFINE	K	Klíčové slovo pro definice maker		+		PM-NC
DELAYFSTOF	P	Definice konce oblasti zastavení-zpoždění	m	+	-	PM-NC
DELAYFSTON	P	Definice začátku oblasti zastavení-zpoždění	m	+	-	PM-NC
DELDL	F	Vymazání aditivních korekcí		+	-	PM-NC
DELDTG	P	Vymazání zbytkové dráhy		-	+	FM-SA
DELETE	P	Vymazání zadaného souboru. Název souboru může být zadán pomocí cesty a identifikace souboru.		+	-	PM-NC
DELMLOWNER	F	Vymazání místa v zásobníku, které je napevno přiřazeno nástroji		+	-	FM-TM
DELMLRES	F	Vymazání rezervace místa v zásobníku		+	-	FM-TM
DELMT	P	Vymazat modul Multitool		+	-	FM-TM
DELOBJ	F	Mazání prvků z kinematických řetězců, chráněných oblastí, prvků chráněných oblastí, dvojic oblastí chráněných kvůli kolizi a dat transformací		+		PM-NC
DELT	P	Vymazání nástroje		+	-	FM-TM
DELTC	P	Vymazání datového bloku držáku nástroje		+	-	FM-TM
DELTOOLENV	F	Vymazání datových bloků pro popis nástrojového prostředí		+	-	PM-NC
DIACYCOFA	K	Modální programování průměrů pro specifickou osu: V cyklech deaktivováno	m	+		FM-A
DIAM90	G	Programování průměrů pro G90, programování rádiusů pro G91	m	+		PM-NC
DIAM90A	K	Modální programování průměrů pro G90 a AC, programování rádiusů pro G91 a IC pro specifickou osu	m	+		PM-NC
DIAMCHAN	K	Převzetí všech os ze strojních parametrů týkajících se funkcí os v kanálovém stavu programování průměrů		+		PM-NC
DIAMCHANA	K	Převzetí programování průměrů ze specifického kanálu		+		PM-NC
DIAMCYCOF	G	Programování průměrů pro specifický kanál: V cyklech deaktivováno	m	+		FM-A

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1][2][3][4][5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
DIAMOF [6]	G	Programování průměrů: Vypnuto Základní nastavení viz údaje od výrobce stroje	m	+		PM-NC
DIAMOFA	K	Zapnutí modálního programování průměrů pro specifickou osu: Vypnuto Základní nastavení viz údaje od výrobce stroje	m	+		PM-NC
DIAMON	G	Programování průměrů: Zapnuto	m	+		PM-NC
DIAMONA	K	Zapnutí modálního programování průměrů pro specifickou osu: Zapnuto Aktivování viz údaje od výrobce stroje	m	+		PM-NC
DIC	K	Relativní blokové programování průměrů pro specifickou osu	s	+		PM-NC
DILF	A	Návratová dráha (délka)	m	+		PM-NC
DISABLE	P	Vypnutí přerušení		+	-	PM-NC
DISC	A	Převýšení přechodové kružnice – korekce rádiusu nástroje	m	+		PM-NC
DISCL	A	Vzdálenost koncového bodu rychlého přísuvného pohybu od roviny obrábění		+		PM-NC
DISPLOF	PA	Potlačení vypisování aktuálního bloku		+		PM-NC
DISPLON	PA	Odstranění potlačení vypisování aktuálního bloku		+		PM-NC
DISPR	A	Rozdíl dráhy pro zpětné polohování	s	+		PM-NC
DISR	A	Vzdálenost pro zpětné polohování	s	+		PM-NC
DISRP	A	Vzdálenost návratové roviny od roviny opracování při měkkém najíždění a odjíždění		+		PM-NC
DITE	A	Výběr závitu	m	+		PM-NC
DITS	A	Náběh závitu	m	+		PM-NC
DIV	K	Celočíselné dělení		+		PM-NC
DL	A	Aktivování lokálně závislé aditivní korekce nástroje (DL, součtová a seřizovací korekce)	m	+		PM-NC
DO	K	Synchronní akce: Spouštění akcí v případě splněné podmínky		-	+	FM-SA
DRFOF	P	Vypnutí posunutí ručním kolečkem (DRF)	m	+	-	PM-NC
DRIVE	G	Zrychlení po dráze závislé na rychlosti	m	+		PM-NC
DRIVEA	P	Zapnutí lomené charakteristiky zrychlení pro naprogramované osy		+	-	PM-NC
DRVPRD	P	Načtení parametru pohonu		+	-	PM-NC
DRVPWR	P	Zápis do parametru pohonu		+	-	PM-NC
DYNFINISH	G	Dynamika pro obrábění načisto	m	+		PM-NC
DYNNORM [6]	G	Normální dynamika	m	+		PM-NC
DYNPOS	G	Dynamika pro režim polohování, vrtání závitů	m	+		PM-NC
DYNPREC	G	Dynamika pro jemné obrábění načisto	m	+		PM-NC
DYNROUGH	G	Dynamika pro obrábění nahrubo	m	+		PM-NC
DYNSEMIFIN	G	Dynamika pro předběžné obrábění načisto	m	+		PM-NC
DZERO	P	Označení všech D-čísel dané jednotky TO jako neplatné		+	-	PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1][2][3][4][5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
EAUTO	G	Definice posledního splinového úseku prostřednictvím posledních 3 bodů	m	+		PM-NC
EGDEF	P	Definice elektronické převodovky		+	-	PM-NC
EGDEL	P	Vymazání definice vazby pro vlečnou osu		+	-	PM-NC
EGOFC	P	Kontinuální vypínání elektronické převodovky		+	-	PM-NC
EGOFS	P	Selektivní vypnutí elektronické převodovky		+	-	PM-NC
EGON	P	Zapnutí elektronické převodovky		+	-	PM-NC
EGONSYN	P	Zapnutí elektronické převodovky		+	-	PM-NC
EGONSYNE	P	Zapnutí elektronické převodovky, spolu se zadáním režimu najíždění		+	-	PM-NC
ELSE	K	NC program: Větvení programu v případě nesplněné podmínky IF		+	-	PM-NC
ELSE	K	Synchronní akce: Spouštění akcí v případě nesplněné podmínky		-	+	FM-SA
ENABLE	P	Zapnutí přerušení		+	-	PM-NC
ENAT [6]	G	Přirozený přechod na následující blok posuvu	m	+		PM-NC
ENDFOR	K	Koncový řádek smyčky FOR se zadaným počtem průchodů		+		PM-NC
ENDIF	K	Koncový řádek větvení IF		+		PM-NC
ENDLABEL	K	Koncové návěští pro opakování výrobního programu pomocí příkazu REPEAT		+		PM-NC, FM-B
ENDLOOP	K	Koncový řádek nekonečné programové smyčky LOOP		+		PM-NC
ENDPROC	K	Koncový řádek programu, na jehož prvním řádku je příkaz PROC.		+		
ENDWHILE	K	Koncový řádek smyčky WHILE		+		PM-NC
ESRR	P	Nastavení parametrů pro odjíždění nezávislé na pohonu (ESR)		+		PM-NC
ESRS	P	Nastavení parametrů pro zastavení nezávislé na pohonu (ESR)		+		PM-NC
ETAN	G	Tangenciální křivkový přechod na následující blok posuvu na začátku splinu	m	+		PM-NC
EVERY	K	Synchronní akce se uskuteční v případě změny stavu podmínky z FALSE na TRUE		-	+	FM-SA
EX	K	Klíčové slovo pro přiřazení hodnoty v exponenciálním způsobu zápisu		+		PM-NC
EXECSTRING	P	Předání proměnné typu String se zpracovatelným řádkem výrobního programu		+	-	PM-NC
EXECTAB	P	Spuštění zpracování prvku z tabulky pohybů		+	-	PM-NC
EXECUTE	P	Zapnutí zpracovávání programu		+	-	PM-NC
EXP	F	Exponenciální funkce ex		+	+	PM-NC
EXTCALL	A	Zpracovávání externího podprogramu		+	+	PM-NC
EXTCLOSE	P	Zavření externího přístroje/souboru otevřeného pro zápis		+	-	PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1][2][3][4][5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
EXTERN	K	Identifikace podprogramu s předáváním parametrů		+		PM-NC
EXTOPEN	P	Otevření externího přístroje/souboru pro daný kanál pro zápis		+	-	PM-NC
F	A	Hodnota posuvu (ve spojení s G4 se pomocí F programuje také doba prodlevy)		+	+	PM-NC
FA	K	Posuv osy	m	+	+	PM-NC
FAD	A	Posuv pro přísuv při měkkém najíždění a odjíždění		+		PM-NC
FALSE	K	Logická konstanta: FALSE		+	+	PM-NC
FB	A	Blokový posuv		+		PM-NC
FCTDEF	P	Definice polynomické funkce		+	-	PM-NC
FCUB	G	Posuv proměnný podle kubického splinu	m	+		PM-NC
FD	A	Posuv po dráze pro korekci pomocí ručního kolečka	s	+		PM-NC
FDA	K	Posuv osy při korekci ručním kolečkem	s	+		PM-NC
FENDNORM [6]	G	Vypnutí zpožďování v rozích	m	+		PM-NC
FFWOF [6]	G	Vypnutí dopředné regulace	m	+		PM-NC
FFWON	G	Zapnutí dopředné regulace	m	+		PM-NC
FGREF	K	Vztažný rádius u kruhových os nebo vztažný faktor dráhy u orientovaných os (vektorová interpolace)	m	+		PM-NC
FGROUP	P	Definice os s posuvem po dráze		+	-	PM-NC
FI	K	Parametr pro přístup k datům framu: Jemné posunutí		+		PM-NC
FIFOCTRL	G	Ovládání zásobníku dopředné regulace	m	+		PM-NC
FILEDATE	P	Zjištění data posledního přístupu za účelem zápisu do daného souboru		+	-	PM-NC
FILEINFO	P	Zjištění parametrů FILEDATE, FILESIZE, FILESTAT a FILETIME najednou		+	-	PM-NC
FILESIZE	P	Zjištění aktuální velikosti souboru		+	-	PM-NC
FILESTAT	P	Zjištění stavových informací o souboru, jako jsou oprávnění pro čtení, zápis, spouštění, vypisování a mazání (rwxsd)		+	-	PM-NC
FILETIME	P	Zjištění času posledního přístupu za účelem zápisu do daného souboru		+	-	PM-NC
FINEA	K	Konec pohybu při dosažení jemného okna přesného najetí	m	+		PM-NC
FL	K	Mezní hodnota rychlosti pro synchronní osy	m	+		PM-NC
FLIM	A	Přizpůsobení maximální rychlosti pohybu po dráze	m	+	-	PM-NC
FLIN	G	Lineárně proměnný posuv	m	+		PM-NC
FMA	K	Větší počet axiálních posuvů	m	+		PM-NC
FNORM [6]	G	Normální posuv podle DIN 66025	m	+		PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1][2][3][4][5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
FOC	K	Omezení momentu/síly s blokovou platností	s	-	+	FM-SA
FOCOF	K	Deaktivování modálního omezení momentu/síly	m	-	+	FM-SA
FOCON	K	Aktivování modálního omezení momentu/síly	m	-	+	FM-SA
FOR	K	Smyčka s pevně daným počtem průchodů		+		PM-NC
FP	A	Pevný bod: Číslo pevného bodu, na který se má najíždět	s	+		PM-NC
FPO	K	Průběh posuvu naprogramovaný pomocí polynomu		+		PM-NC
FPR	P	Označení kruhové osy		+	-	PM-NC
FPRAOF	P	Deaktivování otáčkového posuvu		+	-	PM-NC
FPRAON	P	Aktivování otáčkového posuvu		+	-	PM-NC
FRAME	K	Datový typ určený pro definici souřadných systémů		+		PM-NC
FRC	A	Posuv pro rádius a fasetu	s	+		PM-NC
FRCM	A	Modální posuv pro rádius a fasetu	m	+		PM-NC
FROM	K	Akce se uskuteční, jestliže je daná podmínka jednorázově splněna a pokud je aktivní synchronní akce.		-	+	FM-SA
FTOC	P	Změna jemné korekce nástroje		-	+	FM-SA
FTOCOF [6]	G	Vypnutí on-line působící jemné korekce nástroje	m	+		PM-NC
FTOCON	G	Zapnutí on-line působící jemné korekce nástroje	m	+		PM-NC
FXS	K	Najíždění na pevný doraz	m	+	+	PM-NC
FXST	K	Mezní hodnota momentu pro najíždění na pevný doraz	m	+	+	PM-NC
FXSW	K	Monitorovací okno pro najíždění na pevný doraz		+	+	PM-NC
FZ	K	Posuv na zub	m	+		PM-NC

Příkazy G ... L

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
G0	G	Lineární interpolace rychlým posuvem (pohyb rychlým posuvem)	m	+		PM-NC
G1 [6]	G	Lineární interpolace s pracovním posuvem (přímková interpolace)	m	+		PM-NC
G2	G	Kruhová interpolace ve směru hodinových ručiček	m	+		PM-NC
G3	G	Kruhová interpolace proti směru hodinových ručiček	m	+		PM-NC
G4	G	Doba prodlevy určená časově	s	+		PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
G5	G	Šikmé zapichovací broušení	s	+		PM-NC
G7	G	Vyrovnávací pohyb při šikmém zapichovacím broušení	s	+		PM-NC
G9	G	Přesné najetí – snižování rychlosti	s	+		PM-NC
G17 [6]	G	Volba pracovní roviny X/Y	m	+		PM-NC
G18	G	Volba pracovní roviny Z/X	m	+		PM-NC
G19	G	Volba pracovní roviny Y/Z	m	+		PM-NC
G25	G	Dolní ohraničení pracovního pole	s	+		PM-NC
G26	G	Horní ohraničení pracovního pole	s	+		PM-NC
G33	G	Řezání závitů s konstantním stoupáním	m	+		PM-NC
G34	G	Řezání závitu s lineárně narůstajícím stoupáním	m	+		PM-NC
G35	G	Řezání závitu s lineárně klesajícím stoupáním	m	+		PM-NC
G40 [6]	G	Vypnutí korekce rádiusu nástroje	m	+		PM-NC
G41	G	Korekce rádiusu nástroje vlevo od kontury	m	+		PM-NC
G42	G	Korekce rádiusu nástroje vpravo od kontury	m	+		PM-NC
G53	G	Potlačení aktuálního posunutí počátku (blokové)	s	+		PM-NC
G54	G	1. nastavitelné posunutí počátku	m	+		PM-NC
G55	G	2. nastavitelné posunutí počátku	m	+		PM-NC
G56	G	3. nastavitelné posunutí počátku	m	+		PM-NC
G57	G	4. nastavitelné posunutí počátku	m	+		PM-NC
G58 (840D sl)	G	Absolutní programovatelné posunutí počátku (hrubé posunutí)	s	+		PM-NC
G58 (828D)	G	5. nastavitelné posunutí počátku	m	+		PM-NC
G59 (840D sl)	G	Aditivní programovatelné posunutí počátku (jemné posunutí)	s	+		PM-NC
G59 (828D)	G	6. nastavitelné posunutí počátku	m	+		PM-NC
G60 [6]	G	Přesné najetí – snižování rychlosti	m	+		PM-NC
G62	G	Zpoždění na vnitřních rozích při aktivní korekci rádiusu nástroje (G41, G42)	m	+		PM-NC
G63	G	Vrtání závitů s vyrovnávací hlavičkou	s	+		PM-NC
G64	G	Režim řízení pohybu po dráze	m	+		PM-NC
G70	G	Rozměrové údaje pro geometrické parametry v palcích (délky)	m	+	+	PM-NC
G71 [6]	G	Rozměrové údaje pro geometrické parametry v metrických jednotkách (délky)	m	+	+	PM-NC
G74	G	Najíždění na referenční bod	s	+		PM-NC
G75	G	Najíždění na pevný bod	s	+		PM-NC
G90 [6]	G	Zadávání absolutních rozměrů	m/s	+		PM-NC
G91	G	Zadávání inkrementálních rozměrů	m/s	+		PM-NC
G93	G	Časově reciproční posuv v jednotkách 1/min	m	+		PM-NC
G94 [6]	G	Lineární posuv F v mm/min, v palcích/min nebo stupních/min	m	+		PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
G95	G	Otáčkový posuv F v mm/ot nebo palcích/ot	m	+		PM-NC
G96	G	Otáčkový posuv (stejně jako u G95) a konstantní řezná rychlost)	m	+		PM-NC
G97	G	Otáčkový posuv a konstantní otáčky vřetena (konstantní řezná rychlost je vypnuta)	m	+		PM-NC
G110	G	Programování pólu vztažené na naposled naprogramovanou požadovanou pozici	s	+		PM-NC
G111	G	Programování pólu vzhledem k počátku aktuální souřadné soustavy obrobku	s	+		PM-NC
G112	G	Programování pólu vzhledem k poslednímu platnému pólu	s	+		PM-NC
G140 [6]	G	Směr najíždění WAB definován příkazy G41/G42	m	+		PM-NC
G141	G	Směr najíždění WAB vlevo od kontury	m	+		PM-NC
G142	G	Směr najíždění WAB vpravo od kontury	m	+		PM-NC
G143	G	Směr najíždění WAB v závislosti na tečně	m	+		PM-NC
G147	G	Měkké najíždění po přímce	s	+		PM-NC
G148	G	Měkké odjíždění po přímce	s	+		PM-NC
G153	G	Potlačení aktuálního framu včetně základního framu	s	+		PM-NC
G247	G	Měkké najíždění po čtvrtkruhu	s	+		PM-NC
G248	G	Měkké odjíždění po čtvrtkruhu	s	+		PM-NC
G290 [6]	G	Aktivování přepnutí na režim SINUMERIK	m	+		FM-TM
G291	G	Aktivování přepnutí na režim ISO2/3	m	+		FM-TM
G331	G	Vrtání závitů bez vyrovnávací hlavičky, kladné stoupání, směr otáčení vpravo	m	+		PM-NC
G332	G	Vrtání závitů bez vyrovnávací hlavičky, záporné stoupání, směr otáčení vlevo	m	+		PM-NC
G335	G	Soustružení sudovitého závitu ve směru hodinových ručiček	m	+		PM-NC
G336	G	Soustružení sudovitého závitu proti směru hodinových ručiček	m	+		PM-NC
G340 [6]	G	Prostorový najížděcí blok (hloubka a v rovině stejné – spirála)	m	+		PM-NC
G341	G	Napřed přísuv v kolmé ose (z), pak najíždění v rovině	m	+		PM-NC
G347	G	Měkké najíždění po půlkruhu	s	+		PM-NC
G348	G	Měkké odjíždění po půlkruhu	s	+		PM-NC
G450 [6]	G	Přechodový kruh	m	+		PM-NC
G451	G	Průsečík ekvidistantních drah	m	+		PM-NC
G460 [6]	G	Aktivování protikolizního monitorování pro blok najíždění a odjíždění	m	+		PM-NC
G461	G	Vložení kruhu do bloku korekce rádiusu nástroje	m	+		PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
G462	G	Vložení přímky do bloku korekce rádiusu nástroje	m	+		PM-NC
G500 [6]	G	Deaktivování všech nastavitelných framů, základní frame je aktivní	m	+		PM-NC
G505 ... G599	G	5. ... 99. Nastavitelné posunutí počátku	m	+		PM-NC
G601 [6]	G	Přechod na další blok při jemném přesném najetí	m	+		PM-NC
G602	G	Přechod na další blok při hrubém přesném najetí	m	+		PM-NC
G603	G	Přechod na další blok při konci bloku IPO	m	+		PM-NC
G621	G	Snížení rychlosti na všech rozích	m	+		PM-NC
G641	G	Režim řízení pohybu po dráze s přechodovými zaobleními podle kritéria dráhy (= programovatelná vzdálenost zaoblení)	m	+		PM-NC
G642	G	Režim řízení pohybu po dráze s přechodovými zaobleními při dodržení definovaných tolerancí	m	+		PM-NC
G643	G	Režim řízení pohybu po dráze s přechodovými zaobleními při dodržení definovaných tolerancí (uvnitř bloku)	m	+		PM-NC
G644	G	Režim řízení pohybu po dráze s přechodovými zaobleními s maximální možnou dynamikou	m	+		PM-NC
G645	G	Režim řízení pohybu po dráze s přechodovými zaobleními v rozích a s tangenciálními přechody mezi bloky při dodržení definovaných tolerancí	m	+		PM-NC
G700	G	Rozměrové údaje pro geometrické a technologické parametry v palcích (délky, posuv)	m	+	+	PM-NC
G710 [6]	G	Rozměrové údaje pro geometrické a technologické parametry v metrických jednotkách (délky, posuv)	m	+	+	PM-NC
G810 [6], ..., G819	G	G-skupina vyhrazená pro uživatele OEM		+		PM-NC
G820 [6], ..., G829	G	G-skupina vyhrazená pro uživatele OEM		+		PM-NC
G931	G	Zadání posuvu pomocí doby posuvu, konstantní rychlost po dráze vypnuta	m	+		
G942	G	Zmrazení lineárního posuvu a konstantní řezné rychlosti nebo otáček vřetena	m	+		
G952	G	Zmrazení otáčkového posuvu a konstantní řezné rychlosti nebo otáček vřetena	m	+		
G961	G	Lineární posuv (stejně jako u G94) a konstantní řezná rychlost	m	+		PM-NC
G962	G	Lineární posuv nebo otáčkový posuv a konstantní řezná rychlost	m	+		PM-NC
G971	G	Lineární posuv a konstantní otáčky vřetena (konstantní řezná rychlost je vypnuta)	m	+		PM-NC
G972	G	Lineární posuv nebo otáčkový posuv a konstantní otáčky vřetena (konstantní řezná rychlost je vypnuta)	m	+		PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
G973	G	Otáčkový posuv bez omezení otáček vřetena a konstantní otáčky vřetena (G97 bez LIMS pro režim ISO)	m	+		PM-NC
GEOAX	P	Nové přiřazení geometrických os 1 - 3 kanálovým osám		+	-	PM-NC
GET	P	Výměna uvolněné osy mezi kanály		+	+	PM-NC
GETACTT	F	Stanovení aktivního nástroje ze skupiny stejnojmenných nástrojů.		+	-	FM-TM
GETACTTD	F	Stanovení odpovídajícího T-čísla k absolutnímu D-číslu		+	-	PM-NC
GETD	P	Výměna uvolněné osy mezi kanály		+	-	PM-NC
GETDNO	F	Zjištění D-čísla břitu (CE) daného nástroje (T)		+	-	PM-NC
GETEXET	P	Načtení T-čísla vyměňovaného nástroje		+	-	FM-TM
GETFREELOC	P	Vyhledání volného místa v zásobníku pro zadaný nástroj		+	-	FM-TM
GETSELT	P	Zjištění předem zvoleného T-čísla		+	-	FM-TM
GETT	F	Přiřazení T-čísla názvu nástroje		+	-	FM-TM
GETTCOR	F	Načtení délek nástroje, příp. složek délky nástroje		+	-	PM-NC
GETTENV	F	Načtení T-čísla, D-čísla a DL-čísla		+	-	PM-NC
GETVARAP	F	Načtení přístupových oprávnění k systémové/uživatelské proměnné		+	-	PM-NC
GETVARDFT	F	Načtení standardní hodnoty systémové/uživatelské proměnné		+	-	PM-NC
GETVARLIM	F	Načtení mezní hodnoty systémové/uživatelské proměnné		+	-	PM-NC
GETVARPHU	F	Načtení fyzikální jednotky systémové/uživatelské proměnné		+	-	PM-NC
GETVARTYP	F	Načtení datového typu systémové/uživatelské proměnné		+	-	PM-NC
GFRAME0 ... GFRAME100	G	Aktivování framu pro broušení <n> ve správě dat v kanálu	m	+		PM-NC
GOTO	K	Příkaz skoku napřed dopředu a potom zpátky (napřed ve směru konce programu a potom směrem k začátku programu)		+		PM-NC
GOTOB	K	Příkaz skoku směrem dozadu (směrem k začátku programu)		+		PM-NC
GOTOC	K	Stejné jako GOTO, ale s potlačením alarmu 14080 „Cíl skoku nenalezen“.		+		PM-NC
GOTOF	K	Příkaz skoku směrem dopředu (směrem ke konci programu)		+		PM-NC
GOTOS	K	Skok zpátky na začátek programu		+		PM-NC
GP	K	Klíčové slovo pro nepřímé programování atributů polohy		+		PM-NC
GROUP_ ADDEND	C (T)	Konec doplňkového bloku najíždění		+		PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
GROUP_BEGIN	C (T)	Začátek programového bloku		+		PM-NC
GROUP_END	C (T)	Konec programového bloku		+		PM-NC
GWPSOF	P	Deaktivování konstantní obvodové rychlosti brusného kotouče	s	+	-	PM-NC
GWPSON	P	Aktivování konstantní obvodové rychlosti brusného kotouče	s	+	-	PM-NC
H...	A	Výstup pomocných funkcí do PLC		+	+	PM-NC, FM-B
HOLES1	C (T)	Řada děr		+		PM-NC
HOLES2	C (T)	Díry na kruhovém oblouku		+		PM-NC
I	A	Interpolační parametr	s	+		PM-NC
I1	A	Souřadnice vnitřního bodu	s	+		PM-NC
IC	K	Zadávání inkrementálních rozměrů	s	+		PM-NC
ICYCOF	P	Všechny bloky technologického cyklu za příkazem ICYCOF zpracovávat v taktu IPO		+	+	FM-SA
ICYCON	P	Každý blok technologického cyklu za příkazem ICYCON zpracovávat v samostatném taktu IPO		+	+	FM-SA
ID	K	Identifikace pro modální synchronní akce	m	-	+	FM-SA
IDS	K	Identifikace pro modální statické synchronní akce		-	+	FM-SA
IF	K	Úvodní příkaz pro podmíněné skoky ve výrobním programu / technologickém cyklu		+	+	PM-NC
INDEX	F	Stanovení indexu znaku ve vstupním řetězci		+	-	PM-NC
INICF	K	Inicializace proměnných při aktivování NEWCONF		+		PM-NC
INIPO	K	Inicializace proměnných při zapnutí systému (Power-On)		+		PM-NC
INIRE	K	Inicializace proměnných při resetu		+		PM-NC
INIT	P	Aktivování určitého NC programu pro zpracovávání v určitém kanálu.		+	-	PM-NC
INITIAL		Vytvoření souboru INI pro všechny oblasti		+		PM-NC
INT	K	Datový typ: Celočíselné hodnoty se znaménkem		+		PM-NC
INTERSEC	F	Výpočet průsečíku mezi dvěma konturovými prvky		+	-	PM-NC
INVCCW	G	Pohyb po evolventě, proti směru hodinových ručiček	m	+		PM-NC
INVCW	G	Pohyb po evolventě, ve směru hodinových ručiček	m	+		PM-NC
INVFRAME	F	Výpočet inverzního framu z daného framu		+	-	FM-B
IP	K	Proměnný interpolační parametr		+		PM-NC
IPOBRKA	P	Kritérium pro řízení pohybu od počátečního bodu skokové změny brzdné charakteristiky	m	+	+	
IPOENDA	K	Konec pohybu při dosažení "IPO Stop"	m	+		PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
IPTRLOCK	P	Napevno nastavit začátek úseku programu, který může být prohledáván, na následující blok s funkcemi stroje	m	+	-	PM-NC
IPTRUNLOCK	P	Nastavení konce úseku programu, který může být prohledáván, na aktuální blok v okamžiku přerušení	m	+	-	PM-NC
IR	A	Souřadnice středu kruhu (směr X) při soustružení sudovitého závitu		+		PM-NC
ISAXIS	F	Kontrola, zda geometrická osa 1 zadaná jako parametr existuje		+	-	PM-NC
ISD	A	Hloubka zajíždění nástroje	m	+		PM-NC
ISFILE	F	Kontrola, zda je daný soubor v uživatelské paměti NC k dispozici		+	-	PM-NC
ISNUMBER	F	Kontrola, zda je možno převést vstupní řetězec na číslo.		+	-	PM-NC
ISOCALL	K	Nepřímé volání programu naprogramovaného v jazyce ISO		+		PM-NC
ISVAR	F	Kontrola, zda předávaný parametr obsahuje proměnnou, která je NC systému známá.		+	-	PM-NC
J	A	Interpolační parametr	s	+		PM-NC
J1	A	Souřadnice vnitřního bodu	s	+		PM-NC
JERKA	P	Aktivování chování při zrychlení, které je nastaveno prostřednictvím MD, pro naprogramované osy		+	-	
JERKLIM	K	Přizpůsobení maximálního osového ryvu	m	+		PM-NC
JERKLIMA	K	Snížení nebo zvýšení maximálního osového ryvu vlečné osy	m	+	+	PM-NC
JR	A	Souřadnice středu kruhu (směr Y) při soustružení sudovitého závitu		+		PM-NC
K	A	Interpolační parametr	s	+		PM-NC
K1	A	Souřadnice vnitřního bodu	s	+		PM-NC
KONT	G	Objíždění kontury s korekcí nástroje	m	+		PM-NC
KONTC	G	Najíždění/odjíždění se spojitým polynomickým zakřivením	m	+		PM-NC
KONTT	G	Najíždění/odjíždění se spojitým tangenciálním polynomem	m	+		PM-NC
KR	A	Souřadnice středu kruhu (směr Z) při soustružení sudovitého závitu		+		PM-NC
L	A	Číslo podprogramu	s	+	+	PM-NC
LEAD	A	Předozadní úhel 1. orientace nástroje 2. polynom orientace	m	+		PM-NC
LEADOF	P	Deaktivování osové vazby pomocí řídící hodnoty		+	+	PM-NC
LEADON	P	Aktivování osové vazby pomocí řídící hodnoty		+	+	PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
LENTOAX	F	Vyvolání informací o přiřazení délek nástroje L1, L2 a L3 aktivního nástroje abscise, ordinátě a aplikátě.		+	-	PM-NC
LFOF [6]	G	Deaktivování rychlého zpětného pohybu při řezání závitů	m	+		PM-NC
LFON	G	Aktivování rychlého zpětného pohybu při řezání závitů	m	+		PM-NC
LFPOS	G	Návrat osy stanovené příkazem POLFMASK nebo POLFLIN na absolutní pozici naprogramovanou pomocí příkazu POLF.	m	+		PM-NC
LFTXT [6]	G	Rovina zpětného pohybu při rychlém pozvednutí nástroje se bude určovat na základě tečny k dráze a momentálního směru nástroje.	m	+		PM-NC
LFWP	G	Rovina zpětného pohybu při rychlém pozvednutí nástroje je určena prostřednictvím aktuální pracovní roviny (G17/G18/G19).	m	+		PM-NC
LIFTFAST	K	Rychlé pozvednutí		+		PM-NC
LIMS	K	Omezení otáček u příkazů G96/G961 a G97	m	+		PM-NC
LLI	K	Spodní mezní hodnota pro proměnné		+		PM-NC
LN	F	Přirozený logaritmus		+	+	PM-NC
LOCK	P	Zablokování synchronní akce s identifikací ID (zastavení technologického cyklu)		-	+	FM-SA
LONGHOLE	C (T)	Podlouhlá díra		+		PM-NC
LOOP	K	Úvodní příkaz nekonečné smyčky		+		PM-NC

Příkazy M ... R

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
M0		Programovatelné zastavení		+	+	PM-NC
M1		Volitelné zastavení		+	+	PM-NC
M2		Konec hlavního programu (jako v případě M30)		+	+	PM-NC
M3		Vřeteno se otáčí vpravo		+	+	PM-NC
M4		Vřeteno se otáčí vlevo		+	+	PM-NC
M5		zastavení vřetena		+	+	PM-NC
M6		Výměna nástroje		+	+	PM-NC
M17		Konec podprogramu		+	+	PM-NC
M19		Nastavení pozice vřetena na polohu zadanou v SD 43240		+	+	PM-NC
M30		Konec hlavního programu (jako v případě M2)		+	+	PM-NC
M40		Automatické přepínání stupňů převodovky		+	+	PM-NC
M41 ... M45		Stupně převodovky 1 ... 5		+	+	PM-NC
M70		Přechod do osového režimu		+	+	PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
MASLDEF	P	Definice spojení os typu Master/Slave		+	+	PM-NC
MASLDEL	P	Zrušení spojení os typu Master/Slave a vymazání definice tohoto spojení		+	+	PM-NC
MASLOF	P	Deaktivování dočasného spojení		+	+	PM-NC
MASLOFS	P	Deaktivování dočasného spojení s automatickým zastavením osy typu Slave		+	+	PM-NC
MASLON	P	Aktivování dočasného spojení		+	+	PM-NC
MATCH	F	Vyhledávání řetězce v rámci jiného řetězce		+	-	PM-NC
MAXVAL	F	Větší z hodnot ve dvou proměnných (aritmetická funkce)		+	+	PM-NC
MCALL	K	Modální volání podprogramu		+		PM-NC
MEAC	K	Kontinuální měření jednou osou bez vymazání zbytkové dráhy	s	+	+	PM-NC
MEAFRAME	F	Výpočet framu na základě změřených bodů		+	-	PM-NC
MEAS	A	Měření s vymazáním zbytkové dráhy	s	+		PM-NC
MEASA	K	Měření jednou osou s vymazáním zbytkové dráhy	s	+	+	PM-NC
MEASURE	F	Metoda výpočtu pro měření obrobku a nástroje		+	-	FM-TE
MEAW	A	Měření bez mazání zbytkové dráhy	s	+		PM-NC
MEAWA	K	Měření jednou osou bez vymazání zbytkové dráhy	s	+	+	PM-NC
MI	K	Přístup k datům framu: Zrcadlové převrácení		+		PM-NC
MINDEX	F	Stanovení indexu znaku ve vstupním řetězci		+	-	PM-NC
MINVAL	F	Menší z hodnot ve dvou proměnných (aritmetická funkce)		+	+	PM-NC
MIRROR	G	Programovatelné zrcadlové převrácení	s	+		PM-NC
MMC	P	Vyvolání interaktivního dialogového okna na HMI z výrobního programu		+	-	PM-NC
MOD	K	Dělení MODULO		+		PM-NC
MODAXVAL	F	Zjištění pozice modulo kruhové osy modulo		+	-	PM-NC
MOV	K	Spuštění polohovací osy		-	+	FM-SA
MOVT	A	Zadání koncového bodu pohybu pracovním posuvem ve směru nástroje				FM-B
MSG	P	Programovatelná hlášení	m	+	-	PM-NC
MVTOOL	P	Příkaz jazyka pro pohyb nástroje		+	-	FM-TM
N	A	Číslo vedlejšího bloku v NC programu		+		PM-NC
NAMETOINT	F	Zjištění indexu systémové proměnné		+		PM-NC
NC	K	Specifikace rozsahu platnosti dat		+		PM-NC
NEWCONF	P	Převzetí změněných strojních parametrů (odpovídá příkazu "Nastavit strojní parametr jako platný")		+	-	PM-NC
NEWMT	F	Založení nového modulu Multitool		+	-	FM-TM
NEWT	F	Založení nového nástroje		+	-	FM-TM

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
NORM [6]	G	Normální nastavení v počátečním a koncovém bodě při korekci nástroje	m	+		PM-NC
NOT	K	Logické NE (negace)		+		PM-NC
NPROT	P	Zapnutí/vypnutí chráněné oblasti pro specifický stroj		+	-	PM-NC
NPROTDEF	P	Definice chráněné oblasti pro specifický stroj		+	-	PM-NC
NUMBER	F	Převedení vstupního řetězce na číslo		+	-	PM-NC
OEMIPO1	G	Interpolace OEM 1	m	+		PM-NC
OEMIPO2	G	Interpolace OEM 2	m	+		PM-NC
OF	K	Klíčové slovo v příkazu větvení CASE		+		PM-NC
OFFN	A	Přídavek rozměru pro naprogramovanou konturu	m	+		PM-NC
OMA1	A	Adresa OEM 1	m	+		PM-NC
OMA2	A	Adresa OEM 2	m	+		PM-NC
OMA3	A	Adresa OEM 3	m	+		PM-NC
OMA4	A	Adresa OEM 4	m	+		PM-NC
OMA5	A	Adresa OEM 5	m	+		PM-NC
OR	K	Logický operátor, spojení typu NEBO		+		PM-NC
ORIAXES	G	Lineární interpolace os stroje nebo orientačních os	m	+		PM-NC
ORIAXPOS	G	Úhel orientace pomocí virtuálních orientačních os s polohováním kruhové osy	m	+		PM-NC
ORIC [6]	G	Změny orientace na vnějších rozích jsou superponovány s vkládaným kruhovým blokem	m	+		PM-NC
ORICONCCW	G	Interpolace po ploše pláště kužele proti směru hodinových ručiček.	m	+		PM-NC, FM-TR
ORICONCW	G	Interpolace po ploše pláště kužele ve směru hodinových ručiček.	m	+		PM-NC, FM-TR
ORICONIO	G	Interpolace po ploše pláště kužele s udáním pomocné meziorientace	m	+		PM-NC, FM-TR
ORICONTO	G	Interpolace po ploše pláště kuželu s tangenciálním přechodem (zadání koncové orientace)	m	+		PM-NC, FM-TR
ORICURVE	G	Interpolace orientace s udáním pohybu dvou kontaktních bodů nástroje	m	+		PM-NC, FM-TR
ORID	G	Změna orientace se bude provádět před kruhovým blokem.	m	+		PM-NC
ORIEULER [6]	G	Úhel orientace pomocí Eulerova úhlu	m	+		PM-NC
ORIMKS	G	Orientace nástroje v souřadném systému stroje	m	+		PM-NC
ORIPATH	G	Orientace nástroje vztažená na dráhu	m	+		PM-NC
ORIPATHS	G	Orientace nástroje vztažená na dráhu, zlom v průběhu orientace se vyhladí	m	+		PM-NC
ORIPLANE	G	Interpolace v rovině (odpovídá ORIVECT) Interpolace pomocí největší kružnice koule	m	+		PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
ORIRESET	P	Základní poloha orientace nástroje s až 3 orientačními osami		+	-	PM-NC
ORIROTA [6]	G	Úhel otočení vztažen ke směru otáčení zadanému absolutně	m	+		PM-NC
ORIROTC	G	Tangenciální vektor otočení k tečně dráhy	m	+		PM-NC
ORIROTR	G	Úhel otočení relativně vůči rovině mezi počáteční a koncovou orientací	m	+		PM-NC
ORIROTT	G	Úhel otočení relativně vůči změně vektoru orientace	m	+		PM-NC
ORIRPY	G	Úhel orientace prostřednictvím úhlu RPY (XYZ)	m	+		PM-NC
ORIRPY2	G	Úhel orientace pomocí úhlu RPY (ZYX)	m	+		PM-NC
ORIS	A	Změna orientace	m	+		PM-NC
ORISOF [6]	G	Vypnutí vyhlazování charakteristiky orientace	m	+		PM-NC
ORISOLH	F	Výpočet orientace		+		PM-NC
ORISON	G	Zapnutí vyhlazování charakteristiky orientace	m	+		PM-NC
ORIVECT [6]	G	Interpolace pomocí největší kružnice koule (identická s ORIPLANE)	m	+		PM-NC
ORIVIRT1	G	Úhel orientace pomocí virtuálních orientačních os (definice 1)	m	+		PM-NC
ORIVIRT2	G	Úhel orientace pomocí virtuálních orientačních os (definice 1)	m	+		PM-NC
ORIWKS [6]	G	Orientace nástroje v souřadném systému obrobku	m	+		PM-NC
OS	K	Oscilační pohyb ZAP/VYP		+		PM-NC
OSB	K	Oscilace: Počáteční bod	m	+		PM-NC
OSC	G	Konstantní vyhlazení orientace nástroje	m	+		PM-NC
OSCILL	K	osa: 1 - 3 přísuvná osa	m	+		PM-NC
OSCTRL	K	Možnosti oscilačního pohybu	m	+		PM-NC
OSD	G	Vyhlazování orientace nástroje zadáním délky zaoblení pomocí parametru SD	m	+		PM-NC
OSE	K	Koncový bod oscilačního pohybu	m	+		PM-NC
OSNSC	K	Oscilace: Počet vyjiskření	m	+		PM-NC
OSOF [6]	G	Vypnutí vyhlazování orientace nástroje	m	+		PM-NC
OSP1	K	Oscilace: levý bod obratu	m	+		PM-NC
OSP2	K	Oscilace: pravý bod obratu	m	+		PM-NC
OSS	G	Vyhlazení orientace nástroje na konci bloku	m	+		PM-NC
OSSE	G	Vyhlazení orientace nástroje na počátku a na konci bloku	m	+		PM-NC
OST	G	Vyhlazování orientace nástroje zadáním úhlové tolerance ve stupních pomocí parametru SD (maximální odchylka od naprogramovaného průběhu orientace)	m	+		PM-NC
OST1	K	Oscilace: stop v levém bodu obratu	m	+		PM-NC
OST2	K	Oscilace: stop v pravém bodu obratu	m	+		PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
OTOL	A	Tolerance orientace pro funkce kompresoru, vyhlazení orientace a druhy zaoblování kontu-rových přechodů	m	+	-	PM-NC
OTOLG0	A	Tolerance orientace při rychlém posuvu	m	+	-	PM-NC
OVR	K	Korekce otáček	m	+		PM-NC
OVRA	K	Korekce otáček pro osu	m	+	+	PM-NC
OVRRAP	K	Korekce rychlého posuvu	m	+		PM-NC
P	A	Počet průchodů podprogramem		+		PM-NC
PACCLIM	A	Přizpůsobení maximálního zrychlení při pohy-bu po dráze	m	+	-	PM-NC
PAROT	G	Srovnání souřadného systému obrobku podle obrobku	m	+		PM-NC
PAROTOF [6]	G	Vypnutí otáčení framu vztahující se na obrobek	m	+		PM-NC
PCALL	K	Volání podprogramu s absolutním udáním ce-sty a předáváním parametrů		+		PM-NC
PDELAYOF	G	Deaktivování zpoždění při lisování	m	+		PM-NC
PDELAYON [6]	G	Aktivování zpoždění při lisování	m	+		PM-NC
PHI	K	Úhel otočení orientace okolo směrové osy ku-žele		+		PM-NC
PHU	K	Fyzikální jednotka proměnné		+		PM-NC
PL	A	1. B-Spline: Vzdálenost uzlů 2. Polynomická interpolace: Délka intervalu parametru při polynomické interpolaci	s	+		PM-NC
PM	K	za minutu		+		PM-NC
PO	K	Koeficient polynomu při polynomické interpo-laci	s	+		PM-NC
POCKET3	C (T)	Frézování pravoúhlé kapsy		+		PM-NC
POCKET4	C (T)	Frézování kruhové kapsy		+		PM-NC
POLF	K	Poloha pro zpětný pohyb LIFTFAST	m	+		PM-NC
POLFA	P	Poloha pro zpětný pohyb jednotlivými osami se spouštěním pomocí parametru $AA_ESR_TRIGGER	m	+	+	PM-NC
POLFMASK	P	Uvolnění os pro zpětný pohyb bez souvislostí mezi jednotlivými osami	m	+	-	PM-NC
POLFMLIN	P	Uvolnění os pro zpětný pohyb s lineární souvi-slostí mezi jednotlivými osami	m	+	-	PM-NC
POLY	G	Polynomická interpolace	m	+		PM-NC
POLYPATH	P	Polynomická interpolace může být vybrána pro skupiny os AXIS nebo VECT	m	+	-	PM-NC
PON	G	Zapnutí lisování	m	+		PM-NC
PONS	G	Zapnutí lisování v taktu IPO	m	+		PM-NC
POS	K	Polohování osy		+	+	PM-NC
POSA	K	Polohování osy přes hranici bloku		+	+	PM-NC
POSM	P	Polohování zásobníku		+	-	FM-TM

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
POSMT	P	Najetí modulem Moltitool v držáku nástroje na číslo místa		+	-	FM-TM
POSP	K	Polohování v úsecích (oscilace)		+		PM-NC
POSRANGE	F	Zjištění, zda se požadovaná poloha osy, jejíž interpolace právě probíhá, nachází v okně okolo předem zadané referenční pozice.		+	+	FM-SA
POT	F	Druhá mocnina (aritmetická funkce)		+	+	PM-NC
PR	K	na otáčku		+		PM-NC
PREPRO	PA	Označení podprogramů s přípravou		+		PM-NC
PRESETON	P	Nastavení skutečné hodnoty se ztrátou stavu najetí na referenční bod		+	+	PM-NC
PRESETONS	P	Nastavení skutečné hodnoty bez ztráty stavu najetí na referenční bod		+	+	PM-NC
PRIO	K	Klíčové slovo pro nastavení priority při spravování přerušení		+		PM-NC
PRLOC	K	Inicializace proměnných po resetu pouze po lokální změně		+		PM-NC
PROC	K	První příkaz programu		+		PM-NC
PROTA	P	Vyžádání nového výpočtu protikolizního modelu		+		PM-NC
PROTD	F	Výpočet vzdálenosti mezi dvěma chráněnými oblastmi		+		PM-NC
PROTS	P	Nastavení stavu chráněné oblasti		+		PM-NC
PSI	K	Úhel kužele		+		PM-NC
PTP	G	Pohyb od bodu k bodu (pohyb PTP)	m	+		PM-NC
PTPG0	G	Pohyb od bodu k bodu jen při G0, jinak pohyb po dráze s CP	m	+		PM-NC
PTPWOC	G	Pohyb od bodu k bodu bez kompenzačních pohybů, které jsou způsobovány změnami orientace	m	+		PM-NC
PUNCHACC	P	Zrychlení při prostřihování závislé na dráze		+	-	PM-NC
PUTFTOC	P	Jemná korekce nástroje pro paralelní orovnávání		+	-	PM-NC
PUTFTOCF	P	Jemná korekce nástroje v závislosti na funkci určené příkazem FCTDEF pro paralelní orovnávání		+	-	PM-NC
PW	A	B-Spline, váha uzlového bodu	s	+		PM-NC
QU	K	Výstup rychlých doplňkových (pomocných) funkcí		+		PM-NC
R...	A	Početní parametr, také jako nastavitelný adresový identifikátor a s numerickým rozšířením		+		PM-NC
RAC	K	Absolutní blokové programování rádiusů pro specifickou osu	s	+		PM-NC
RDISABLE	P	Zablokování načítání		-	+	FM-SA

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
READ	P	Načtení jednoho nebo více řádků ze zadaného souboru a uložení načtených informací do pole		+	-	PM-NC
REAL	K	Datový typ: Proměnná s plovoucí řádovou čárkou a se znaménkem (reálná čísla)		+		PM-NC
REDEF	K	Opětovná definice systémových proměnných, uživatelských proměnných a příkazů NC jazyka		+		PM-NC
RELEASE	P	Odblokování os stroje za účelem výměny os		+	+	PM-NC
REP	K	Klíčové slovo pro inicializaci všech prvků pole se stejnou hodnotou		+		PM-NC
REPEAT	K	Opakování programové smyčky		+		PM-NC
REPEATB	K	Opakování programového řádku		+		PM-NC
REPOSA	G	Zpětné najíždění na konturu lineárně všemi osami	s	+		PM-NC
REPOSH	G	Najetí zpět na konturu po půlkruhu	s	+		PM-NC
REPOSHA	G	Najetí zpět na konturu všemi osami; geometrické osy po půlkruhu	s	+		PM-NC
REPOSL	G	Najíždění na konturu po čtvrtkruhu	s	+		PM-NC
REPOSQ	G	Najetí zpět na konturu po čtvrtkruhu	s	+		PM-NC
REPOSQA	G	Najetí zpět na konturu lineárně všemi osami; geometrické osy po čtvrtkruhu	s	+		PM-NC
RESETMON	P	Příkaz jazyka pro aktivování požadované hodnoty		+	-	FM-TM
RET	P	Konec podprogramu		+	+	PM-NC
RETB	P	Konec podprogramu		+	+	PM-NC
RIC	K	Relativní blokové programování rádiusů pro specifickou osu	s	+		PM-NC
RINDEX	F	Stanovení indexu znaku ve vstupním řetězci		+	-	PM-NC
RMB	G	Zpětné najíždění na začátek bloku	m	+		PM-NC
RMBBL	G	Zpětné najíždění na začátek bloku	s	+		PM-NC
RME	G	Zpětné najíždění na konec bloku	m	+		PM-NC
RMEBL	G	Zpětné najíždění na konec bloku	s	+		PM-NC
RMI [6]	G	Zpětné najíždění na místo přerušení	m	+		PM-NC
RMIBL [6]	G	Zpětné najíždění na místo přerušení	s	+		PM-NC
RMN	G	Zpětné najíždění na nejbližší blok s bodem dráhy	m	+		PM-NC
RMNBL	G	Zpětné najíždění na nejbližší blok s bodem dráhy	s	+		PM-NC
RND	A	Zaoblení rohů kontury	s	+		PM-NC
RNDM	A	Modální zaoblení	m	+		PM-NC
ROT	G	Programovatelné otočení	s	+		PM-NC
ROTS	G	Programovatelná otáčení framu o prostorový úhel	s	+		PM-NC
ROUND	F	Zaokrouhlení desetinných míst		+	+	PM-NC
ROUNDUP	F	Zaokrouhlování vstupní hodnoty		+	+	PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
RP	A	Polární rádius	m/s	+		PM-NC
RPL	A	Rotace v rovině	s	+		PM-NC
RT	K	Parametr pro přístup k datům framu: Otočení		+		PM-NC
RTLIOF	G	G0 bez lineární interpolace (interpolace jednotlivých os)	m	+		PM-NC
RTLION [6]	G	G0 s lineární interpolací	m	+		PM-NC

Příkazy S ... Z

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
S	A	Otáčky vřetena (u G4, G96/G961 jiný význam)	m/s	+	+	PM-NC
SAVE	PA	Atribut pro záchranu informací při vyvolávání podprogramů		+		PM-NC
SBLOF	P	Potlačení zpracovávání blok po bloku		+	-	PM-NC
SBLON	P	Odstranění blokování zpracovávání blok po bloku		+	-	PM-NC
SC	K	Parametr pro přístup k datům framu: Změna měřítka		+		PM-NC
SCALE	G	Programovatelná změna měřítka	s	+		PM-NC
SCC	K	Selektivní přiřazení příčné osy příkazu G96/G961/G962. Identifikátorem osy mohou být geometrická, kanálová nebo strojní osa.		+		PM-NC
SCPARA	K	Programování bloku parametrů servomechanismu		+	+	PM-NC
SD	A	Stupeň splinu	s	+		PM-NC
SET	K	Klíčové slovo pro inicializaci všech prvků pole se zadanými hodnotami		+		PM-NC
SETAL	P	Aktivování alarmu		+	+	PM-NC
SETDNO	F	Přiřazení D-čísla břitu (CE) daného nástroje (T)		+	-	PM-NC
SETINT	K	Stanovení, která rutina přerušení má být aktivována, když se aktivuje daný vstup NC.		+		PM-NC
SETM	P	Nastavování značek ve vlastním kanálu		+	+	PM-NC
SETMS	P	Zpětné přepnutí na řídící vřeteno určené ve strojním parametru		+	-	PM-NC
SETMS(n)	P	Vřeteno n má platit jako řídící vřeteno		+		PM-NC
SETMTH	P	Definice čísla držáku hlavního nástroje		+	-	FM-TM
SETPIECE	P	Nastavení počtu kusů pro všechny nástroje, které jsou přiřazeny danému vřetenu.		+	-	FM-TM
SETTA	P	Aktivování nástroje ze skupiny opotřebení		+	-	FM-TM
SETTCOR	F	Změna komponent nástroje; při této změně jsou zohledňovány všechny okrajové podmínky		+	-	PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
SETTIA	P	Deaktivování nástroje ze skupiny opotřebení		+	-	FM-TM
SF	A	Úhlové posunutí počátečního bodu pro řezání závitů	m	+		PM-NC
SIN	F	Sinus (trigonometrická funkce)		+	+	PM-NC
SIRELAY	F	Aktivování bezpečnostních funkcí stanovených pomocí parametrů SIRELILN, SIRELOUT, SIRELTIME		-	+	FM-SI
SIRELIN	P	Inicializace vstupních veličin funkčního modulu		+	-	FM-SI
SIRELOUT	P	Inicializace výstupních veličin funkčního modulu		+	-	FM-SI
SIRELTIME	P	Inicializace časovače funkčního modulu		+	-	FM-SI
SLOT1	C (T)	Podélná drážka		+		PM-NC
SLOT2	C (T)	Kruhová drážka		+		PM-NC
SOFT	G	Zrychlení po dráze s omezením ryvu	m	+		PM-NC
SOFTA	P	Aktivování změn zrychlení pro naprogramované osy s omezením trhavých pohybů		+	-	PM-NC
SON	G	Aktivování prostřihování	m	+		PM-NC
SONS	G	Zapnutí prostřihování v taktu IPO	m	+		PM-NC
SPATH [6]	G	Referenční dráha pro osy v FGROUP je délka oblouku	m	+		PM-NC
SPCOF	P	Přepnutí řídícího vřetena nebo vřetena (n) z režimu polohové regulace do režimu regulace otáček	m	+	-	PM-NC
SPCON	P	Přepnutí řídícího vřetena nebo vřetena (n) z režimu regulace otáček do režimu regulace polohy	m	+	-	PM-NC
SPI	F	Převedení čísla vřetena v identifikátoru osy		+	-	PM-NC
SPIF1 [6]	G	Rychlé vstupy/výstupy NC pro lisování/prostřihování, byte 1	m	+		FM-TE
SPIF2	G	Rychlé vstupy/výstupy NC pro lisování/prostřihování, byte 2	m	+		FM-TE
SPLINEPATH	P	Definice pásma hodnot pro spliny		+	-	PM-NC
SPN	A	Počet úseků na blok	s	+		PM-NC
SPOF [6]	G	Vypnutí zdvihu, vypnutí lisování, prostřihování	m	+		PM-NC
SPOS	K	Poloha vřetena	m	+	+	PM-NC
SPOSA	K	Polohování vřetena přes hranice bloků	m	+		PM-NC
SPP	A	Délka úseku	m	+		PM-NC
SPRINT	F	Vrací zpět vstupní řetězec formátovaný		+		PM-NC
SQRT	F	Druhá odmocnina (aritmetická funkce) (square root)		+	+	PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
SR	A	Zpětná dráha oscilačního pohybu pro synchronní akci	s	+		PM-NC
SRA	K	Zpětná dráha oscilačního pohybu osy při externím vstupu pro synchronní akci	m	+		PM-NC
ST	A	Doba vyjiskřování s oscilačním pohybem pro synchronní akci	s	+		PM-NC
STA	K	Doba vyjiskřování s oscilačním pohybem osy pro synchronní akci	m	+		PM-NC
START	P	Spuštění zvoleného programu současně ve více kanálech z momentálně zpracovávaného programu		+	-	PM-NC
STARTFIFO [6]	G	Zpracovávat; souběžně s plněním paměti preprocesoru	m	+		PM-NC
STAT		Poloha kloubu	s	+		PM-NC
STOLF	A	Toleranční faktor G0	m	+		PM-NC
STOPFIFO	G	Zastavení zpracování; plnění paměti preprocesoru, dokud není zjištěn příkaz STARTFIFO, naplnění paměti preprocesoru nebo konec programu	m	+		PM-NC
STOPRE	P	Zastavení předběžného zpracování, dokud nejsou zpracovány všechny připravené bloky z hlavního zpracování programu.		+	-	PM-NC
STOPREOF	P	Odblokování zastavení předběžného zpracování		-	+	FM-SA
STRING	K	Datový typ: Řetězec znaků		+		PM-NC
STRINGIS	F	Kontrola existujícího rozsahu jazyka NC systému a speciálně u tohoto příkazu odpovídajících názvů NC-cyklů, uživatelských proměnných, maker a názvů návěští, zda existují, zda jsou platné, definované nebo aktivní.		+	-	PM-NC
STRLEN	F	Zjištění délky řetězce		+	-	PM-NC
SUBSTR	F	Stanovení indexu znaku ve vstupním řetězci		+	-	PM-NC
SUPA	G	Potlačení aktuálního posunutí počátku, včetně programovatelných posunutí, systémových framů, posunutí ručním kolečkem (DRF), externích posunutí a superponovaných pohybů	s	+		PM-NC
SUPD	G	Potlačení aktivních korekčních parametrů nástroje	s	+	-	PM-NC
SVC	K	Řezná rychlost nástroje	m	+		PM-NC
SYNFCT	P	Vyhodnocování polynomu v závislosti na podmínce v pohybové synchronní akci		-	+	FM-SA
SYNR	K	Načtení proměnné se uskutečňuje synchronně, tzn. v okamžiku zpracovávání		+		PM-NC
SYNRW	K	Načtení a zápis do proměnné se uskutečňuje synchronně, tzn. v okamžiku zpracovávání		+		PM-NC
SYNW	K	Zápis do proměnné se uskutečňuje synchronně, tzn. v okamžiku zpracovávání		+		PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
T	A	Vyvolání nástroje (výměna jen tehdy, je-li nastaveno strojním parametrem, jinak je třeba příkaz M6)		+		PM-NC
TAN	F	Tangens (trigonometrická funkce)		+	+	PM-NC
TANG	P	Tangenciální řízení: Definice vazby		+	-	PM-NC
TANGDEL	P	Tangenciální řízení: Vymazání vazby		+	-	PM-NC
TANGOF	P	Tangenciální řízení: Deaktivování vazby		+	-	PM-NC
TANGON	P	Tangenciální řízení: Aktivování vazby		+	-	PM-NC
TCA (828D: _TCA)	P	Volba nástroje / výměna nástroje nezávisle na jeho stavu		+	-	FM-TM
TCARR	A	Vyžádání držáku nástroje (s číslem „m“)		+		PM-NC
TCI	P	Výměna nástroje ze schránky do zásobníku		+	-	FM-TM
TCOABS [6]	G	Stanovení délkových složek nástroje z aktuální orientace nástroje	m	+		PM-NC
TCOFR	G	Stanovení složek délky nástroje z orientace aktivního framu	m	+		PM-NC
TCOFRX	G	Stanovení orientace nástroje aktivního framu při volbě tohoto nástroje, nástroj nasměrovaný v ose X	m	+		PM-NC
TCOFRY	G	Stanovení orientace nástroje aktivního framu při volbě tohoto nástroje, nástroj nasměrovaný v ose Y	m	+		PM-NC
TCOFRZ	G	Stanovení orientace nástroje aktivního framu při volbě tohoto nástroje, nástroj nasměrovaný v ose Z	m	+		PM-NC
THETA	A	Úhel otočení	s	+		PM-NC
TILT	A	Úhel bočního naklonění	m	+		PM-NC
TLIFT	P	Tangenciální řízení: Aktivování vytváření pomocných bloků		+	-	PM-NC
TML	P	Volba nástroje spolu s jeho číslem místa v zásobníku		+	-	FM-TM
TMOF	P	Deaktivování monitorování nástroje		+	-	PM-NC
TMON	P	Aktivování monitorování nástroje		+	-	PM-NC
TO	K	Příkaz pro zadání koncové hodnoty ve smyčce FOR s počítadlem		+		PM-NC
TOFF	A	Offset délky nástroje ve směru délkové složky nástroje, která se uplatňuje rovnoběžně s geometrickou osou zadanou v indexu.	m	+		PM-NC
TOFFL	A	Offset délky nástroje ve směru délkové složky nástroje L1, L2, příp. L3.	m	+		PM-NC
TOFFLR	A	Simultánní zadání offsetu délky a rádiusu nástroje	m	+		PM-NC
TOFFOF	P	Vynulování on-line korekce délky nástroje		+	-	PM-NC
TOFFON	P	Aktivování on-line korekce délky nástroje		+	-	PM-NC
TOFFR	A	Offset rádiusu nástroje	m	+		PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
TOFRAME	G	Nastavení osy Z systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje	m	+		PM-NC
TOFRAMEX	G	Nastavení osy X systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje	m	+		PM-NC
TOFRAMEY	G	Nastavení osy Y systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje	m	+		PM-NC
TOFRAMEZ	G	Stejné jako příkaz TOFRAME	m	+		PM-NC
TOLOWER	F	Přeměna všech písmen v řetězci na malá písmena		+	-	PM-NC
TOOLENV	F	Uložení do paměti všech aktuálních stavů, které jsou důležité pro vyhodnocování parametrů nástroje uchovávaných v paměti.		+	-	PM-NC
TOOLGNT	F	Zjištění počtu nástrojů ve skupině nástrojů		+	-	FM-TM
TOOLGT	F	Zjištění T-čísla nástroje z dané skupiny nástrojů		+	-	FM-TM
TOROT	G	Nastavení osy Z systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje	m	+		PM-NC
TOROTOF [6]	G	Zrušení otočení framu ve směru nástroje	m	+		PM-NC
TOROTX	G	Nastavení osy X systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje	m	+		PM-NC
TOROTY	G	Nastavení osy Y systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje	m	+		PM-NC
TOROTZ	G	Stejné jako příkaz TOROT	m	+		PM-NC
TOUPPER	F	Přeměna všech písmen v řetězci na velká písmena		+	-	PM-NC
TOWBCS	G	Hodnoty opotřebení v základním souřadném systému (BCS)	m	+		PM-NC
TOWKCS	G	Hodnoty opotřebení v souřadném systému hlavy nástroje při kinetické transformaci (liší se od MCS otočením nástroje)	m	+		PM-NC
TOWMCS	G	Hodnoty opotřebení v souřadném systému stroje (MCS)	m	+		PM-NC
TOWSTD [6]	G	Základní nastavení pro korekce hodnoty ve směru délky nástroje	m	+		PM-NC
TOWTCS	G	Hodnoty opotřebení v souřadném systému nástroje (vztažný bod držáku nástroje T na nástrojovém sklíčidle)	m	+		PM-NC
TOWWCS	G	Hodnoty opotřebení v souřadném systému obrobku (WCS)	m	+		PM-NC
TR	K	Složka posunutí proměnné typu FRAME		+		PM-NC
TRAANG	P	Transformace šikmé osy		+	-	PM-NC
TRACON	P	Kaskádová transformace		+	-	PM-NC
TRACYL	P	Válec: Transformace plášťové plochy		+	-	PM-NC
TRAFOOF	P	Vypnutí aktivních transformací v kanálu		+	-	PM-NC
TRAFOON	P	Aktivování transformace definované prostřednictvím kinematického řetězce		+	-	PM-NC
TRAILOF	P	Vypnutí asynchronního vlečení		+	+	PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
TRAILON	P	Zapnutí asynchronního vlečení		+	+	PM-NC
TRANS	G	Absolutní programovatelné posunutí počátku	s	+		PM-NC
TRANSMIT	P	Polární transformace (obrábění šikmých ploch)		+	-	PM-NC
TRAORI	P	4-, 5-osá transformace, generická transformace		+	-	PM-NC
TRUE	K	Logická konstanta: TRUE		+		PM-NC
TRUNC	F	Odříznutí míst za desetinnou čárkou		+	+	PM-NC
TU		Úhel osy	s	+		PM-NC
TURN	A	Počet závitů u spirály	s	+		PM-NC
ULI	K	Horní mezní hodnota pro proměnné		+		PM-NC
UNLOCK	P	Odblokování synchronní akce s identifikací ID (pokračování technologického cyklu)		-	+	FM-SA
UNTIL	K	Podmínka pro ukončení smyčky REPEAT		+		PM-NC
UPATH	G	Referenční dráha pro osy v FGROUP je křivkový parametr	m	+		PM-NC
VAR	K	Klíčové slovo: Druh předávání parametrů		+		PM-NC
VELOLIM	K	Přizpůsobení maximální rychlosti osy nebo otáček vřetena	m	+	-	PM-NC
VELOLIMA	K	Snížení nebo zvýšení maximální rychlosti vlečné osy	m	+	+	PM-NC
WAITC	P	Čekání, až bude pro osu/vřeteno splněno kritérium pro změnu bloku vazby.		+	-	PM-NC
WAITE	P	Čekání na konec programu v jiném kanálu		+	-	PM-NC
WAITENC	P	Čekání na synchronizované, příp. restaurované polohy os		+	-	PM-NC
WAITM	P	Čekání na značku v uvedeném kanálu; konec předešlého bloku s přesným najetím.		+	-	PM-NC
WAITMC	P	Čekání na značku v uvedeném kanálu; přesné najetí jen tehdy, pokud ostatní kanály značky ještě nedosáhly.		+	-	PM-NC
WAITP	P	Čekání na konec posuvu polohovací osy		+	-	PM-NC
WAITS	P	Čekání na dosažení pozice vřetena		+	-	PM-NC
WALCS0 [6]	G	Deaktivování ohraničení pracovního pole ve WCS	m	+	-	PM-NC
WALCS1	G	Skupina 1 ohraničení pracovního pole ve WCS aktivní	m	+	-	PM-NC
WALCS2	G	Skupina 2 ohraničení pracovního pole ve WCS aktivní	m	+	-	PM-NC
WALCS3	G	Skupina 3 ohraničení pracovního pole ve WCS aktivní	m	+	-	PM-NC
WALCS4	G	Skupina 4 ohraničení pracovního pole ve WCS aktivní	m	+	-	PM-NC
WALCS5	G	Skupina 5 ohraničení pracovního pole ve WCS aktivní	m	+	-	PM-NC
WALCS6	G	Skupina 6 ohraničení pracovního pole ve WCS aktivní	m	+	-	PM-NC

Příkaz	Typ [1]	Význam	W [2]	TP [3]	SA [4]	Popis viz [5]
[1] [2] [3] [4] [5] **Pokud budete potřebovat vysvětlení, viz Legenda (Strana 1213).**						
WALCS7	G	Skupina 7 ohraničení pracovního pole ve WCS aktivní	m	+	-	PM-NC
WALCS8	G	Skupina 8 ohraničení pracovního pole ve WCS aktivní	m	+	-	PM-NC
WALCS9	G	Skupina 9 ohraničení pracovního pole ve WCS aktivní	m	+	-	PM-NC
WALCS10	G	Skupina 10 ohraničení pracovního pole ve WCS aktivní	m	+	-	PM-NC
WALIMOF	G	Vypnutí ohraničení pracovního pole BCS	m	+	-	PM-NC
WALIMON [6]	G	Zapnutí ohraničení pracovního pole BCS	m	+	-	PM-NC
WHEN	K	Pokud je podmínka jedenkrát splněna, daná akce se jedenkrát uskuteční.		-	+	FM-SA
WHENEVER	K	Pokud je podmínka splněna, v každém taktu interpolátoru se bude daná akce cyklicky provádět.		-	+	FM-SA
WHILE	K	Začátek programové smyčky WHILE		+		PM-NC
WRITE	P	Zápis textu do systému souborů. Vložení bloku na konec zadaného souboru.		+	-	PM-NC
WRTPR	P	Zápis řetězce znaků do proměnné BTSS		+	-	PM-NC
X	A	Název osy	m/s	+	+	PM-NC
XOR	O	Logické XOR		+		PM-NC
Y	A	Název osy	m/s	+	+	PM-NC
Z	A	Název osy	m/s	+	+	PM-NC

Legenda

[1] Druh příkazu:

A Adresa

Identifikátor, kterému je hodnota přiřazována (např. OVR=10). Existují také některé adresy, které aktivují nebo deaktivují funkci i bez přiřazení nějaké hodnoty (např. CPLON a CPLOF).

C (A) Cyklus AST

Předem definovaný NC program pro automatickou dodatečnou optimalizaci s AST (= automatická optimalizace servomechanismu) Přizpůsobování podmínkám konkrétní situace při optimalizaci se uskutečňuje prostřednictvím parametrů, které jsou předávány při volání cyklu.

C (M) Měřicí cyklus

Předem definovaný NC-program, ve kterém je naprogramována určitá všeobecně platná měřicí operace, jako např. určování vnitřního průměru válcového obrobku. Přizpůsobování podmínkám konkrétní situace při měření se uskutečňuje prostřednictvím parametrů, které jsou předávány při volání cyklu.

C (T) Technologický cyklus

Předem definovaný NC program, ve kterém je naprogramována určitá obráběcí operace, jako je například vrtání závitu nebo frézování kapsy tak, aby byl všeobecně platný. Přizpůsobování podmínkám konkrétní situace při obrábění se uskutečňuje prostřednictvím parametrů, které jsou předávány při volání cyklu.

F Předdefinovaná funkce (poskytuje výslednou (vracenou) hodnotu)

Volání předem definované funkce se může nacházet i jako operand ve výrazu.

G G-příkaz

G-příkazy jsou rozděleny do skupin. V jednom bloku může být napsán jen jeden příkaz G-funkce z určité skupiny. Příkaz G-funkce může mít modální platnost (do odvolání jiným příkazem ze stejné skupiny) nebo platí jen pro blok, ve kterém se nachází (bloková působnost).

K Klíčové slovo

Identifikátor, který určuje syntaxi daného bloku. Klíčovému slovu nemůže být přiřazena žádná hodnota a pomocí klíčového slova nemůže být aktivována/deaktivována žádná funkce NC systému.

Příklady: Řídící struktury (IF, ELSE, ENDIF, WHEN, ...), zpracování programu(GOTOB, GOTO, RET ...)

O Operátor

Operátor pro matematické, porovnávací nebo logické operace

P Předdefinovaná procedura (neposkytuje žádnou výslednou (vracenou) hodnotu)

PA Atribut programu

Atributy programu se nacházejí na konci definičního řádku podprogramu:
`PROC <název programu>(...) <Atribut programu>`

Určují chování při zpracovávání podprogramu.

[2] Platnost příkazu:

m modální

s bloková

[3] Možnosti programování ve výrobním programu:

\+ programovatelné

\- nelze naprogramovat

M Může být naprogramováno jen výrobcem stroje

[4] Lze naprogramovat v synchronních akcích:
+ programovatelné
- nelze naprogramovat
T Lze naprogramovat jen v technologických cyklech

[5] Odkaz na dokument, který obsahuje podrobný popis příkazu:
FM-A Příručka k funkcím, Osy a vřetena
FM-B Příručka k funkcím, Základní funkce
FM-SA Příručka k funkcím, Synchronní akce
FM-SI Příručka k funkcím, Safety Integrated
FM-TE Příručka k funkcím, Technologie
FM-TM Příručka k funkcím, Správa nástrojů
FM-TR Příručka k funkcím, Transformace
PM-MC Programovací příručka, Měřicí cykly
PM-NC Programovací příručka, Programování NC systémů

[6] Standardní nastavení na začátku programu (stav systému při dodávce řídícího systému, pokud není naprogramováno nic jiného).

5.2 Adresy

5.2.1 Adresová písmena

Písmeno	Význam	Numerické rozšíření
A	Nastavitelný adresový identifikátor	x
B	Nastavitelný adresový identifikátor	x
C	Nastavitelný adresový identifikátor	x
D	Aktivování/deaktivování korekce délky nástroje, břit nástroje	
E	Nastavitelný adresový identifikátor	x
F	Posuv Doba prodlevy v sekundách	x
G	Příkaz G-funkce	
H	H-funkce	x
I	Nastavitelný adresový identifikátor	x
J	Nastavitelný adresový identifikátor	x
K	Nastavitelný adresový identifikátor	x
L	Název podprogramu, volání podprogramu	
M	M-funkce	x
N	Číslo vedlejšího bloku	
O	volné	

Písmeno	Význam	Numerické rozšíření
P	Počet průchodů programem	
Q	Nastavitelný adresový identifikátor	x
R	Identifikátor proměnných (R-parametr) Nastavitelný adresový identifikátor (bez numerického rozšíření)	x
S	Hodnota vřetena Doba prodlevy v otáčkách vřetena	x x
T	Číslo nástroje	x
U	Nastavitelný adresový identifikátor	x
V	Nastavitelný adresový identifikátor	x
W	Nastavitelný adresový identifikátor	x
X	Nastavitelný adresový identifikátor	x
Y	Nastavitelný adresový identifikátor	x
Z	Nastavitelný adresový identifikátor	x
%	Počáteční a oddělovací znak při předávání souborů	
:	Číslo hlavního bloku	
/	Označení přeskakovaného řádku	

5.2.2 Pevné adresy

Pevné adresy bez axiálního rozšíření

Identifikátor adresy	Typ adresy	modální/ bloková	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	QU	Datový typ přiřazované hodnoty
D	Číslo korekce	m								x	bez znaménka INT
F	Posuv, doba prodlevy	m, s	x							x	bez znaménka REAL
FLIM	Maximální rychlost pohybu po dráze	m									bez znaménka REAL
G	G-příkaz	Viz seznam G-funkcí									bez znaménka INT
H	Pomocné funkce	s								x	M: bez znaménka INT H: REAL

Identifikátor adresy	Typ adresy	modální/ bloková	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	QU	Datový typ přiřazované hodnoty
L	Číslo podprogramu	s									bez znaménka INT
M	Pomocné funkce	s								x	M: bez znaménka INT H: REAL
N	Číslo bloku	s									bez znaménka INT
OVR	Override	m									bez znaménka REAL
OVRRAP	Korekce (override) pro rychlost rychlého posuvu	m									bez znaménka REAL
P	Počet průchodů podprogramem	s									bez znaménka INT
PACCLIM	Maximální zrychlení při pohybu po dráze	m									bez znaménka REAL
S	Vřeteno, doba prodlevy	m, s								x	bez znaménka REAL
SCC	Přiřazení příčné osy příkazu G96/G961/ G962	m									REAL
SPOS	Poloha vřetena	m				x	x	x			REAL
SPOSA	Polohování vřetena přes hranice bloků	m				x	x	x			REAL
T	Číslo nástroje	m								x	bez znaménka INT

Pevné adresy s axiálním rozšířením

Identifikátor adresy	Typ adresy	modální/ bloková	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	QU	Datový typ přiřazované hodnoty
ACC	Zrychlení osy	m									bez znaménka REAL
ACCLIMA	Mezní hodnota zrychlení vlečné osy	m									bez znaménka REAL
AX	Proměnný identifikátor osy	[1]	x	x	x	x	x	x			REAL
FA	Posuv osy	m	x							x	bez znaménka REAL
FDA	Posuv osy při korekci ručním kolečkem	s	x								bez znaménka REAL
FGREF	Vztažný rádius	m	x	x							bez znaménka REAL
FL	Mezní hodnota osového posuvu	m	x								bez znaménka REAL
FMA	Synchronní posuv pro osu	m									bez znaménka REAL
FOC	Aktivování blokového najíždění s omezeným momentem	s									REAL
FOCOF	Deaktivování modálního najíždění s omezeným momentem	m									REAL
FOCON	Aktivování modálního najíždění s omezeným momentem	m									REAL
FXS	Aktivování najíždění na pevný doraz	m									bez znaménka INT
FXST	Mezní hodnota momentu pro najíždění na pevný doraz:	m									REAL

Identifikátor adresy	Typ adresy	modální/ bloková	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	QU	Datový typ přiřazované hodnoty
FXSW	Monitorovací okno pro najíždění na pevný doraz	m									REAL
IP	Proměnný interpolační parametr	s	x	x	x	x	x				REAL
JERKLIM	Omezení ryvu osy	m									bez znaménka REAL
JERKLIMA	Mezní hodnota ryvu vlečné osy	m									bez znaménka REAL
MEAC	Cyklické měření	s									INT Režim a 1 - 4 spouštěcí události
MEASA	Měření jednou osou s vymazáním zbytkové dráhy	s									INT Režim a 1 - 4 spouštěcí události
MEAWA	Měření jednou osou bez vymazání zbytkové dráhy	s									INT Režim a 1 - 4 spouštěcí události
MOV	Spuštění polohovací osy	m	x	x	x	x	x	x	x		REAL
OS	Oscilační pohyb ZAP/VYP	m									bez znaménka INT
OSB	Počáteční bod oscilačního pohybu	m	x	x	x	x	x	x			REAL
OSCILL	Přiřazení os pro oscilační pohyb, zapnutí oscilačního pohybu	m									Osa: 1 - 3 přísuvné osy
OSCTRL	Možnosti oscilačního pohybu	m									Bez znaménka INT: Možnosti nastavení, bez znaménka INT: Možnosti pro deaktivování

Identifikátor adresy	Typ adresy	modální/ blokovà	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	QU	Datový typ přiřazované hodnoty
OSE	Koncový bod oscilačního pohybu	m	x	x	x	x	x	x			REAL
OSNSC	Počet vyjiskřovacích cyklů – oscilační pohyb	m									bez znaménka INT
OSP1	Levý bod obratu (oscilační pohyb)	m	x	x	x	x	x	x			REAL
OSP2	Pravý bod obratu (oscilační pohyb)	m	x	x	x	x	x	x			REAL
OST1	Doba zastavení v levém bodu obratu (oscilační pohyb)	m									REAL
OST2	Doba zastavení v pravém bodu obratu (oscilační pohyb)	m									REAL
OVRA	Override (korekce) osy	m	x								bez znaménka REAL
PO	Koeficienty polynomu	s	x	x		x	x	x			bez znaménka REAL
POLF	Pozice LIFTFAST	m	x	x							bez znaménka REAL
POS	Polohovací osa	m	x	x	x	x	x	x	x		REAL
POSA	Polohovací osa za hranice bloku	m	x	x	x	x	x	x	x		REAL
POSP	Polohování v úsecích (oscilace)	m	x	x	x	x	x	x			REAL: Koncová pozice Real: Délka úseku INT: Volitelný doplněk
STA	Doba vyjiskřování pro osu	m									bez znaménka REAL

Identifikátor adresy	Typ adresy	modální/ blokový	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	QU	Datový typ přiřazované hodnoty
SRA	Zpětný návrat při externím vstupním signálu (axiální)	m									bez znaménka REAL
VELOLIM	Mezní hodnota rychlosti osy	m									bez znaménka REAL
VELOLIMA	Mezní hodnota rychlosti vlečné osy	m									bez znaménka REAL

[1] absolutní koncové body: modální, inkrementální koncové body: blokové, jinak modální/blokové v závislosti na syntaktických pravidlech příkazu G-funkce.

5.2.3 Nastavitelné adresy

Adresový identifikátor (standardní nastavení)	Typ adresy	modální/ blokový	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	PR, PM	QU	Max. počet	Datový typ přiřazované hodnoty
Hodnoty os a koncové body											
X, Y, Z, A, B, C	Osa	[1]	x	x	x	x				8	REAL
AP	Polární úhel	m/s [1]	x	x	x					1	REAL
RP	Polární rádius	m/s [1]	x	x	x					1	bez znaménka REAL
Orientace nástroje											
A2, B2, C2	Eulerův úhel nebo úhel RPY	s								3	REAL
A3, B3, C3	Složky směrového vektoru	s								3	REAL
A4, B4, C4	Složky normálového vektoru plochy na začátku bloku	s								3	REAL
A5, B5, C5	Složky normálového vektoru plochy na konci bloku	s								3	REAL

Adresový identifikátor (standardní nastavení)	Typ adresy	modální/ blokový	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	PR, PM	QU	Max. počet	Datový typ přiřazované hodnoty
A6, B6, C6	Složky směrového vektoru pro osu kužele	s								3	REAL
A7, B7, C7	Vektorové složky orientace ve vnitřním bodě na plášťové ploše kužele	s								3	REAL
LEAD	Předozadní úhel	m								1	REAL
THETA	Úhel otočení okolo směru nástroje	m		x	x					1	REAL
TILT	Úhel bočního naklonění	m								1	REAL
ORIS	Změna orientace (vztažená na dráhu)	m								1	REAL
Interpolační parametr											
I, J, K	Interpolační parametry souřadnice vnitřního bodu	s		x [2)]	x [2)]					3	REAL
I1, J1, K1		s	x	x	x					3	REAL
RPL	Rotace v rovině	s								1	REAL
CR	Rádius kruhu	s								1	bez znaménka REAL
AR	Úhel kruhové výseče	s								1	bez znaménka REAL
TURN	Počet závitů u spirály	s								1	bez znaménka INT
PL	Parametr interval-délka	s								1	bez znaménka REAL
PW	Bod – Hmotnost	s								1	bez znaménka REAL
SD	Stupeň splinu	m								1	bez znaménka INT
TU	Úhel osy	s								1	bez znaménka INT
STAT	Poloha kloubu	m								1	bez znaménka INT
SF	Úhlové posunutí počátečního bodu pro řezání závitů	m								1	REAL

Adresový identifikátor (standardní nastavení)	Typ adresy	modální/ blokový	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	PR, PM	QU	Max. počet	Datový typ přiřazované hodnoty
DISCL	Safety distance (Bezpečnostní vzdálenost) WAB	s								1	bez znaménka REAL
DISR	Vzdálenost pro zpětné polohování / vzdálenost WAB	s								1	bez znaménka REAL
DISPR	Rozdíl dráhy pro zpětné polohování	s								1	bez znaménka REAL
ALF	Úhel pro rychlé pozvednutí	m								1	bez znaménka INT
DILF	Délka pro rychlé pozvednutí	m								1	REAL
FP	Pevný bod: Číslo pevného bodu, na který se má najíždět	s								1	bez znaménka INT
RNDM	Modální zaoblení	m								1	bez znaménka REAL
RND	Blokové zaoblení	s								1	bez znaménka REAL
CHF	Bloková faseta	s								1	bez znaménka REAL
CHR	Faseta ve směru předcházejícího pohybu	s								1	bez znaménka REAL
ANG	Úhel konturové křivky	s								1	REAL
ISD	Hloubka zajíždění nástroje	m								1	REAL
DISC	Převýšení přechodové kružnice – korekce rádiusu nástroje	m								1	bez znaménka REAL
OFFN	Offset kontury	m								1	REAL
DITS	Náběh závitu	m								1	REAL
DITE	Výběr závitu	m								1	REAL
Kritéria pro přechodová zaoblení											
ADIS	Vzdálenost zaoblení	m								1	bez znaménka REAL
ADISPOS	Vzdálenost pro zaoblení pro rychlý posuv	m								1	bez znaménka REAL

Adresový identifikátor (standardní nastavení)	Typ adresy	modální/ blokový	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	PR, PM	QU	Max. počet	Datový typ přiřazované hodnoty
Měření											
MEAS	Měření se spínací sondou	s								1	bez znaménka INT
MEAW	Měření se spínací sondou bez vymazání zbytkové dráhy	s								1	bez znaménka INT
Chování os a vřeten											
LIMS	Omezení otáček vřetena	m								1	bez znaménka REAL
COARSEA	Chování při přechodu na další blok: Hrubé přesné axiální najetí	m									
FINEA	Chování při přechodu na další blok: Jemné přesné axiální najetí	m									
IPOENDA	Chování při přechodu na další blok: Axiální zastavení interpolátoru	m									
DIACYCOFA	Příčná osa: Vypnutí programování průměrů pro specifickou osu v cyklech	m									
DIAM90A	Příčná osa: Axiální programování průměrů při příkazu G90	m									
DIAMCHAN	Příčná osa: Převzetí programování průměrů všech příčných os v daném kanálu	m									
DIAMCHANA	Příčná osa: Převzetí programování průměrů ze specifického kanálu	m									

Adresový identifikátor (standardní nastavení)	Typ adresy	modální/ blokový	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	PR, PM	QU	Max. počet	Datový typ přiřazované hodnoty
DIAMOFA	Příčná osa: Axiální vypnutí programování průměrů	m									
DIAMONA	Příčná osa: Axiální zapnutí programování průměrů	m									
GP	Poloha: Nepřímé programování atributů polohy	m									
Posuvy											
FAD	Rychlost pomaléhoho přísuvného pohybu	s						x		1	bez znaménka REAL
FD	Posuv po dráze pro korekci pomocí ručního kolečka	s								1	bez znaménka REAL
FRC	Posuv pro rádius a fasetu	s								1	bez znaménka REAL
FRCM	Modální posuv pro rádius a fasetu	m								1	bez znaménka REAL
FB	Blokový posuv	s								1	bez znaménka REAL
Prostřihování / lisování											
SPN	Počet úseků na blok	s								1	INT
SPP	Délka úseku	m								1	REAL
Broušení											
ST	Doba vyjiskřování	s								1	bez znaménka REAL
SR	Návratová dráha	s								1	bez znaménka REAL
Volba nástroje											
TCARR	Držák nástroje	m								1	INT
Správa nástrojů											
DL	Číslo korekce nástroje	m								1	INT

Adresový identifikátor (standardní nastavení)	Typ adresy	modální/ blokový	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	PR, PM	QU	Max. počet	Datový typ přiřazované hodnoty
Adresy OEM											
OMA1	Adresa OEM 1	m		x	x	x				1	REAL
OMA2	Adresa OEM 2	m		x	x	x				1	REAL
OMA3	Adresa OEM 3	m		x	x	x				1	REAL
OMA4	Adresa OEM 4	m		x	x	x				1	REAL
OMA5	Adresa OEM 5	m		x	x	x				1	REAL
Ostatní											
CUTMOD	Modifikace korekčních parametrů u točivých nástrojů (ve spojení s orientovatelnými držáky nástrojů)	m									INT
CUTMODK	Modifikace korekčních parametrů u točivých nástrojů (ve spojení s transformacemi orientace, které byly definovány prostřednictvím kinematického řetězce)	m									STRING
TOFF	Offset délky nástroje rovnoběžně se zadanou geometrickou osou	m									REAL
TOFFL	Offset délky nástroje ve směru délkové složky nástroje L1, L2, příp. L3.	m									REAL
TOFFR	Offset rádiusu nástroje	m									REAL
TOFFLR	Simultánní zadání offsetu délky a rádiusu nástroje	m									REAL

[1] absolutní koncové body: modální, inkrementální koncové body: blokové, jinak modální/blokové v závislosti na syntaktických pravidlech příkazu G-funkce.

[2] Jako střed kruhu se parametr IPO chová inkrementálně. Pomocí AC můžete naprogramovat absolutně. Při jiných významech (např. stoupání závitu) je modifikace adresy ignorována.

5.3 G-příkazy

5.3.1 G-příkazy

G-příkazy jsou rozděleny do G-skupin. Ve výrobním programu nebo v synchronní akci smí být v jednom bloku napsán jen jeden G-příkaz z jedné G-skupiny. G-příkaz může mít modální nebo blokovou platnost.

Modální: až do naprogramování nějakého jiného G-příkazu ze stejné G.skupiny.

5.3.2 G-skupina 1: Příkazy pohybu s modální platností

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G0	1	Pohyb rychlým posuvem	+	m		
G1	2	Lineární interpolace (přímková interpolace)	+	m	x	
G2	3	Kruhová interpolace ve směru hodinových ručiček	+	m		
G3	4	Kruhová interpolace proti směru hodinových ručiček	+	m		
CIP	5	Kruhová interpolace přes vnitřní bod	+	m		
ASPLINE	6	Akimovy spliny	+	m		
BSPLINE	7	B-Spline	+	m		
CSPLINE	8	Kubické spliny	+	m		
POLY	9	Polynomická interpolace	+	m		
G33	10	Řezání závitů s konstantním stoupáním	+	m		
G331	11	Vrtání závitů	+	m		
G332	12	Zpětný pohyb (vrtání závitu)	+	m		
OEMIPO1	13	rezervováno	+	m		
OEMIPO2	14	rezervováno	+	m		
CT	15	Kruh s tangenciálním přechodem	+	m		
G34	16	Řezání závitu s lineárně narůstajícím stoupáním	+	m		
G35	17	Řezání závitu s lineárně klesajícím stoupáním	+	m		
INVCW	18	Evolventní interpolace ve směru hodinových ručiček	+	m		
INVCCW	19	Evolventní interpolace proti směru hodinových ručiček	+	m		
G335	20	Soustružení sudovitého závitu ve směru hodinových ručiček	+	m		
G336	21	Soustružení sudovitého závitu proti směru hodinových ručiček	+	m		

5.3.3 G-skupina 2: Pohyby s blokovou platností, doba prodlevy

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G4	1	Doba prodlevy určená časově	-	s		
G63	2	Vrtání závitů bez synchronizace	-	s		
G74	3	Najíždění na referenční bod se synchronizací	-	s		
G75	4	Najíždění na pevný bod	-	s		
REPOSL	5	Najíždění na konturu po čtvrtkruhu	-	s		
REPOSQ	6	Najetí zpět na konturu po čtvrtkruhu	-	s		
REPOSH	7	Najetí zpět na konturu po půlkruhu	-	s		
REPOSA	8	Zpětné najíždění na konturu lineárně všemi osami	-	s		
REPOSQA	9	Najetí zpět na konturu všemi osami po půlkruhu; geometrické osy po čtvrtkruhu	-	s		
REPOSHA	10	Najetí zpět na konturu všemi osami po půlkruhu; geometrické osy po půlkruhu	-	s		
G147	11	Najíždění na konturu po přímce	-	s		
G247	12	Najíždění na konturu po čtvrtkruhu	-	s		
G347	13	Najíždění na konturu po půlkruhu	-	s		
G148	14	Odjíždění od kontury po přímce	-	s		
G248	15	Odjíždění od kontury po čtvrtkruhu	-	s		
G348	16	Odjíždění od kontury po půlkruhu	-	s		
G5	17	Šikmé zapichovací broušení	-	s		
G7	18	Vyrovnávací pohyb při šikmém zapichovacím broušení	-	s		

5.3.4 G-skupina 3: Programovatelné framy, omezení pracovního pole a programování pólu

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
TRANS	1	TRANSLATION: Programovatelné posunutí	-	s		
ROT	2	ROTATION: Programovatelné otočení	-	s		
SCALE	3	SCALE: Programovatelná změna měřítka	-	s		
MIRROR	4	MIRROR: Programové zrcadlové převrácení	-	s		
ATRANS	5	Additive TRANSLATION: Aditivní programovatelné posunutí	-	s		
AROT	6	Additive ROTATION: Programovatelné otočení	-	s		
ASCALE	7	Additive SCALE: Programovatelná změna měřítka	-	s		
AMIRROR	8	Additive MIRROR: Programové zrcadlové převrácení	-	s		
-	9	volné	-	-		
G25	10	Dolní ohraničení pracovního pole/omezení otáček vřetena	-	s		

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G26	11	Horní ohraničení pracovního pole/omezení otáček vřetena	-	s		
G110	12	Programování pólu vztažené na naposled naprogramovanou požadovanou pozici	-	s		
G111	13	Programování pólu vzhledem k počátku aktuální souřadné soustavy obrobku	-	s		
G112	14	Programování pólu vzhledem k poslednímu platnému pólu	-	s		
G58	15	Absolutní programovatelné posunutí počátku	-	s		
G59	16	Aditivní programovatelné posunutí počátku	-	s		
ROTS	17	Otáčení o prostorový úhel	-	s		
AROTS	18	Aditivní otáčení o prostorový úhel	-	s		

5.3.5 G-skupina 4: FIFO

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
STARTFIFO	1	Spuštění FIFO Zpracovávání a souběžně s tím plnění paměti předběžného zpracování	+	m	x	
STOPFIFO	2	Stop FIFO Zastavení zpracování; plnění paměti předběžného zpracování, dokud není zjištěn příkaz STARTFIFO, naplnění paměti předběžného zpracování nebo konec programu	+	m		
FIFOCTRL	3	Spuštění automatického ukládání do paměti předběžného zpracovávání	+	m		

5.3.6 G-skupina 6: Volba roviny

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G17	1	Volba roviny 1. - 2. Geometrická osa	+	m	x	
G18	2	Volba roviny 3. - 1. Geometrická osa	+	m		
G19	3	Volba roviny 2. - 3. Geometrická osa	+	m		

5.3.7 G-skupina 7: Korekce rádiusu nástroje

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G40	1	Žádná korekce rádiusu nástroje	+	m	x	
G41	2	Korekce rádiusu nástroje vlevo od kontury	-	m		
G42	3	Korekce rádiusu nástroje vpravo od kontury	-	m		

5.3.8 G-skupina 8: Nastavitelné posunutí počátku

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G500	1	Deaktivování nastavitelných posunutí počátku (G54 ... G57, G505 ... G599)	+	m	x	
G54	2	1. nastavitelné posunutí počátku	+	m		
G55	3	2. nastavitelné posunutí počátku	+	m		
G56	4	3. nastavitelné posunutí počátku	+	m		
G57	5	4. nastavitelné posunutí počátku	+	m		
G505	6	5. nastavitelné posunutí počátku	+	m		
...	...	...	+	m		
G599	100	99. nastavitelné posunutí počátku	+	m		
S G-příkazy z této G-skupiny se vždy aktivuje příslušný nastavitelný uživatelský frame $P_UIFR[ ]. G54 odpovídá framu $P_UIFR[1], G505 odpovídá framu $P_UIFR[5]. Počet nastavitelných uživatelských framů a tím pádem i počet G-příkazů v této G-skupině může být nastaven pomocí parametru MD28080 $MC_MM_NUM_USER_FRAMES.						

5.3.9 G-skupina 9: Potlačení framových a nástrojových korekcí

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G53	1	Potlačení aktuálního framu: Programovatelný frame včetně systémového framu pro TOROT a TOFRAME a aktivního nastavitelného framu (G54 ... G57, G505 ... G599)	-	s		
SUPA	2	Stejné jako G153 včetně potlačení systémových framů pro nastavení skutečné hodnoty, škrábnutí, externích posunutí počátku, PAROT včetně posunutí ručním kolečkem (DRF), [externí posunutí počátku], superponované pohyby	-	s		
G153	3	Stejné jako G53 včetně potlačení všech specifických kanálových a/nebo v NCU globálního základního framu	-	s		
SUPD	4	Potlačení aktivních korekčních parametrů nástroje	-	s		

5.3.10 G-skupina 10 Přesné najetí – režim řízení pohybu po dráze

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G60	1	Přesné najetí	+	m	x	
G64	2	Režim řízení pohybu po dráze	+	m		
G641	3	Režim řízení pohybu po dráze s přechodovými zaobleními podle kritéria dráhy (= programovatelná vzdálenost zaoblení)	+	m		
G642	4	Režim řízení pohybu po dráze s přechodovými zaobleními při dodržení definovaných tolerancí	+	m		
G643	5	Režim řízení pohybu po dráze s přechodovými zaobleními při dodržení definovaných tolerancí (uvnitř bloku)	+	m		
G644	6	Režim řízení pohybu po dráze s přechodovými zaobleními s maximální možnou dynamikou	+	m		
G645	7	Režim řízení pohybu po dráze s přechodovými zaobleními v rozích a s tangenciálními přechody mezi bloky při dodržení definovaných tolerancí	+	m		

5.3.11 G-skupina 11: Blokové přesné najetí

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G9	1	Přesné najetí	-	s		

5.3.12 G-skupina 12: Kritéria přechodu na další blok při přesném najetí (G60/G9)

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G601	1	Přechod na další blok při jemném přesném najetí	+	m	x	
G602	2	Přechod na další blok při hrubém přesném najetí	+	m		
G603	3	Přechod na další blok při konci bloku IPO	+	m		

5.3.13 G-skupina 13: Kótování obrobku v palcích/metrických jednotkách

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G70	1	Jednotky pro zadávání rozměrů – palce (délky)	+	m		
G71	2	Jednotky pro zadávání rozměrů – metrické jednotky mm (délky)	+	m	x	

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G700	3	Jednotky pro zadávání rozměrů – palce, palce/min (délky + rychlost + systémové proměnné	+	m		
G710	4	Jednotky pro zadávání rozměrů – mm, mm/min (délky + rychlost + systémové proměnné	+	m		

5.3.14 G-skupina 14: Kótování obrobku absolutní/inkrementální

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G90	1	Zadávání absolutních rozměrů	+	m	x	
G91	2	Zadávání inkrementálních rozměrů	+	m		

5.3.15 G-skupina 15: Typ posuvu

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G93	1	Časově reciproční posuv v jednotkách 1/min	+	m		
G94	2	Lineární posuv mm/min, palce/min	+	m	x	
G95	3	Otáčkový posuv v jednotkách mm/ot, palců/ot	+	m		
G96	4	Otáčkový posuv (stejně jako u G95) a konstantní řezná rychlost)	+	m		
G97	5	Otáčkový posuv a konstantní otáčky vřetena (konstantní řezná rychlost je vypnuta)	+	m		
G931	6	Zadání posuvu pomocí doby posuvu, konstantní rychlost po dráze vypnuta	+	m		
G961	7	Lineární posuv (stejně jako u G94) a konstantní řezná rychlost	+	m		
G971	8	Lineární posuv a konstantní otáčky vřetena (konstantní řezná rychlost je vypnuta)	+	m		
G942	9	Lineární posuv a konstantní řezná rychlost nebo konstantní otáčky vřetena	+	m		
G952	10	Otáčkový posuv a konstantní řezná rychlost nebo konstantní otáčky vřetena	+	m		
G962	11	Lineární posuv nebo otáčkový posuv a konstantní řezná rychlost	+	m		
G972	12	Lineární posuv nebo otáčkový posuv a konstantní otáčky vřetena (konstantní řezná rychlost je vypnuta)	+	m		
G973	13	Otáčkový posuv bez omezení otáček vřetena a konstantní otáčky vřetena (G97 bez LIMS pro režim ISO)	+	m		

5.3.16 G-skupina 16: Korekce posuvu (override) na vnitřních a vnějších zakřiveních

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
CFC	1	Konstantní posuv na kontuře je v platnosti na vnitřních a vnějších zakřiveních	+	m	x	
CFTCP	2	Konstantní posuv na vztažném bodu nástroje (středová dráha)	+	m		
CFIN	3	Konstantní posuv na vnitřních zakřiveních, zrychlení na vnějších zakřiveních	+	m		

5.3.17 G-skupina 17: Chování při najíždění/odjíždění s korekcí rádiusu nástroje

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
NORM	1	Normální nastavení v počátečním/koncovém bodě	+	m	x	
KONT	2	Objíždění kontury v počátečním/koncovém bodě	+	m		
KONTT	3	Tangenciální najíždění/odjíždění	+	m		
KONTC	4	Najíždění/odjíždění se spojitým zakřivením	+	m		

5.3.18 G-skupina 18: Chování korekce nástroje v rozích

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G450	1	Přechodový prvek kruh (nástroj objíždí rohy obrobku po kruhové dráze)	+	m	x	
G451	2	Průsečík ekvidistantních drah (nástroj řeže ostré rohy obrobku)	+	m		

5.3.19 G-skupina 19: Křivkový přechod na začátku splinu

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
BNAT	1	Přirozený křivkový přechod na první splinový blok	+	m	x	
BTAN	2	Tangenciální křivkový přechod na první splinový blok	+	m		
BAUTO	3	Definice prvního splinového úseku prostřednictvím následujících 3 bodů	+	m		

5.3.20 G-skupina 20: Křivkový přechod na konci splinu

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
ENAT	1	Přirozený přechod na následující blok posuvu	+	m	x	
ETAN	2	Tangenciální křivkový přechod na následující blok posuvu	+	m		
EAUTO	3	Definice posledního splinového úseku prostřednictvím posledních 3 bodů	+	m		

5.3.21 G-skupina 21: Profil zrychlení

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
BRISK	1	Skokové změny zrychlení po dráze	+	m	x	
SOFT	2	Zrychlení po dráze s omezením ryvu	+	m		
DRIVE	3	Zrychlení po dráze závislé na rychlosti	+	m		

5.3.22 G-skupina 22: Typ korekce nástroje

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
CUT2D	1	2½-D-korekce rádiusu nástroje	+	m	x	
CUT2DF	2	2½-D-korekce rádiusu nástroje, vzhledem k aktuálnímu framu (nakloněná rovina)	+	m		
CUT3DC	3	3D korekce rádiusu nástroje pro obvodové frézování	+	m		
CUT3DF	4	3D korekce rádiusu nástroje pro frézování na čelní ploše se změnou orientace	+	m		
CUT3DFS	5	3D korekce rádiusu nástroje pro frézování na čelní ploše s konstantní orientací. Orientace nástroje je definována příkazy G17 - G19 a nemůže být ovlivňována pomocí framů.	+	m		
CUT3DFF	6	3D korekce rádiusu nástroje pro frézování na čelní ploše s konstantní orientací. Orientace nástroje je definována příkazy G17 - G19 a v případě potřeby může být pomocí framu pootočena do jiného směru.	+	m		
CUT3DCC	7	3D korekce rádiusu nástroje pro obvodové frézování se zohledňováním omezovací plochy s 3D korekcí rádiusu: Kontura na obráběné ploše	+	m		
CUT3DCCD	8	3D korekce rádiusu nástroje pro obvodové frézování se zohledňováním omezovací plochy s diferenčním nástrojem na dráze středu nástroje: Přísuv k omezující ploše.	+	m		

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
CUT2DD	9	2½-D-korekce rádiusu nástroje vztažená na diferenční nástroj	+	m		
CUT2DFD	10	2½-D-korekce rádiusu nástroje vztažená na diferenční nástroj, vzhledem k aktuálnímu framu (šikmá rovina)	+	m		
CUT3DCD	11	3D korekce rádiusu nástroje pro obvodové frézování vztažená na diferenční nástroj	+	m		
CUT3DFD	12	3D korekce rádiusu nástroje pro frézování na čelní ploše se změnou orientace vztažená na diferenční nástroj	+	m		

5.3.23 G-skupina 23: Monitorování možnosti kolize na vnitřních konturách

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
CDOF	1	Vypnutí monitorování kolizí	+	m	x	
CDON	2	Zapnutí monitorování kolizí	+	m		
CDOF2	3	Vypnutí monitorování kolizí při 3D obvodovém frézování	+	m		

5.3.24 G-skupina 24: Dopředná regulace

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
FFWOF	1	Vypnutí dopředné regulace	+	m	x	
FFWON	2	Zapnutí dopředné regulace	+	m		

5.3.25 G-skupina 25: Reference orientace nástroje

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
ORIWKS	1	Orientace nástroje v souřadném systému obrobku (WCS)	+	m	x	
ORIMKS	2	Orientace nástroje v souřadném systému stroje (MCS)	+	m		

5.3.26 G-skupina 26: Režim opětovného najíždění pro funkci REPOS (s modální platností)

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
RMB	1	Zpětné najíždění na začátek bloku	-	m		
RMI	2	Zpětné najíždění na místo přerušení	-	m	x	
RME	3	Zpětné najíždění na konec bloku	-	m		
RMN	4	Zpětné najíždění na nejbližší blok s bodem dráhy	-	m		

5.3.27 G-skupina 27: Korekce nástroje při změnách orientace na vnějších rozích

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
ORIC	1	Změny orientace na vnějších rozích jsou superponovány s vkládaným kruhovým blokem	+	m	x	
ORID	2.	Změna orientace se bude provádět před kruhovým blokem.	+	m		

5.3.28 G-skupina 28: Ohraničení pracovního pole

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
WALIMON	1	Zapnutí ohraničení pracovního pole	+	m	x	
WALIMOF	2	Vypnutí ohraničení pracovního pole	+	m		

5.3.29 G-skupina 29: Programování rádiusů/průměrů

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
DIAMOF	1	Vypnutí kanálového programování průměrů s modální platností Pomocí této funkce je po vypnutí v platnosti kanálové programování průměrů.	+	m	x	
DIAMON	2	Zapnutí nezávislého kanálového programování průměrů s modální platností Funkce nezávisí na naprogramovaném režimu zadávání rozměrů (G90/G91).	+	m		

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
DIAM90	3	Zapnutí závislého kanálového programování průměrů s modální platností Funkce závisí na naprogramovaném režimu zadávání rozměrů (G90/G91).	+	m		
DIAMCYCOF	4	Vypnutí kanálového programování průměrů s modální platností v průběhu zpracování cyklů	+	m		

5.3.30 G-skupina 30: Komprese NC bloků

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
COMPOF	1	Vypnutí komprese NC bloků	+	m	x	
COMPON	2	Zapnutí funkce kompresoru COMPON	+	m		
COMPCURV	3	Zapnutí funkce kompresoru COMPCURV	+	m		
COMPCAD	4	Zapnutí funkce kompresoru COMPCAD	+	m		
COMPSURF	5	Aktivování funkce kompresoru COMPSURF	+	m		

5.3.31 G-skupina 31: G-příkazy pro OEM

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G810	1	G-příkaz pro OEM	-	m		
G811	2	G-příkaz pro OEM	-	m		
G812	3	G-příkaz pro OEM	-	m		
G813	4	G-příkaz pro OEM	-	m		
G814	5	G-příkaz pro OEM	-	m		
G815	6	G-příkaz pro OEM	-	m		
G816	7	G-příkaz pro OEM	-	m		
G817	8	G-příkaz pro OEM	-	m		
G818	9	G-příkaz pro OEM	-	m		
G819	10	G-příkaz pro OEM	-	m		
Pro uživatele OEM jsou k dispozici dvě G-skupiny. To umožňuje OEM programovat funkce, které mohou být uživatelsky definovány.						

5.3.32 G-skupina 32: G-příkazy pro OEM

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G820	1	OEM - G-příkaz	-	m		
G821	2	OEM - G-příkaz	-	m		
G822	3	OEM - G-příkaz	-	m		
G823	4	OEM - G-příkaz	-	m		
G824	5	OEM - G-příkaz	-	m		
G825	6	OEM - G-příkaz	-	m		
G826	7	OEM - G-příkaz	-	m		
G827	8	OEM - G-příkaz	-	m		
G828	9	OEM - G-příkaz	-	m		
G829	10	OEM - G-příkaz	-	m		
Pro uživatele OEM jsou k dispozici dvě G-skupiny. To umožňuje OEM programovat funkce, které mohou být uživatelsky definovány.						

5.3.33 G-skupina 33: Nastavitelná jemná korekce nástroje

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
FTOCOF	1	Vypnutí on-line působící jemné korekce nástroje	+	m	x	
FTOCON	2	Zapnutí on-line působící jemné korekce nástroje	-	m		

5.3.34 G-skupina 34: Vyhlazení orientace nástroje

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
OSOF	1	Vypnutí vyhlazování orientace nástroje	+	m	x	
OSC	2	Konstantní vyhlazení orientace nástroje	+	m		
OSS	3	Vyhlazení orientace nástroje na konci bloku	+	m		
OSSE	4	Vyhlazení orientace nástroje na počátku a konci bloku	+	m		
OSD	5	Interní blokové zaoblování rohů se specifikovanou délkou dráhy	+	m		
OST	6	Interní blokové zaoblování rohů se specifikovanou úhlovou tolerancí	+	m		

5.3.35 G-skupina 35: Lisování a prostřihování

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
SPOF	1	Vypnutí zdvihu, vypnutí prostřihování a lisování	+	m	x	
SON	2	Aktivování prostřihování	+	m		
PON	3	Zapnutí lisování	+	m		
SONS	4	Zapnutí prostřihování v taktu IPO	-	m		
PONS	5	Zapnutí lisování v taktu IPO	-	m		

5.3.36 G-skupina 36: Lisování se zpožděním

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
PDELAYON	1	Aktivování zpoždění při lisování	+	m	x	
PDELAYOF	2	Deaktivování zpoždění při lisování	+	m		

5.3.37 G-skupina 37: Profil posuvu

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
FNORM	1	Normální posuv podle DIN 66025	+	m	x	
FLIN	2	Lineárně proměnný posuv	+	m		
FCUB	3	Posuv proměnný podle kubického splinu	+	m		

5.3.38 G-skupina 38: Přiřazení rychlých vstupů/výstupů pro lisování/prostřihování

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
SPIF1	1	Rychlé vstupy/výstupy NC pro lisování/prostřihování, byte 1	+	m	x	
SPIF2	2	Rychlé vstupy/výstupy NC pro lisování/prostřihování, byte 2	+	m		

5.3.39 G-skupina 39: Programovatelná přesnost kontury

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
CPRECOF	1	Vypnutí programovatelné přesnosti kontury	+	m	x	
CPRECON	2	Zapnutí programovatelné přesnosti kontury	+	m		

5.3.40 G-skupina 40: Konstantní korekce rádiusu nástroje

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
CUTCONOF	1	Vypnutí konstantní korekce rádiusu nástroje	+	m	x	
CUTCONON	2	Aktivování konstantní korekce rádiusu nástroje	+	m		

5.3.41 G-skupina 41: Možnost přerušení řezání závitu

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
LFOF	1	Možnost přerušení řezání závitu vypnuta	+	m	x	
LFON	2	Možnost přerušení řezání závitu zapnuta	+	m		

5.3.42 G-skupina 42: Držák nástroje

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
TCOABS	1	Stanovení délkových složek nástroje z aktuální orientace nástroje	+	m	x	
TCOFR	2	Stanovení složek délky nástroje z orientace aktivního framu	+	m		
TCOFRZ	3	Stanovení orientace nástroje aktivního framu při volbě tohoto nástroje, nástroj nasměrovaný v ose Z	+	m		
TCOFRY	4	Stanovení orientace nástroje aktivního framu při volbě tohoto nástroje, nástroj nasměrovaný v ose Y	+	m		
TCOFRX	5	Stanovení orientace nástroje aktivního framu při volbě tohoto nástroje, nástroj nasměrovaný v ose X		m		

5.3.43 G-skupina 43: Směr najíždění WAB

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G140	1	Směr najíždění WAB definován příkazy G41/G42	+	m	x	
G141	2	Směr najíždění WAB vlevo od kontury	+	m		
G142	3	Směr najíždění WAB vpravo od kontury	+	m		
G143	4	Směr najíždění WAB v závislosti na tečně	+	m		

5.3.44 G-skupina 44: Rozdělení dráhy WAB

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G340	1	Najíždějící bloku v prostoru, tzn. přísuv do hloubky a najíždění v rovině v jednom bloku	+	m	x	
G341	2	Napřed přísuv v kolmé ose (Z), pak najíždění v rovině	+	m		

5.3.45 G-skupina 45: Vztah dráhy a os v FGROUP

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
SPATH	1	Referenční dráha pro osy v FGROUP je délka oblouku	+	m	x	
UPATH	2	Referenční dráha pro osy v FGROUP je křivkový parametr	+	m		

5.3.46 G-skupina 46: Volba roviny pro rychlé pozvednutí

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
LFTXT	1	Rovina je určena na základě tečny ke dráze a aktuální orientace nástroje	+	m	x	
LFWP	2	Rovina je určena prostřednictvím aktuální pracovní roviny (G17/G18/G19).	+	m		
LFPOS	3	Pozvednutí osy na pozici	+	m		

5.3.47 G-skupina 47: Přepínání režimu pro externí NC-kód

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G290	1	Aktivování režimu jazyka systému SINUMERIK	+	m	x	
G291	2	Aktivování režimu jazyka ISO	+	m		

5.3.48 G-skupina 48: Chování při najíždění/odjíždění s korekcí rádiusu nástroje

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G460	1	Aktivování monitorování kolizí pro na-/odjížděcí blok	+	m	x	
G461	2	Není-li v bloku korekce rádiusu nástroje žádný průsečík, prodloužení okrajového bloku s kruhovým obloukem	+	m		
G462	3	Není-li v bloku korekce rádiusu nástroje žádný průsečík, prodloužení okrajového bloku přímkou	+	m		

5.3.49 G-skupina 49: Pohyb od bodu k bodu

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
CP	1	Pohyb po dráze	+	m	x	
PTP	2	Pohyb od bodu k bodu (pohyb synchronní osy)	+	m		
PTPG0	3	Pohyb od bodu k bodu jen při G0, jinak pohyb po dráze s CP	+	m		
PTPWOC	4	Pohyb od bodu k bodu bez kompenzačních pohybů, které jsou způsobovány změnami orientace	+	m		

5.3.50 G-skupina 50: Programování orientace

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
ORIEULER	1	Úhel orientace pomocí Eulerova úhlu	+	m	x	
ORIRPY	2	Úhel orientace prostřednictvím úhlu RPY (posloupnost otáčení XYZ)	+	m		
ORIVIRT1	3	Úhel orientace pomocí virtuálních orientačních os (definice 1)	+	m		
ORIVIRT2	4	Úhel orientace pomocí virtuálních orientačních os (definice 2)	+	m		

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
ORIAXPOS	5	Úhel orientace pomocí virtuálních orientačních os s polohováním kruhové osy	+	m		
ORIRPY2	6	Úhel orientace pomocí úhlu RPY (posloupnost otáčení ZYX)	+	m		

5.3.51 G-skupina 51: Programování druhu interpolace orientace

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
ORIVECT	1	Interpolace pomocí největší kružnice koule (identická s ORIPLANE)	+	m	x	
ORIAXES	2	Lineární interpolace os stroje nebo orientačních os	+	m		
ORIPATH	3	Dráha orientace nástroje vztažená na dráhu	+	m		
ORIPLANE	4	Interpolace v rovině (odpovídá ORIVECT)	+	m		
ORICONCW	5	Interpolace po ploše pláště kužele ve směru hodinových ručiček	+	m		
ORICONCCW	6	Interpolace po ploše pláště kužele proti směru hodinových ručiček	+	m		
ORICONIO	7	Interpolace po ploše pláště kužele s udáním pomocné meziorientace	+	m		
ORICONTO	8	Interpolace na ploše pláště kuželu s tangenciálním přechodem	+	m		
ORICURVE	9	Interpolace s dodatečnou prostorovou křivkou pro orientaci	+	m		
ORIPATHS	10	Orientace nástroje vztažená na dráhu, zlom v průběhu orientace se vyhladí	+	m		

5.3.52 G-skupina 52: Otáčení framu vztahující se na obrobek

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
PAROTOF	1	Vypnutí otáčení framu vztahující se na obrobek	+	m	x	
PAROT	2	Zapnutí otáčení framu vztahující se na obrobek Srovnání souřadného systému obrobku s obrobkem.	+	m		

5.3.53 G-skupina 53: Otáčení framu vztahující se na nástroj

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
TOROTOF	1	Vypnutí otáčení framu vztahující se na nástroj	+	m	x	
TOROT	2	Nastavení osy Z systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje	+	m		
TOROTZ	3	Stejné jako příkaz TOROT	+	m		
TOROTY	4	Nastavení osy Y systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje	+	m		
TOROTX	5	Nastavení osy X systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje	+	m		
TOFRAME	6	Nastavení osy Z systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje	+	m		
TOFRAMEZ	7	Stejné jako příkaz TOFRAME	+	m		
TOFRAMEY	8	Nastavení osy Y systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje	+	m		
TOFRAMEX	9	Nastavení osy X systému WCS prostřednictvím otáčení framu rovnoběžně s orientací nástroje	+	m		

5.3.54 G-skupina 54: Otáčení vektoru při programování polynomu

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
ORIROTA	1	Otočení vektoru absolutní	+	m	x	
ORIROTR	2	Otočení vektoru relativní	+	m		
ORIROTT	3	Otočení vektoru tangenciální	+	m		
ORIROTC	4	Tangenciální vektor otočení k tečně dráhy	+	m		

5.3.55 G-skupina 55: Rychlý posuv s/bez lineární interpolace

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
RTLION	1	Zapnutí rychlého posuvu s lineární interpolací	+	m	x	
RTLIOF	2	Vypnutí rychlého posuvu s lineární interpolací Pohyb rychlým posuvem se uskutečňuje s interpolací jednotlivých os.	+	m		

5.3.56 G-skupina 56: Započítávání hodnot opotřebení nástroje

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
TOWSTD	1	Základní nastavení pro korekce hodnoty ve směru délky nástroje	+	m	x	
TOWMCS	2	Hodnoty opotřebení v souřadném systému stroje (MCS)	+	m		
TOWWCS	3	Hodnoty opotřebení v souřadném systému obrobku	+	m		
TOWBCS	4	Hodnoty opotřebení v základním souřadném systému (BCS)	+	m		
TOWTCS	5	Hodnoty opotřebení v souřadném systému nástroje (vztažný bod držáku nástroje T na nástrojovém sklíčidle)	+	m		
TOWKCS	6	Hodnoty opotřebení v souřadném systému hlavy nástroje při kinetické transformaci (liší se od MCS otočením nástroje)	+	m		

5.3.57 G-skupina 57: Zpoždění v rozích

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
FENDNORM	1	Vypnutí zpožďování v rozích	+	m	x	
G62	2	Zpoždění na vnitřních rozích při aktivní korekci rádiusu nástroje (G41/G42)	+	m		
G621	3	Snížení rychlosti na všech rozích	+	m		

5.3.58 G-skupina 59: Režim dynamiky pro dráhovou interpolaci

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
DYNNORM	1	Normální dynamika	+	m	x	
DYNPOS	2	Režim polohování, vrtání závitu	+	m		
DYNROUGH	3	Obrábění nahrubo	+	m		
DYNSEMIFIN	4	Předběžné obrábění načisto	+	m		
DYNFINISH	5	Obrábění načisto	+	m		
DYNPREC	6	Jemné obrábění načisto	+	m		

5.3.59 G-skupina 60: Ohraničení pracovního pole

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
WALCS0	1	Deaktivování ohraničení pracovního pole ve WCS	+	m	x	
WALCS1	2	Skupina 1 ohraničení pracovního pole ve WCS aktivní	+	m		
WALCS2	3	Skupina 2 ohraničení pracovního pole ve WCS aktivní	+	m		
WALCS3	4	Skupina 3 ohraničení pracovního pole ve WCS aktivní	+	m		
WALCS4	5	Skupina 4 ohraničení pracovního pole ve WCS aktivní	+	m		
WALCS5	6	Skupina 5 ohraničení pracovního pole ve WCS aktivní	+	m		
WALCS6	7	Skupina 6 ohraničení pracovního pole ve WCS aktivní	+	m		
WALCS7	8	Skupina 7 ohraničení pracovního pole ve WCS aktivní	+	m		
WALCS8	9	Skupina 8 ohraničení pracovního pole ve WCS aktivní	+	m		
WALCS9	10	Skupina 9 ohraničení pracovního pole ve WCS aktivní	+	m		
WALCS10	11	Skupina 10 ohraničení pracovního pole ve WCS aktivní	+	m		

5.3.60 G-skupina 61: Vyhlazení orientace nástroje

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
ORISOF	1	Vypnutí vyhlazování orientace nástroje	+	m	x	
ORISON	2	Zapnutí vyhlazování orientace nástroje	+	m		

5.3.61 G-skupina 62: Režim opětovného najíždění pro funkci REPOS (s blokovou platností)

G-příkaz	Č. [1]	Význam	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
RMBBL	1	Zpětné najíždění na začátek bloku	-	s		
RMIBL	2	Zpětné najíždění na místo přerušení	-	s	x	
RMEBL	3	Zpětné najíždění na konec bloku	-	s		
RMNBL	4	Zpětné najíždění na nejbližší blok s bodem dráhy	-	s		

5.3.62 G-skupina 64: Frame pro broušení

G-příkaz	Č. [1]	Význam Aktivní frame pro broušení v kanálu $P_GFRAME =	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
GFRAME[0]	1	Frame pro broušení ve správě dat $P_GFR[0] (nulový frame)	+	m	x	
GFRAME[1]	2	Frame pro broušení ve správě dat $P_GFR[1]	+	m		
GFRAME[2]	3	Frame pro broušení ve správě dat $P_GFR[2]	+	m		
...	...		+	m		
GFRAME[100]	101	Frame pro broušení ve správě dat $P_GFR[100]	+	m		

Legenda

1) Interní číslo (např. pro rozhraní PLC)

2) Možnost konfiguračního nastavení G-příkazu jako implicitní nastavení skupiny G-funkcí při náběhu systému, resetu, příp. po skončení výrobního programu (pomocí parametru MD20150 $MC_GCODE_RESET_VALUES):

+ může být nastavena v konfiguraci

- nemůže být nastavena v konfiguraci

3) Působnost G-příkazu:

m modální (přes hranice bloku)

s bloková

4) Nastavení deaktivování, viz následující strojní parametry:

- MD20149 $MC_GCODE_RESET_S_VALUES (nastavení deaktivování G-skupin (fix))
- MD20150 $MC_GCODE_RESET_VALUES (nastavení deaktivování G-skupin)
- MD20151 $MC_GCODE_RESET_S_MODE (chování G-skupin při resetu (fix))
- MD20152 $MC_GCODE_RESET_MODE (chování G-skupin při resetu)
- MD20154 $MC_EXTERN_GCODE_RESET_VALUES (nastavení deaktivování G-skupin v režimu ISO)
- MD20156 $MC_EXTERN_GCODE_RESET_MODE (chování externích G-skupin při resetu)

SAG Standardní nastavení firmy **S**iemens **AG**

MH Standardní nastavení od výrobce stroje (**M**aschinen**h**ersteller) (viz informace od výrobce stroje)

5.4 Předem definované procedury

Prostřednictvím volání předem definované procedury se spouští zpracování předem definované funkce NC systému. Předem definovaná procedura na rozdíl od předem definovaných funkcí neposkytuje **žádnou** hodnotu předávanou systémem.

Souřadný systém					
Identifikátor	**Parametr**				**Vysvětlení**
	1.	**2.**	**3. - 15.**	**4. - 16.**	
PRESETON	AXIS *): Identifikátor osy Strojní osa	REAL: Předvolba posunutí G700/G710 kontext	jako 1 ...	jako 2 ...	Nastavení skutečné hodnoty pro naprogramované osy se ztrátou stavu najetí na referenční bod
PRESETONS	AXIS *): Identifikátor osy Strojní osa	REAL: Předvolba posunutí G700/G710 kontext	jako 1 ...	jako 2 ...	Nastavení skutečné hodnoty pro naprogramované osy bez ztráty stavu najetí na referenční bod
DRFOF					Vymazání posunutí DRF pro všechny osy přiřazené danému kanálu.

*) Na místě identifikátoru osy stroje se obecně může vyskytovat také identifikátor geometrické nebo pomocné osy, pokud je však možné jednoznačné přiřazení.

Skupiny os					
Identifikátor	**Parametr**				**Vysvětlení**
GEOAX	**1.**	**2.**	**3. / 5.**	**4. / 6.**	Volba paralelního souřadného systému
	INT: Číslo geometrické osy 1 - 3	AXIS: Identifikátor kanálové osy	jako 1	jako 2	
FGROUP	**1. – 8.**				Proměnné přiřazení hodnoty F: Stanovení os, na které se vztahuje posuv po dráze Maximální počet os: 8 Pomocí příkazu FGROUP ( ) bez udání parametru se aktivuje standardní nastavení pro přiřazení hodnoty F.
	AXIS: Identifikátor kanálové osy				
SPLINEPATH	**1.**	**2. - 9.**			Definice skupiny splinů Maximální počet os: 8
	INT: Skupina splinů (musí být 1)	AXIS: Identifikátor geometrické nebo doplňkové osy			

Skupiny os				
Identifikátor	**Parametr**			**Vysvětlení**
POLYPATH	**1.**	**2.**		Nastavení polynomické interpolace pro selektivní skupiny os
	STRING	STRING		

Vlečení							
Identifikátor	**Parametr**						**Vysvětlení**
	1.	**2.**	**3.**	**4.**	**5.**	**6.**	
TANG	AXIS: Název vlečené osy	AXIS: Řídící osa 1	AXIS: Řídící osa 2	REAL : Faktor vazby	CHAR: Volitelný doplněk: "B": Vlečení v BCS "W": Vlečení ve WCS	CHAR: Optimalizace: "S": Standardní "P": automaticky s dráhou zaoblení, úhlová tolerance	Tangenciální řízení: Definice vazby Z obou uvedených řídících os se stanoví tečna pro vlečení. Faktor vazby udává souvislost mezi změnou úhlu tečny a vlečenou osou. Zpravidla má hodnotu 1.
TANGON	AXIS: Název vlečené osy	REAL : Úhlový offset	REAL: Dráha zaoblení	REAL: Úhlová tolerance			Tangenciální řízení: Aktivování vazby
TANGOF	AXIS: Název vlečené osy						Tangenciální řízení: Deaktivování vazby
TLIFT	AXIS: Vlečená osa						Tangenciální řízení: Aktivování vytváření pomocných bloků
TRAILON	AXIS: Vlečná osa	AXIS: Řídící osa	REAL : Faktor vazby				Aktivování asynchronního vlečení
TRAILOF	AXIS: Vlečná osa	AXIS: Řídící osa					Deaktivování asynchronního vlečení
TANGDEL	AXIS: Vlečná osa						Tangenciální řízení: Vymazání vazby

Tabulky křivek						
Identifikátor	**Parametr**					**Vysvětlení**
	1.	**2.**	**3.**	**4.**	**5.**	
CTABDEF	AXIS: Vlečná osa	AXIS: Řídící osa	INT: Číslo tabulky	INT: Chování na okrajích definiční oblasti	STRING: Zadání místa uložení v paměti	Zapnutí definice tabulek Následující pohybové bloky určují tabulku křivky.
CTABEND	AXIS: Vlečná osa	AXIS: Řídící osa	INT: Číslo tabulky	INT: Chování na okrajích definiční oblasti		Vypnutí definice tabulek
CTABDEL	INT: Čísla tabulek n	INT: Čísla tabulek m	STRING: Zadání místa uložení v paměti			Vymazání tabulky křivek
CTABLOCK	INT: Čísla tabulek n					Zablokuje tabulku křivky s číslem n, tzn. tato tabulka nemůže být vymazána/přepsána.
CTABUNLOCK	INT: Čísla tabulek n					U tabulky s číslem n, jejíž ochrana byla aktivována příkazem CTABLOCK , tuto ochranu zruší.
LEADON	AXIS: Vlečná osa	AXIS: Řídící osa	INT: Číslo tabulky			Aktivování vazby řídící hodnotou
LEADOF	AXIS: Vlečná osa	AXIS: Řídící osa				Deaktivování vazby řídící hodnotou

Profil zrychlení pro danou osu		
Identifikátor	**Parametr**	**Vysvětlení**
	1. – 8.	
BRISKA	AXIS	Aktivování skokových změn zrychlení pro naprogramované osy
SOFTA	AXIS	Aktivování změn zrychlení pro naprogramované osy s omezením trhavých pohybů
DRIVEA	AXIS	Zapnutí lomené charakteristiky zrychlení pro naprogramované osy
JERKA	AXIS	Pro naprogramované osy bude v platnosti chování zrychlení nastavené strojním parametrem $MA_AX_JERK_ENABLE.

Otáčkový posuv			
Identifikátor	**Parametr**		**Vysvětlení**
FPRAON	**1.**	**2.**	Aktivování otáčkového posuvu osy
	AXIS: Osa, pro kterou se aktivuje otáčkový posuv	AXIS: Osa/vřeteno, od kterých bude otáčkový posuv odvozen. Pokud není naprogramována žádná osa, bude otáčkový posuv odvozen od řídícího vřetena.	
FPRAOF	**1. - n.**		Deaktivování otáčkového posuv osy Otáčkový posuv může být aktivován i pro více os najednou. Může být naprogramováno tolik os, kolik jich může být maximálně obsaženo v bloku.
	AXIS: Osy, pro které se deaktivuje otáčkový posuv		
FPR	**1.**		Volba kruhové osy/vřetena, od kterých se bude odvozovat otáčkový posuv dráhy u G95. Nastavení pomocí příkazu FPR má modální platnost.
	AXIS: Osa/vřeteno, od kterých bude otáčkový posuv odvozen. Pokud není naprogramována žádná osa, bude otáčkový posuv odvozen od řídícího vřetena.		

Transformace				
Identifikátor	**Parametr**			**Vysvětlení**
	1.	**2.**	**3.**	
TRACYL	REAL: Pracovní průměr	INT: Číslo transformace		Válec: Transformace plášťové plochy Pro jeden kanál může být nastaveno i několik transformací. Číslo transformace udává, která z transformací má být aktivována. Pokud je 2. parametr vypuštěn, aktivuje se skupina transformace definovaná strojním parametrem.
TRANSMIT	INT: Číslo transformace			Transmit: Polární transformace Pro jeden kanál může být nastaveno i několik transformací. Číslo transformace udává, která z transformací má být aktivována. Pokud je parametr vypuštěn, aktivuje se skupina transformace definovaná strojním parametrem.

Transformace				
Identifikátor	**Parametr**			**Vysvětlení**
	1.	**2.**	**3.**	
TRAANG	REAL: Úhel	INT: Číslo transformace		Transformace šikmé osy Pro jeden kanál může být nastaveno i několik transformací. Číslo transformace udává, která z transformací má být aktivována. Pokud je 2. parametr vypuštěn, aktivuje se skupina transformace definovaná strojním parametrem. Pokud úhel není naprogramován (TRAANG (,2) nebo TRAANG) má poslední použitý úhel modální platnost.
TRAORI	INT: Číslo transformace			4-, 5-osá transformace Pro jeden kanál může být nastaveno i několik transformací. Číslo transformace udává, která z transformací má být aktivována.
TRACON	INT: Číslo transformace	REAL : Další parametry závisí na MD.		Kaskádová transformace Význam parametrů závisí na druhu kaskádového řazení.
TRAFOOF				Deaktivování transformace
TRAFOON	STRING: Název datového bloku transformace	REAL: Vztažný, příp. pracovní průměr (pouze TRACYL)	BOOL: s/bez korekce stěny drážky (pouze TRACYL)	Aktivování transformace definované prostřednictvím kinematického řetězce

Vřeteno			
Identifikátor	**Parametr**		**Vysvětlení**
	1	**2. - n.**	
SPCON	INT: Číslo vřetena	INT: Číslo vřetena	Přepnutí do režimu vřetena s polohovou regulací.
SPCOF	INT: Číslo vřetena	INT: Číslo vřetena	Přepnutí do režimu vřetena s regulací otáček.
SETMS	INT: Číslo vřetena		Deklarace vřetena jako řídícího vřetena pro aktuální kanál Příkazem SETMS( ) bez udání parametrů se aktivuje nastavení předdefinované strojními parametry.

Broušení		
Identifikátor	**Parametr**	**Vysvětlení**
	1.	
GWPSON	INT: Číslo vřetena	Aktivování konstantní obvodové rychlosti brusného kotouče. Pokud číslo vřetena není naprogramováno, bude se aktivovat obvodová rychlost kotouče pro vřeteno aktivního nástroje.
GWPSOF	INT: Číslo vřetena	Deaktivování konstantní obvodové rychlosti brusného kotouče. Pokud číslo vřetena není naprogramováno, bude se deaktivovat obvodová rychlost kotouče pro vřeteno aktivního nástroje.
TMON	INT: T-číslo	Aktivování specifického monitorování pro brusné nástroje Pokud není žádné T-číslo naprogramováno, bude aktivováno monitorování pro aktivní nástroj.
TMOF	INT: T-číslo	Deaktivování monitorování nástroje Pokud není žádné T-číslo naprogramováno, bude deaktivováno monitorování pro aktivní nástroj.

Oddělování třísky					
Identifikátor	**Parametr**				**Vysvětlení**
	1.	**2.**	**3.**	**4.**	
CONTPRON	REAL [,11]: Tabulka kontury	CHAR: Způsob opracování	INT: Počet podříznutí	INT: Status výpočtu	Aktivování referenční přípravy Konturové programy, příp. NC-bloky vyvolávané v následujícím jsou rozděleny do jednotlivých pohybů a uloženy do tabulky kontury. Počet podříznutí se vrací.
CONTDCON	REAL [, 6]: Tabulka kontury	INT: Směr obrábění			Dekódování kontury Bloky kontury se ukládají do tabulky s určitým názvem. Každý řádek tabulky vytvoří jeden blok, aby se ušetřilo místo v paměti.
EXECUTE	INT: Status chyby				Spuštění zpracovávání programu Z režimu editace referenčního bodu nebo po nastavení chráněné oblasti se přepne zpět do normálního režimu zpracovávání programu.

Zpracovávání tabulky		
Identifikátor	**Parametr**	**Vysvětlení**
	1.	
EXECTAB	REAL [11]: Prvek z tabulky pohybů	Spuštění zpracování prvku z tabulky pohybů

Chráněné oblasti						
Identifikátor	**Parametr**					**Vysvětlení**
	1.	**2.**	**3.**	**4.**	**5.**	
CPROTDEF	INT: Číslo chráněné oblasti	BOOL: TRUE: Chráněná oblast orientovaná podle nástroje	INT: 0: 4. a 5. parametr nebudou vyhodnocovány 1: 4. parametr bude vyhodnocován 2: 5. parametr bude vyhodnocován 3: 4. a 5. parametry nebudou vyhodnocovány	REAL : Ohraničení v kladném směru	REAL : Ohraničení v záporném směru	Definice chráněné oblasti pro specifický kanál
NPROTDEF	INT: Číslo chráněné oblasti	BOOL: TRUE: Chráněná oblast orientovaná podle nástroje	INT: 0: 4. a 5. parametr nebudou vyhodnocovány 1: 4. parametr bude vyhodnocován 2: 5. parametr bude vyhodnocován 3: 4. a 5. parametry nebudou vyhodnocovány	REAL : Ohraničení v kladném směru	REAL : Ohraničení v záporném směru	Definice chráněné oblasti pro specifický stroj

Chráněné oblasti						
Identifikátor	**Parametr**					**Vysvětlení**
	1.	**2.**	**3.**	**4.**	**5.**	
CPROT	INT: Číslo chráněné oblasti	INT: Volitelný doplněk 0: Chráněná oblast deaktivována 1: Chráněnou oblast předběžně aktivovat 2: Chráněná oblast aktivována 3: Předběžné aktivování chráněné oblasti s podmíněným zastavování, jen u aktivních chráněných oblastí	REAL : Posunutí chráněné oblasti ve směru první geometrické osy	REAL : Posunutí chráněné oblasti ve směru druhé geometrické osy	REAL : Posunutí chráněné oblasti ve směru třetí geometrické osy	Zapnutí/vypnutí chráněné oblasti pro specifický kanál
NPROT	INT: Číslo chráněné oblasti	INT: Volitelný doplněk 0: Chráněná oblast deaktivována 1: Chráněnou oblast předběžně aktivovat 2: Chráněná oblast aktivována 3: Předběžné aktivování chráněné oblasti s podmíněným zastavování, jen u aktivních chráněných oblastí	REAL : Posunutí chráněné oblasti ve směru první geometrické osy	REAL : Posunutí chráněné oblasti ve směru druhé geometrické osy	REAL : Posunutí chráněné oblasti ve směru třetí geometrické osy	Zapnutí/vypnutí chráněné oblasti pro specifický stroj

Předběžné zpracování/blok po bloku		
Identifikátor	**Parametr**	**Vysvětlení**
STOPRE		Zastavení předběžného zpracování, dokud nejsou zpracovány všechny připravené bloky z hlavního zpracování programu.
SBLOF		Potlačení zpracování blok po bloku
SBLON		Odstranění potlačení zpracování blok po bloku

Přerušení		
Identifikátor	**Parametr**	**Vysvětlení**
	1.	
DISABLE	INT: Číslo vstupu přerušení	Rutina přerušení, která je přiřazena danému hardwarovému vstupu, se přepne do neaktivního stavu. Neprovede se ani rychlé pozvednutí. Přiřazení provedené příkazem SETINT mezi hardwarovým vstupem a rutinou přerušení zůstává zachováno a může být příkazem ENABLE znovu aktivováno.
ENABLE	INT: Číslo vstupu přerušení	Opětovné aktivování přiřazení rutiny přerušení, která byla příkazem DISABLE přepnuta do neaktivního stavu.
CLRINT	INT: Číslo vstupu přerušení	Vymazání přiřazení mezi rutinou přerušení a atributy a vstupem přerušení. Rutina přerušení je tím deaktivována. Vyskytne-li se přerušení, neuskuteční se žádná akce.

Synchronní akce		
Identifikátor	**Parametr**	**Vysvětlení**
	1. – n.	
CANCEL	INT: Číslo synchronní akce	Přerušení modální synchronní akce s uvedeným identifikačním číslem (ID). Může být uveden i větší počet identifikačních čísel - oddělených čárkami.
CANCELSUB		Přerušení aktuální úrovně podprogramu

Definice funkcí					
Identifikátor	**Parametr**				**Vysvětlení**
	1.	**2.**	**3.**	**4.-7.**	
FCTDEF	INT: Číslo funkce	REAL: Dolní mezní hodnota	REAL: Horní mezní hodnota	REAL: Koeficienty a0-a3	Definice polynomické funkce Ty se pak vyhodnocují ve funkcích SYNFCT a PUTFTOCF.

Komunikace				
Identifikátor	**Parametr**			**Vysvětlení**
	1.	**2.**		

Komunikace						
Identifikátor	**Parametr**					**Vysvětlení**
MMC	STRING: Příkaz	CHAR: režim potvrzování*) "N": bez potvrzování "S": synchronní potvrzování "A": asynchronní potvrzování				Příkaz do překladače (Interpreter) HMI pro konfiguraci oken pomocí NC programu.
	1.	**2.**	**3.**	**4.**	**5.**	
DRVPRD	VAR INT: Status	AXIS: Název osy stroje	INT: Číslo parametru pohonu	INT: Index parametru pohonu	VAR REAL: Hodnota parametru pohonu, která má být načtena	Čtení parametrů pohonu na úrovni výrobního programu
DRVPWR	VAR INT: Status	AXIS: Název osy stroje	INT: Číslo parametru pohonu	INT: Index parametru pohonu	REAL: Hodnota parametru pohonu, která se má zapsat	Zápis do parametrů pohonu na úrovni výrobního programu

*) Příkazy jsou potvrzovány na základě žádosti od uvedeného komponentu (kanál, NC systém,...).

Koordinování programů					
Identifikátor	**Parametr**				**Vysvětlení**
INIT	**1.**	**2.**	**3.**		Zvolení určitého NC programu pro zpracovávání v určitém kanálu.
	INT: Číslo kanálu **nebo** Název kanálu z MD20000*)	STRING: Údaj cesty	CHAR: Režim potvrzování **)		
	1. - n.				
START	INT: Číslo kanálu **nebo** Název kanálu z MD20000*)				Spuštění zvoleného programu současně ve více kanálech z momentálně zpracovávaného programu Tento příkaz nemá na svůj vlastní kanál žádný vliv.
WAITE	INT: Číslo kanálu **nebo** Název kanálu z MD20000*)				Čekání na konec programu v jednom nebo ve více jiných kanálech

Koordinování programů			
Identifikátor	**Parametr**		**Vysvětlení**
	1.	**2. - n.**	
WAITM	INT: Číslo značky	INT: Číslo kanálu **nebo** Název kanálu z MD20000*)	Čekání na dosažení značky v uvedených kanálech. Předcházející blok se ukončí s přesným najetím.
WAITMC	INT: Číslo značky	INT: Číslo kanálu **nebo** Název kanálu z MD20000*)	Čekání na dosažení značky v uvedených kanálech. Přesné najetí se uskuteční jen tehdy, pokud v jiných kanálech nebyly značky dosud dosaženy.
	1. - n.		
SETM	INT: Číslo značky		Nastavení jedné nebo více značek pro koordinaci kanálů Zpracování ve vlastním kanálu není touto funkcí nijak ovlivněno.
CLEARM	INT: Číslo značky		Vymazání jedné nebo více značek pro koordinaci kanálů Zpracování ve vlastním kanálu není touto funkcí nijak ovlivněno.
	1. - n.		
WAITP	AXIS: Identifikátor osy		Čekání, dokud uvedená polohovací osa, pro kterou byl předtím naprogramován příkaz POSA, nedosáhne svého naprogramovaného koncového bodu.
WAITS	INT: Číslo vřetena		Čekání, dokud uvedené vřeteno, pro které byl předtím naprogramován příkaz SPOSA, nedosáhne svého naprogramovaného koncového bodu.

Koordinování programů					
Identifikátor	**Parametr**				**Vysvětlení**
RET	**1.**	**2.**	**3.**	**4.**	Konec podprogramu bez výstupu funkcí do PLC. Jestliže je zadán 1. parametr (cíl skoku), uskuteční se skok zpět napřed na blok, který následuje za blokem volání. Potom se cíl vyhledává v závislosti na naprogramovaném příkazu (RET nebo RETB) podle následující strategie: • RET: Vyhledávání směrem ke konci programu. Pokud je vyhledávání neúspěšné, spustí se další hledání, tentokrát směrem k začátku programu. • RETB:: Vyhledávání směrem k začátku programu. Pokud je vyhledávání neúspěšné, spustí se další hledání, tentokrát směrem ke konci programu.
	INT (nebo STRING): Cíl skoku (číslo bloku / značka) pro návrat	INT: 0: Návrat na cíl skoku z 1. parametru > 0: Skok zpět na následující blok	INT: Počet úrovní podprogramu, které mají být přeskočeny	BOOL: Návrat se uskuteční na první blok v hlavním programu.	
RETB	INT (nebo STRING): Cíl skoku (číslo bloku / značka) pro návrat	INT: 0: Návrat na cíl skoku z 1. parametru > 0: Skok zpět na následující blok	INT: Počet úrovní podprogramu, které mají být přeskočeny	BOOL: Návrat se uskuteční na první blok v hlavním programu.	
	1. - n.				
GET	AXIS: Identifikátor osy ***)				Obsazení os stroje Uvedené osy musí být v jiném kanálu uvolněny pomocí příkazu RELEASE.
GETD	AXIS: Identifikátor osy ***)				Přímé obsazení strojních os Uvedené osy **nesmí** být uvolněny pomocí příkazu RELEASE.
RELEASE	AXIS: Identifikátor osy ***)				Uvolnění strojních os
	1.	**2.**	**3.**	**4.**	
PUTFTOC	REAL: Hodnota korekce	INT: Číslo parametru	INT: Číslo kanálu **nebo** Název kanálu z MD20000*)	INT: Číslo vřetena	Změna jemné korekce nástroje
PUTFTOCF	INT: Číslo funkce	VAR REAL: vztažná hodnota	INT: Číslo parametru	INT: Číslo kanálu **nebo** Název kanálu z MD20000*)	Změna jemné korekce nástroje v závislosti na funkci určené příkazem FCTDEF (polynom max. 3. stupně). U příkazu FCTDEF je třeba uvádět zde použité číslo.

Koordinování programů					
Identifikátor	**Parametr**				**Vysvětlení**
AXTOCHAN	**1.**	**2.**	**3. - n.**	**4. - m.**	Předání osy do jiného kanálu
	AXIS: Identifikátor osy	INT: Číslo kanálu **nebo** Název kanálu z MD20000*)	jako 1 ...	jako 2 ...	

*) Namísto čísel kanálů mohou být naprogramovány také názvy kanálů, které jsou definovány pomocí parametru MD20000 $MC_CHAN_NAME.

**) Příkazy jsou potvrzovány na základě žádosti od uvedeného komponentu (kanál, NC systém,...).

***) Na místě osy může být pomocí funkce SPI naprogramováno také vřeteno: např. GET(SPI(1))

Přístup k datům		
Identifikátor	**Parametr**	**Vysvětlení**
CHANDATA	**1.**	Zjištění čísla kanálu za účelem přístupu k datům kanálu (přípustné pouze v inicializačním modulu) Následující přístup se vztahuje na kanál nastavený pomocí příkazu CHANDATA.
	INT: Číslo kanálu	
NEWCONF		Převzetí změněných strojních parametrů

Hlášení			
Identifikátor	**Parametr**		**Vysvětlení**
	1.	**2.**	
MSG	STRING: Hlášení	INT: Zpracování	Na uživatelském rozhraní se ve formě hlášení vypíše libovolný řetězec
WRTPR	STRING: Řetězec znaků	INT: Zpracování	Zápis řetězce znaků do proměnné BTSS

Přístup k souborům						
Identifikátor	**Parametr**					**Vysvětlení**
READ	**1.**	**2.**	**3.**	**4.**	**5.**	Načítání bloků ze systému souborů
	VAR INT: Chyba	CHAR[160]: Název souboru	INT: Počáteční řádek oblasti v souboru, která se má číst	INT: Počet řádků, které se mají načíst	VAR CHAR[255]: Pole proměnných, do něhož jsou načtené informace ukládány	

Přístup k souborům						
Identifikátor	**Parametr**					**Vysvětlení**
WRITE	**1.**	**2.**	**3.**	**4.**		Zápis bloku v systému souborů (nebo na externím zařízení/v souboru)
	VAR INT: Chyba	CHAR[160]: Název souboru	STRING: Externí zařízení/soubor, do kterého se má zapisovat	CHAR[200]: Blok		
DELETE	**1.**	**2.**				Vymazání souboru
	VAR INT: Chyba	CHAR[160]: Název souboru				

Alarmy			
Identifikátor	**Parametr**		**Vysvětlení**
	1.	**2.**	
SETAL	INT: Číslo alarmu (alarmy cyklů)	STRING: Řetězec znaků	Aktivování alarmu Kromě alarmového čísla může být zadán ještě řetězec znaků obsahující až 4 parametry. Budou vám k dispozici následující předdefinované parametry: %1 = číslo kanálu %2 = číslo bloku, návěští %3 = textový index pro alarmy cyklů %4 = doplňkový alarmový parametr

Správa nástrojů				
Identifikátor	**Parametr**			**Vysvětlení**
	1.	**2.**		
DELDL	INT: T-číslo	INT: D-číslo		Vymazání všech součtových korekcí břitu (nebo nástroje, pokud D-číslo není udáno)
DELT	STRING [32]: Identifikátor nástroje	INT: Duplo-číslo		Vymazání nástroje Duplo-číslo může odpadnout.
DELTC	INT: Datový blok číslo n	INT: Datový blok číslo m		Vymazání čísla datového bloku držáku nástroje n až m
DZERO				D-čísla všech nástrojů jednotky TO přiřazené danému kanálu se stanou neplatnými.

Správa nástrojů							
Identifikátor	**Parametr**						**Vysvětlení**
	1.	**2.**	**3.**	**4.**	**5.**	**6.**	
GETFREELOC	VAR INT: Č. zásobníku (výsledná hodnota)	VAR INT: Č. místa (výsledná hodnota)	INT: T-číslo	INT: Č referenčního zásobníku	CHAR: Údaj v závislosti na 4. parametru	INT: Režim rezervace	Vyhledání prázdného místa pro nástroj
	1.	**2.**					
GETSELT	VAR INT: T-číslo (výsledná hodnota)	INT: Číslo vřetena					Poskytuje T-číslo nástroje, který byl předem vybrán pro dané vřeteno
GETEXET	VAR INT: T-číslo (výsledná hodnota)	INT: Číslo vřetena					Poskytuje T-číslo nástroje, který je aktivní z pohledu NC programu
GETTENV	STRING: Název nástrojového prostředí	INT ARRAY[3]: Výsledná hodnota					Načtení T-, D- a DL-čísel uloženy v nástrojovém prostředí
	1.	**2.**	**3.**	**4.**			
POSM	INT: Číslo místa, kam se má umístit	INT: Číslo zásobníku, který se má pohybovat	INT: Číslo místa interního zásobníku	INT: Číslo interního zásobníku			Polohování zásobníku
RESETMON	VAR INT: Status = výsledek operace (výsledná hodnota)	INT: Interní T-číslo	INT: D-číslo nástroje	INT: Volitelný bitově kódovaný parametr			Nastavení skutečné hodnoty nástroje na požadovanou hodnotu
SETDNO	**1.**	**2.**	**3.**				Nastavení čísla korekce (D) břitu nástroje (T)
	INT: T-číslo	INT: Č. břitu	INT: D-číslo				
SETMTH	**1.**						Nastavení čísla držáku nástroje
	INT: Číslo držáku nástroje						

Správa nástrojů							
Identifikátor	**Parametr**						**Vysvětlení**
SETPIECE	**1.**	**2.**					Čítač obrobků vřetena zmenšit o určitou hodnotu Tímto způsobem může uživatel aktualizovat parametry monitorování podle počtu kusů pro nástroje podílející se na obráběcím procesu.
	INT: Hodnota, o kterou se zmenšování provádí	INT: Č. vřetena					
	1.	**2.**	**3.**	**4.**			
SETTA	VAR INT: Status = výsledek operace (výsledná hodnota)	INT: Č. zásobníku	INT: Č. skupiny opotřebení	INT: Podskupina nástrojů			Aktivování nástroje ze skupiny opotřebení
SETTIA	VAR INT: Status = výsledek operace (výsledná hodnota)	INT: Č. zásobníku	INT: Č. skupiny opotřebení	INT: Podskupina nástrojů			Deaktivování nástroje ze skupiny opotřebení
TCA	**1.**	**2.**	**3.**				Volba nástroje / výměna nástroje nezávisle na jeho stavu
	STRING[32]: Identifikátor nástroje	INT: Duplo-číslo	INT: Číslo držáku nástroje				
TCI	**1.**	**2.**					Výměna nástroje ze schránky do zásobníku
	INT: Číslo schránky	INT: Číslo držáku nástroje					
MVTOOL	**1.**	**2.**	**3.**	**4.**	**5.**		Příkaz jazyka pro pohyb nástroje
	INT: Status	INT: Č. zásobníku	INT: Číslo místa	INT: Číslo zásobníku po pohybu	INT: Číslo cílového místa po pohybu		

Orientace nástroje				
Identifikátor	**Parametr**			**Vysvětlení**
	1.	**2.**	**3.**	
ORIRESET	REAL: Základní poloha 1. geometrické osy	REAL: Základní poloha 2. geometrické osy	REAL: Základní poloha 3. geometrické osy	Základní nastavení orientace nástroje

Synchronní vřeteno							
Identifikátor	**Parametr**						**Vysvětlení**
	1.	**2.**	**3.**	**4.**	**5.**	**6.**	
COUPDEF	AXIS: Vlečné vřeteno	AXIS: Řídící vřeteno	REAL: Čitatel převodového poměru	REAL: Jmenovatel převodového poměru	STRING[8]: Chování při přechodu na další blok	STRING[2]: Druh vazby	Definice vazby synchronizovaných vřeten
COUPDEL	AXIS: Vlečné vřeteno	AXIS: Řídící vřeteno					Zrušení vazby synchronních vřeten
COUPRES	AXIS: Vlečné vřeteno	AXIS: Řídící vřeteno					Reset parametrů vazby na hodnoty nastavené v konfiguraci pomocí hodnot MD a SD
COUPON	AXIS: Vlečné vřeteno	AXIS: Řídící vřeteno	REAL: Pozice pro zapnutí vlečného vřetena				Aktivování vazby synchronních vřeten Pokud je pro vlečné vřeteno zadána pozice pro zapnutí (úhlové posunutí mezi FS a LS, které se - absolutně nebo inkrementálně - vztahuje na pozici nula stupňů LS v kladném směru otáčení) potom se vazba aktivuje teprve tehdy, , když byla uvedená pozice přejeta.
COUPONC	AXIS: Vlečné vřeteno	AXIS: Řídící vřeteno					Aktivování vazby synchronních vřeten Pomocí příkazu COUPONC se při aktivování vazby převezmou momentálně platné otáčky vlečného vřetena (M3/M4 S).
COUPOF	AXIS: Vlečné vřeteno	AXIS: Řídící vřeteno	REAL: Pozice pro vypnutí vlečného vřetena (absolutní)	REAL: Pozice pro vypnutí hlavního vřetena (absolutní)			Deaktivování vazby synchronních vřeten Pokud jsou uvedeny pozice, vazba se rozpojí až tehdy, jsou-li všechny uvedené pozice přejety. Vlečné vřeteno se dále otáčí s otáčkami, se kterými se otáčelo těsně před rozpojením vazby.

Synchronní vřeteno							
Identifikátor	**Parametr**						**Vysvětlení**
	1.	**2.**	**3.**	**4.**	**5.**	**6.**	
COUPOFS	AXIS: Vlečné vřeteno	AXIS: Řídící vřeteno	REAL: Pozice pro vypnutí vlečného vřetena (absolutní)				Deaktivování vazby vřeten se zastavením vlečného vřetena Pokud je uvedena pozice, vazba se rozpojí až tehdy, když byla uvedená pozice přejeta.
WAITC	AXIS: Vlečné vřeteno	STRING [8]: Chování při přechodu na další blok	AXIS: Vlečné vřeteno	STRING[8]: Chování při přechodu na další blok			Čekání, až bude pro vřetena (max. 2) splněno kritérium pro změnu bloku vazby. Jestliže chování pro změnu bloku uvedeno, platí chování při přechodech mezi bloky zadané při definici v příkazu COUPDEF.

Elektronická převodovka							
Identifikátor	**Parametr**						**Vysvětlení**
EGDEL	**1.**						Vymazání definice vazby pro vlečnou osu
	AXIS: Vlečná osa						
EGDEF	**1.**	**2. / 4. / 6. / 8. / 10.**	**3. / 5. / 7. / 9. / 11.**				Definice elektronické převodovky
	AXIS: Vlečná osa	AXIS: Řídící osa	INT: Druh vazby				
EGON	**1.**	**2.**	**3. / 6. / 9. / 12. / 15.**	**4. / 7. / 10. / 13. / 16.**	**5. / 8. / 11. / 14. / 17.**		Aktivování elektronické převodovky bez synchronizace
	AXIS: Vlečná osa	STRING: Chování při změně mezi bloky	AXIS: Řídící osa	REAL: Čitatel faktoru vazby	REAL: Jmenovatel faktoru vazby		

Elektronická převodovka									
Identifikátor	**Parametr**								**Vysvětlení**
EGONSYN	**1.**	**2.**	**3.**	**4. / 8. / 12. / 16. / 20.**	**5. / 9. / 13. / 17. / 21.**	**6. / 10. / 14. / 18. / 22.**	**7. / 11. / 15. / 19. / 23.**		Aktivování elektronické převodovky se synchronizací
	AXIS: Vlečná osa	STRING: Chování při změně mezi bloky	REAL: Synchronní pozice vlečné osy	AXIS: Řídící osa	REAL: Synchronní pozice řídící osy	REAL: Čitatel faktoru vazby	REAL: Jmenovatel faktoru vazby		
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.**	Aktivování elektronické převodovky se synchronizací a zadání režimu najíždění
	AXIS: Vlečná osa	STRING: Chování při změně mezi bloky	REAL: Synchronní pozice vlečné osy	STRING: Režim najíždění	AXIS: Řídící osa	REAL: Synchronní pozice řídící osy	REAL: Čitatel faktoru vazby	REAL: Jmenovatel faktoru vazby	
EGOFS	**1.**	**2. - n.**							Selektivní vypnutí elektronické převodovky
	AXIS: Vlečná osa	AXIS: Řídící osa							
EGOFC	**1.**								Deaktivování elektronické převodovky (varianta jen pro vřetena)
	AXIS: Vlečné vřeteno								

Prostřihování					
Identifikátor	**Parametr**				**Vysvětlení**
	1.	**2.**	**3.**	**4.**	
PUNCHAAC	REAL: Nejmenší vzdálenost od díry	REAL: Počáteční zrychlení	REAL: Největší vzdálenost od díry	REAL: Koncové zrychlení	Aktivování zrychlení závisejícího na dráze

Informační funkce v pasivním systému souborů				
Identifikátor	**Parametr**			**Vysvětlení**
	1.	**2.**	**3.**	
FILEDATE	VAR INT: Chybové hlášení	CHAR[160]: Název souboru	VAR CHAR[8]: Datum ve formátu "dd.mm.rr"	Poskytuje datum posledního přístupu do souboru za účelem zápisu
FILETIME	VAR INT: Chybové hlášení	CHAR[160]: Název souboru	VAR CHAR[8]: Hodinový čas ve formátu "hh.mm.ss"	Poskytuje přesný čas posledního přístupu do souboru za účelem zápisu
FILESIZE	VAR INT: Chybové hlášení	CHAR[160]: Název souboru	VAR INT: Velikost souboru	Poskytuje údaj o momentální velikosti souboru
FILESTAT	VAR INT: Chybové hlášení	CHAR[160]: Název souboru	VAR CHAR[5]: Datum ve formátu "rwxsd"	Poskytuje informace o stavu souboru, které se týkají následujících oprávnění: • Čtení (r: read) • Zápis (w: write) • Spustit (x: execute) • Zobrazit (s: show) • Vymazat (d: delete)
FILEINFO	VAR INT: Chybové hlášení	CHAR[160]: Název souboru	VAR CHAR[32]: Datum ve formátu "rwxsd nnnnnnnn dd.mm.rr hh:mm:ss"	Pro daný soubor poskytuje součet všech informací, které je možné načíst pomocí funkcí FILEDATE, FILETIME, FILESIZE a FILESTAT.

Osový zásobník		
Identifikátor	**Parametr**	**Vysvětlení**
	1. - n.	
AXCTSWE	AXIS: Osový zásobník	Otočení osového zásobníku
AXCTSWED	AXIS: Osový zásobník	Otočení osového zásobníku (varianta příkazu pro uvádění do provozu!)
AXCTSWEC:	AXIS: Osový zásobník	Uvolnění pro reset otočení osového zásobníku

Vazba master-slave		
Identifikátor	**Parametr**	**Vysvětlení**
	1. - n.	
MASLON	AXIS: Identifikátor osy	Aktivování vazby typu master/slave
MASLOF	AXIS: Identifikátor osy	Zrušení vazby typu master/slave
MASLOFS	AXIS: Identifikátor osy	Zrušení vazby Master/Slave a automatické zabrždění vřeten typu Slave
MASLDEF	AXIS: Identifikátor osy	Definice vazby Master/Slave Poslední osa je osou typu Master.
MASLDEL	AXIS: Identifikátor osy	Zrušení vazby Master/Slave a vymazání definice tohoto spojení

On-line korekce délky nástroje			
Identifikátor	**Parametr**		**Vysvětlení**
	1.	**2.**	
TOFFON	AXIS: Směr korekce	REAL: Hodnota offsetu ve směru korekce	Aktivování on-line korekce délky nástroje v zadaném směru korekce
TOFFOF	AXIS: Směr korekce		Zrušení on-line korekce délky nástroje v zadaném směru korekce

SERUPRO		
Identifikátor	**Parametr**	**Vysvětlení**
IPTRLOCK		Začátek úseku programu pro vyhledávání
IPTRUNLOCK		Konec úseku programu pro vyhledávání

Návratová dráha		
Identifikátor	**Parametr**	**Vysvětlení**
	1. - n.	
POLFMASK	AXIS: Identifikátor geometrické, příp. strojní osy	Uvolnění os pro rychlý zpětný pohyb (bez souvislostí mezi jednotlivými osami)
POLFMLIN	AXIS: Identifikátor geometrické, příp. strojní osy	Uvolnění os pro lineární rychlý zpětný pohyb

Návratová dráha				
Identifikátor	**Parametr**			**Vysvětlení**
POLFA	**1.**	**2.**	**3.**	Pozice pro návrat pro samostatnou osu
	AXIS: Identifikátor kanálové osy	INT: Typ	REAL: Hodnota	

Prevence kolize			
Identifikátor	**Parametr**		**Vysvětlení**
	1.		
PROTA	STRING: "R"		Vyžádání nového výpočtu protikolizního modelu
PROTS	**1.**	**2. - n.**	Nastavení stavu chráněné oblasti
	CHAR: Status	STRING: Název chráněné oblasti	

Inteligentní přizpůsobování zátěže					
Identifikátor	**Parametr**				**Vysvětlení**
CADAPTON	**1.**	**2.**	**3.**	**4.**	Aktivování přizpůsobování zátěže
	INT: Status	AXIS: Název osy stroje	INT: vstupní veličina	REAL: Vstupní hodnota (volitelné)	
CADAPTOF	**1.**	**2.**	**3.**		Deaktivování vazby přizpůsobování zátěže
	INT: Status	AXIS: Název osy stroje	INT: vstupní veličina		

Přizpůsobení ryvu		
Identifikátor	**Parametr**	**Vysvětlení**
CALCFIR		Přizpůsobení filtru ryvu FIR režimu dynamiky Procedura CALCFIR se používá v cyklu výrobce CUST_832.
AFISON		Aktivování automatického přepínání filtračního řetězce
AFISOF		Deaktivování automatického přepínání filtračního řetězce

5.5 Předdefinované procedury v synchronních akcích

Následující předdefinované procedury jsou Vám k dispozici pouze v synchronních akcích.

Synchronní procedury		
Identifikátor	**Parametr**	**Vysvětlení**
STOPREOF		Odblokování zastavení předběžného zpracování Synchronní akce s příkazem STOPREOF způsobí zastavení předběžného zpracování po následujícím bloku na řídící jednotce. Zastavení předběžného zpracování bude zrušeno na konci výstupního bloku nebo když bude splněna podmínka příkazu STOPREOF. Všechny příkazy synchronizované akce pak budou s příkazem STOPREOF platit jako zpracované.
RDISABLE		Zablokování načítání
DELDTG	**1.** AXIS: Osa pro vymazání axiální zbytkové dráhy (volba). Pokud osa odpadne, vymazání zbytkové vzdálenosti se provede pro dráhu.	Vymazání zbytkové dráhy Synchronní akce s příkazem DELDTG způsobí zastavení předběžného zpracování po následujícím bloku na řídící jednotce. Zastavení preprocesoru bude zrušeno na konci výstupního bloku nebo když bude splněna první podmínka příkazu DELDTG. V $AA_DELT[<osa>] se nachází osová vzdálenost k cíli při axiálním vymazání zbytkové dráhy, v $AC_DELT zbytková dráha.

Koordinování programů, technologické cykly		
Identifikátor	**Parametr**	**Vysvětlení**
	1.	
LOCK	INT: Identifikační číslo synchronní akce, která má být zablokována	Zastavení synchronní akce s identifikací ID, příp. zastavení technologického cyklu Může být naprogramováno jedno nebo více identifikačních čísel.
UNLOCK	INT: Identifikační číslo synchronní akce, která má být uvolněna	Odblokování synchronní akce s identifikací ID, příp. pokračování technologického cyklu Může být naprogramováno jedno nebo více identifikačních čísel.
ICYCON		Každý blok technologického cyklu za příkazem ICYCON zpracovávat v samostatném taktu IPO
ICYCOF		Všechny bloky technologického cyklu za příkazem ICYCOF zpracovávat v taktu IPO

Polynomické funkce						
Identifikátor	**Parametr**					**Vysvětlení**
SYNFCT	**1.**	**2.**	**3.**			Jestliže v synchronní pohybové akci je splněna podmínka, na vstupní proměnné se vyhodnotí polynom stanovený prvním výrazem. Horní a dolní hranice rozsahu hodnot jsou omezeny a vstupní proměnné jsou přiřazeny.
	INT: Číslo polynomické funkce, která byla definována pomocí FCTDEF.	VAR REAL: Proměnná pro výsledek *)	VAR REAL: Vstupní proměnná **)			
FTOC	**1.**	**2.**	**3.**	**4.**	**5.**	Změna jemné korekce nástroje v závislosti na funkci určené příkazem FCTDEF (polynom max. 3. stupně). V příkazu FCTDEF musí být uvedeno zde použité číslo.
	INT: Číslo polynomické funkce, která byla definována pomocí FCTDEF.	VAR REAL: Vstupní proměnná **)	INT: Délka 1, 2, 3	INT: Číslo kanálu	INT: Číslo vřetena	

*) Jako proměnné výsledku jsou přípustné pouze speciální systémové proměnné (viz Příručka k funkcím, Synchronní akce).

**) Jako vstupní proměnné jsou přípustné pouze speciální systémové proměnné (viz Příručka k funkcím, Synchronní akce).

5.6 Předem definované funkce

Prostřednictvím volání předem definované funkce se spouští zpracování předem definované funkce NC systému, které narozdíl od předem definovaných procedur poskytují nějakou výslednou hodnotu. Volání předem definované funkce se může nacházet i jako operand ve výrazu.

Souřadný systém						
Identifikátor	**Výsledná hodnota**	**Parametr**				**Vysvětlení**
		1.	**2.**	**3. - 15.**	**4. - 16.**	
CTRANS	FRAME	AXIS: Identifikátor osy	REAL : Posunutí	AXIS: Identifikátor osy	REAL: Posunutí	Translace: Hrubé posunutí počátku pro více os.
CFINE	FRAME	AXIS: Identifikátor osy	REAL : Posunutí	AXIS: Identifikátor osy	REAL: Posunutí	Translace: Jemné posunutí počátku pro více os.
CSCALE	FRAME	AXIS: Identifikátor osy	REAL: Faktor měřítka	AXIS: Identifikátor osy	REAL: Faktor měřítka	Scale: Faktor změny měřítka pro více os

Souřadný systém						
Identifikátor	**Výsledná hodnota**	**Parametr**				**Vysvětlení**
		1.	**2.**	**3. a 5.**	**4. a 6.**	
CROT	FRAME	AXIS: Identifikátor osy	REAL : Otočení	AXIS: Identifikátor osy	REAL: Otočení	Rotace: Otočení aktuálního souřadného systému Maximální počet parametrů: 6 (jeden identifikátor osy a jedna hodnota na každou geometrickou osu).
CROTS	FRAME	AXIS: Identifikátor osy	REAL : Otáčení o prostorový úhel	AXIS: Identifikátor osy	REAL: Otáčení o prostorový úhel	Rotace: Otočení aktuálního souřadného systému o prostorový úhel Maximální počet parametrů: 6 (jeden identifikátor osy a jedna hodnota na každou geometrickou osu).
CMIRROR		**1.**	**2. - 8.**			Mirror: Zrcadlové převrácení souřadné osy.
	FRAME	AXIS	AXIS			
		1.	**2.**			
CRPL	FRAME	INT: Otočná osa	REAL: Úhel otočení			Otočení framu v libovolné rovině
ADDFRAME	INT: 0: OK 1: Údaj cíle (String) je nesprávný 2: Cílový frame není konfigurován 3: Otáčení framu není dovoleno	FRAME Aditivní změřený nebo vypočítaný FRAME	STRING: Specifikuje cílový frame			Vypočítá cílový frame, který je specifikován v řetězci String. Cílový frame se vypočítá tak, že se nový celkový frame vypočítá jako zřetězení starého celkového framu s předávaným framem.
INVFRAME	FRAME	**1.**				Výpočet inverzního framu z daného framu Zřetězením jakéhokoli framu s jeho invertovaným framem má vždy za následek nulový frame.
		FRAME				

Souřadný systém						
Identifikátor	**Výsledná hodnota**	**Parametr**				**Vysvětlení**
MEAFRAME	FRAME	**1.**	**2.**	**3.**		Výpočet framu na základě 3 změřených bodů v prostoru.
		REAL[3,3]: Souřadnice změřeného bodu v prostoru	REAL[3,3]: Souřadnice požadovaného bodu	VAR REAL: Proměnná, pomocí které systém vrací informace o kvalitě výpočtu proměnné typu FRAME.		

Geometrické funkce					
Identifikátor	**Výsledná hodnota**	**Parametr**			**Vysvětlení**
		1.	**2.**	**3.**	
CALCDAT	BOOL: Chybový stav	VAR REAL [n, 2]: Tabulka (abscisa, ordináta) bodů 1 až n	INT: Počet bodů	VAR REAL [3]: Výsledek: Abscisa, ordináta a rádius vypočteného středu kruhu	Tato funkce vypočítá souřadnice středu a rádius kruhu ze 3 nebo 4 bodů. Body musí být různé.
INTERSEC	BOOL: Chybový stav	VAR REAL [11]: První prvek kontury:	VAR REAL [11]: Druhý prvek kontury:	VAR REAL [2]: Výsledný vektor k souřadnicím průsečíku: abscisa a ordináta	Vypočítá souřadnice průsečíku mezi dvěma konturovými prvky. Chybový stav ukazuje, zda byl průsečík nalezen.

Funkce tabulek křivek								
Identifikátor	**Výsledná hodnota**	**Parametr**						**Vysvětlení**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
CTAB	REAL: Pozice vlečné osy	REAL: Pozice řídící osy	INT: Číslo tabulky	VAR REAL[ ]: Výsledné stoupání	AXIS: Vlečná osa pro nastavení měřítka	AXIS: Řídící osa pro nastavení měřítka		Zjišťuje polohu vlečné osy k uvedené poloze řídící osy z tabulky křivky. Pokud nejsou naprogramovány parametry 4/5, bude se počítat se standardním měřítkem.
CTABINV	REAL: Pozice řídící osy	REAL: Pozice vlečné osy	REAL: Řídící poloha	INT: Číslo tabulky	VAR REAL[ ]: Výsledné stoupání	AXIS: Vlečná osa pro nastavení měřítka	AXIS: Řídící osa pro nastavení měřítka	Zjišťuje polohu řídící osy k uvedené poloze vlečné osy z tabulky křivky. Pokud nejsou naprogramovány parametry 5/6, bude se počítat se standardním měřítkem.

Funkce tabulek křivek								
Identifikátor	**Výsledná hodnota**	**Parametr**						**Vysvětlení**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
CTABID	INT: Číslo tabulky křivky	INT: Číslo položky v paměti	STRING: Místo v paměti: "SRAM", "DRAM"					Zjišťuje číslo tabulky křivky, která je v paměti uložena pod zadaným číslem.
CTABISLOCK	INT: Stav blokování	INT: Číslo tabulky						Zjišťuje stav blokování tabulky křivky: > 0: Tabulka je blokována 1: CTABLOCK 2: aktivní vazba 3: CTABLOCK a aktivní vazba 0: Tabulka není blokována -1: Tabulka neexistuje
CTABEXISTS	INT: Existence	INT: Číslo tabulky						Zjišťuje, zda tabulka křivky existuje ve statické nebo dynamické paměti NC systému: 0: FALSE 1: TRUE
CTABMEMTYP	INT: Místo uložení	INT: Číslo tabulky						Zjišťuje místo uložení tabulky křivky: 1: DRAM 0: SRAM -1: Tabulka neexistuje
CTABPERIOD	INT: Periodicita	INT: Číslo tabulky						Zjišťuje periodicitu tabulky křivky 0: není periodická 1: periodická v řídící ose 2: periodická v řídící a vlečné ose -1: Tabulka neexistuje
CTABNO	INT: Počet tabulek křivek							Zjišťuje počet definovaných tabulek křivek (ve statické a dynamické paměti NC systému)
CTABNOMEM	INT: Počet tabulek křivek	STRING: Místo v paměti: "SRAM", "DRAM"						Zjišťuje počet definovaných tabulek křivek v zadané paměti

Funkce tabulek křivek								
Identifikátor	**Výsledná hodnota**	**Parametr**						**Vysvětlení**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
CTABFNO	INT: Počet tabulek	STRING: Místo v paměti: "SRAM", "DRAM"						Zjišťuje počet tabulek křivek v zadané paměti, které ještě mohou být definovány
CTABSEG	INT: Počet křivkových segmentů	STRING: Místo v paměti: "SRAM", "DRAM"	STRING: Druh segmentu: "L": Lineární "P": Polynom					Zjišťuje počet použitých křivkových segmentů zadaného druhu v dané paměti >=0: Počet -1: Typ paměti je neplatný Pokud parametr 2 není naprogramován, výsledkem bude součet lineárních a polynomických segmentů.
CTABFSEG	INT: Počet křivkových segmentů	STRING: Místo v paměti: "SRAM", "DRAM"	STRING: Druh segmentu: "L": Lineární "P": Polynom					Zjišťuje počet ještě možných úseků křivky odpovídajících zadanému druhu segmentu, které mohou být uloženy v zadané paměti >=0: Počet -1: Typ paměti je neplatný
CTABSEGID	INT: Počet křivkových segmentů	INT: Číslo tabulky	STRING: Druh segmentu: "L": Lineární "P": Polynom					Zjišťuje počet křivkových úseků odpovídajících zadanému druhu úseku, které jsou využívány danou tabulkou křivky >=0: Počet -1: Tabulka neexistuje
CTABMSEG	INT: Počet křivkových segmentů	STRING: Místo v paměti: "SRAM", "DRAM"	STRING: Druh segmentu: "L": Lineární "P": Polynom					Zjišťuje maximální možný počet úseků křivky odpovídajících zadanému druhu segmentu, které mohou být uloženy v zadané paměti >=0: Počet -1: Tabulka neexistuje
CTABPOL	INT: Počet křivkových polynomů	STRING: Místo v paměti: "SRAM", "DRAM"						Zjišťuje počet používaných křivkových polynomů v zadané paměti >=0: Počet -1: Tabulka neexistuje

Funkce tabulek křivek								
Identifikátor	**Výsledná hodnota**	**Parametr**						**Vysvětlení**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
CTABPOLID	INT: Počet křivkových polynomů	INT: Číslo tabulky						Zjišťuje počet křivkových polynomů, které jsou využívány tabulkou křivky >=0: Počet -1: Tabulka neexistuje
CTABFPOL	INT: Počet křivkových polynomů	STRING: Místo v paměti: "SRAM", "DRAM"						Zjišťuje počet maximálně možných křivkových polynomů v zadané paměti: >=0: Počet -1: Tabulka neexistuje
CTABMPOL	INT: Počet křivkových polynomů	STRING: Místo v paměti: "SRAM", "DRAM"						Zjišťuje počet maximálně možných křivkových polynomů v zadané paměti: >=0: Počet -1: Tabulka neexistuje
CTABSSV	REAL: Pozice vlečné osy	REAL: Pozice řídící osy	INT: Číslo tabulky	VAR REAL[ ]: Výsledné stoupání	AXIS: Vlečná osa pro nastavení měřítka	AXIS: Řídící osa pro nastavení měřítka		Zjišťuje polohu vlečné osy na začátku úseku křivky, který patří k zadané hodnotě řídící osy.
CTABSEV	REAL: Pozice vlečné osy	REAL: Pozice řídící osy	INT: Číslo tabulky	VAR REAL[ ]: Výsledné stoupání	AXIS: Vlečná osa pro nastavení měřítka	AXIS: Řídící osa pro nastavení měřítka		Zjišťuje polohu vlečné osy na konci úseku křivky, který patří k zadané hodnotě řídící osy.
CTABTSV	REAL: Pozice vlečné osy	INT: Číslo tabulky	VAR REAL[ ]: Výsledné stoupání na začátku tabulky	AXIS: Vlečná osa				Zjišťuje polohu vlečné osy na začátku tabulky křivky.
CTABTEV	REAL: Pozice vlečné osy	INT: Číslo tabulky	VAR REAL[ ]: Výsledné stoupání na konci tabulky	AXIS: Vlečná osa				Zjišťuje polohu vlečné osy na konci tabulky křivky.
CTABTSP	REAL: Pozice řídící osy	INT: Číslo tabulky	VAR REAL[ ]: Výsledné stoupání na začátku tabulky	AXIS: Řídící osa				Zjišťuje polohu řídící osy na začátku tabulky křivky.

Funkce tabulek křivek								
Identifikátor	**Výsledná hodnota**	**Parametr**						**Vysvětlení**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
CTABTEP	REAL: Pozice řídící osy	INT: Číslo tabulky	VAR REAL[ ]: Výsledné stoupání na konci tabulky	AXIS: Řídící osa				Zjišťuje polohu řídící osy na konci tabulky křivky.
CTABTMIN	REAL: Minimální hodnota	INT: Číslo tabulky	REAL: Spodní mezní hodnota intervalu řídících hodnot	REAL: Horní mezní hodnota intervalu řídících hodnot	AXIS: Vlečná osa	AXIS: Řídící osa		Zjišťuje minimální hodnotu vlečné osy v celém definičním rozsahu tabulky křivky nebo v definovaném intervalu.
CTABTMAX	REAL: Maximální hodnota	INT: Číslo tabulky	REAL: Spodní mezní hodnota intervalu řídících hodnot	REAL: Horní mezní hodnota intervalu řídících hodnot	AXIS: Vlečná osa	AXIS: Řídící osa		Zjišťuje maximální hodnotu vlečné osy v celém definičním rozsahu tabulky křivky nebo v definovaném intervalu.
Upozornění: Funkce tabulky křivky mohou být programovány i v synchronních akcích.								

Funkce pro osy						
Identifikátor	**Výsledná hodnota**	**Parametr**				
		1.	**2.**	**3.**	**4.**	**Vysvětlení**
AXNAME	AXIS: Identifikátor osy	STRING [ ]: Vstupní řetězec				Konvertovaný vstupní řetězec v identifikátoru osy
AXSTRING	STRING[ ]: Název osy	AXIS: Identifikátor osy				Převod identifikátoru osy na řetězec znaků.
ISAXIS	BOOL: Osa je k dispozici (TRUE) nebo není (FALSE)	INT: Číslo geometrické osy (1 až 3)				Provádí kontrolu, jestli geometrická osa 1 až 3 zadaná jako parametr této funkce je k dispozici v souladu se strojním parametrem MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB.
SPI	AXIS: Identifikátor osy	INT: Číslo vřetena				Převedení čísla vřetena na identifikátor osy
AXTOSPI	INT: Číslo vřetena	AXIS: Identifikátor osy				Převedení identifikátoru osy na číslo vřetena

Funkce pro osy						
Identifikátor	**Výsledná hodnota**	**Parametr**				
		1.	**2.**	**3.**	**4.**	**Vysvětlení**
MODAXVAL	REAL: Hodnota funkce modulo	AXIS: Identifikátor osy	REAL: Poloha osy			Výpočet zbytku modulo ze zadané polohy osy Pokud uvedená osa není osou typu modulo, bude výsledkem nezměněná poloha osy.
POSRANGE	BOOL: Požadovaná poloha se nachází uvnitř intervalu poloh (TRUE) nebo nenachází (FALSE)	AXIS: Identifikátor osy	REAL: Referenční poloha v souřadném systému	REAL: Šířka intervalu hodnot polohy	INT: Souřadný systém	Zjištění, zda se požadovaná poloha osy, jejíž interpolace právě probíhá, nachází v okně okolo předem zadané referenční pozice.

Správa nástrojů					
Identifikátor	**Výsledná hodnota**	**Parametr**			**Vysvětlení**
		1.	**2.**	**3.**	
CHKDM	INT: Status Výsledek zkoušky	INT: Číslo zásobníku	INT: D-číslo		Kontrola jednoznačnosti D-čísla v rámci zásobníku
CHKDNO	INT: Status Výsledek zkoušky	INT: T-číslo 1. nástroje	INT: T-číslo 2. nástroje	INT: D-číslo	Kontrola jednoznačnosti D-čísla
GETACTT	INT: Status	INT: T-číslo	STRING[32]: Název nástroje		Zjišťuje aktivní nástroj ze skupiny stejnojmenných nástrojů
GETACTTD	INT: Status Výsledek zkoušky	VAR INT: Nalezené T-číslo (výsledná hodnota)	INT: D-číslo		Zjišťuje odpovídající T-číslo k absolutnímu D-číslu
GETDNO	INT: D-číslo	INT: T-číslo	INT: Číslo břitu		Zjišťuje D-číslo břitu nástroje T
GETT	INT: T-číslo	STRING[32]: Název nástroje	INT: Duplo-číslo		Zjišťuje T-číslo k zadanému názvu nástroje
NEWT	INT: T-číslo	STRING[32]: Název nástroje	INT: Duplo-číslo		Založení nového nástroje (připravte si parametry nástroje) Duplo-číslo může odpadnout.
TOOLENV	INT: Status	STRING: Název			Uloží prostředí nástroje se zadanými názvy do statické paměti NC systému

Správa nástrojů					
Identifikátor	**Výsledná hodnota**	**Parametr**			**Vysvětlení**
		1.	**2.**	**3.**	
DELTOOLENV	INT: Status	STRING: Název			Vymaže prostředí nástroje se zadanými názvy ze statické paměti NC systému Vymazání všech prostředí nástroje, pokud nebyl uveden žádný název.
GETTENV	INT: Status	STRING: Název	VAR INT: T-číslo [0] D-číslo [1] DL-číslo [2]		Zjišťuje T-číslo, D-číslo a DL-číslo z okolí nástroje s uvedeným názvem

Aritmetika					
Identifikátor	**Výsledná hodnota**	**Parametr**			**Vysvětlení**
		1.	**2.**	**3.**	
SIN	REAL	REAL			sinus
ASIN	REAL	REAL			arkus sinus
COS	REAL	REAL			kosinus
ACOS	REAL	REAL			arkus kosinus
TAN	REAL	REAL			tangens
ATAN2	REAL	REAL	REAL		Arcus-Tangens 2
SQRT	REAL	REAL			druhá odmocnina
POT	REAL	REAL			Druhá mocnina
TRUNC	REAL	REAL			Celočíselná složka
ROUND	REAL	REAL			Zaokrouhlování
ROUNDUP	REAL	REAL			Zaokrouhlování
ABS	REAL	REAL			Absolutní hodnota
LN	REAL	REAL			Přirozený logaritmus
EXP	REAL	REAL			Exponenciální funkce e^x
MINVAL	REAL	REAL	REAL		Zjištění menší hodnoty ze dvou parametrů
MAXVAL	REAL	REAL	REAL		Zjištění větší hodnoty ze dvou parametrů
BOUND	REAL: Stav zkoušky	REAL: Spodní mezní hodnota	REAL: Horní mezní hodnota	REAL: Porovnávaná hodnota	Zjišťuje, zda se porovnávaná hodnota nachází v rámci mezních hodnot.
Upozornění: Aritmetické funkce mohou být programovány i v synchronních akcích. Výpočet, příp. vyhodnocování těchto aritmetických funkcí se pak provádějí v hlavním zpracování. Pro výpočty a jako pomocná paměť se mohou využívat také parametry synchronních akcí $AC_PARAM[<n>].					

Funkce pro práci s řetězci					
Identifikátor	**Výsledná hodnota**	**Parametr**			**Vysvětlení**
		1.	**2.**	**3.**	
ISNUMBER	BOOL	STRING: Vstupní řetězec			Kontroluje, zda vstupní řetězec může být převeden na číslo.
NUMBER	REAL	STRING: Vstupní řetězec			Převod vstupního řetězce na číslo.
TOUPPER	STRING	STRING: Vstupní řetězec			Převádí vstupní řetězec na velká písmena
TOLOWER	STRING	STRING: Vstupní řetězec			Převádí vstupní řetězec na malá písmena
STRLEN	INT	STRING: Vstupní řetězec			Zjišťuje délku vstupního řetězce až do jeho konce (/0).
INDEX	INT	STRING: Vstupní řetězec	CHAR: Hledaný znak		Zjišťuje pozici znaku ve vstupním řetězci směrem zleva doprava. 1. znak řetězce zleva má index 0.
RINDEX	INT	STRING: Vstupní řetězec	CHAR: Hledaný znak		Zjišťuje pozici znaku ve vstupním řetězci směrem zprava doleva. 1. znak řetězce zprava má index 0.
MINDEX	INT	STRING: Vstupní řetězec	STRING: Hledaný znak		Zjišťuje pozici znaku zadaného ve 2. parametru ve vstupním řetězci směrem zleva doprava. 1. znak vstupního řetězce zleva má index 0.
SUBSTR	STRING	STRING: Vstupní řetězec	INT	INT	Zjišťuje dílčí řetězec ze vstupního řetězce, který je popsán počátečním znakem (2. parametr) a počtem znaků (3. parametr)
SPRINT	STRING	STRING: Vstupní řetězec			Zjišťuje naformátovaný vstupní řetězec

Funkce pro měřicí cykly								
Identifikátor	**Výsledná hodnota**	**Parametr**						**Vysvětlení**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
CALCPOSI	INT: Status	REAL[3]: Výchozí pozice ve WCS	REAL[3]: Inkrementální zadání cesty vztažené na výchozí pozici	REAL[5]: Minimální vzdálenosti k monitorovaným mezním hodnotám	REAL[3]: Pole pro výslednou možnou inkrementální dráhu	BOOL: Přepočítávání systémů rozměrových jednotek ano/ne	INT: Druh monitorování mezních hodnot	Kontroluje, zda mohou geometrické osy počínaje ze zadaného počátečního bodu urazit zadanou dráhu, aniž by došlo k narušení mezních hodnot pro osu Pro případ, že požadovanou dráhu nelze bez narušení hranic objet, vrací se maximální přípustná hodnota.
GETTCOR	INT: Status	REAL [11]:	STRING: Složky délky nástroje: Souřadný systém	STRING: Název nástrojového prostředí	INT: Interní T-číslo nástroje	INT: Číslo břitu (D-číslo) nástroje	INT: Číslo korekce závislé na místě (DL-číslo nástroje)	Zjišťuje délky nástroje a složky délky nástroje z okolí nástroje, příp. z aktuálního okolí.
LENTOAX	INT: Status	INT[3]: Přiřazení os geometrickým osám	REAL[3]: Matice pro zobrazení délek nástroje v souřadném systému	STRING: Souřadný systém pro přiřazení				Zjišťuje informace o přiřazení délek nástroje L1, L2, L3 aktivního nástroje abscise, ordinátě a aplikátě. Přiřazení geometrickým osám je ovlivňováno framy a aktivní rovinou (G17-G19).

SETTCOR	INT: Status	**1.**	**2.**	**3.**	**4.**	**5.**	**6.**	**7.**	**8.**	**9.**	
		REAL [3]: Korekční vektor v prostoru	STR.: Identifikátor složek	INT: ke kor. Komponenty 0 - 11	INT: Druh operace zápisu 0 - 3	INT: Index geometrické osy	STRING: Název prostředí nástroje	INT: interní T-číslo nástroje	INT: D-číslo nástroje	INT: DL-číslo nástroje	Změna složek nástroje při zohledňování všech okrajových podmínek, které se podílejí na vyhodnocování jednotlivých složek

Speciální funkce								
Identifikátor	**Výsledná hodnota**	**Parametr**						**Vysvětlení**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
STRINGIS	INT: Informace o řetězci	STRING: Název kontrolovaného prvku						Kontrola, zda je zadaný řetězec k dispozici jako prvek aktuálního rozsahu programovacího jazyka NC systému.
ISVAR	BOOL: Proměnná je známa ano/ne	STRING: Název proměnné						Kontrola, zda předávaný parametr obsahuje proměnnou, která je NC systému známá (strojní parametr, nastavovaný parametr, systémová proměnná, obecná proměnná jako např. GUD)
GETVARTYP	INT: Datový typ	STRING: Název proměnné						Zjišťuje datový typ systémové/uživatelské proměnné
GETVARPHU	INT: Číselná hodnota fyzikální jednotky	STRING: Název proměnné						Zjišťuje fyzikální jednotky systémové/uživatelské proměnné
GETVARAP	INT: Úroveň ochrany proti přístupu	STRING: Název proměnné	STRING: Druh přístupu					Zjišťuje přístupová oprávnění k systémové/uživatelské proměnné
GETVARLIM	INT: Status	STRING: Název proměnné	CHAR: Udává, která mezní hodnota má být načtena.	VAR REAL: Výsledná mezní hodnota				Zjišťuje dolní/horní mezní hodnotu systémové/uživatelské proměnné
GETVARDFT	INT: Status	STRING: Název proměnné	VAR REAL/ STRING/ FRAME: Výsledkem je standardní hodnota	INT: Index na první rozměr (není nutné)	INT: Index na druhý rozměr (není nutné)	INT: Index na třetí rozměr (není nutné)		Zjišťuje standardní hodnotu systémové/uživatelské proměnné
COLLPAIR	INT: Výsledek zkoušky	STRING: Název 1. chráněné oblasti	STRING: Název 2. chráněné oblasti	BOOL: Potlačení alarmů (nepovinné)				Kontroluje příslušnost k dvojici oblastí chráněných kvůli kolizi

Speciální funkce								
Identifikátor	**Výsledná hodnota**	**Parametr**						**Vysvětlení**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
PROTD	REAL: Vzdálenost dvou chráněných oblastí	STRING: Název 1. chráněné oblasti	STRING: Název 2. chráněné oblasti	VAR REAL: Výsledná hodnota: 3-rozměrný vektor vzdálenosti	BOOL: Systém měřicích jednotek pro vzdálenost a vektor vzdálenosti (nepovinný)			Zjišťuje vzdálenost mezi dvěma zadanými chráněnými oblastmi
DELOBJ	INT: Číslo chyby	STRING: Typ komponentu, který má být vymazán	INT: Počáteční index komponent, které mají být vymazány (nepovinné)	INT: Koncový index komponent, které mají být vymazány (nepovinné)	BOOL: Potlačení alarmů (nepovinné)			Mazání prvků z kinematických řetězců, chráněných oblastí, prvků chráněných oblastí, dvojic oblastí chráněných kvůli kolizi a dat transformací
NAMETOINT	INT: Index systémové proměnné	STRING: Název pole systémových proměnných	STRING: Řetězec znaků/ název	BOOL: Potlačení alarmů (nepovinné)				Zjištění příslušného indexu systémové proměnné na základě řetězce znaků
ORISOLH	INT: Číslo chyby	INT: Ovládá chování funkce	REAL: První úhel	REAL: Druhý úhel				Tato funkce pomáhá uživateli nastavovat polohy kruhových stroje tak, aby byl soustružnický nástroj přestaven do definované polohy vzhledem k obrobku, a to nezávisle na kinematice stroje. Předpoklad: Je aktivována 6-osá transformace, jejíž parametry jsou nastaveny prostřednictvím kinematického řetězce.
CORRTRAFO	INT: Číslo chyby	REAL: Vektor korekce	INT: Prvek, který má být upraven	INT: Režim korekce	BOOL: Potlačení alarmů (nepovinné)			Změna offsetových vektorů nebo směrových vektorů orientačních os v kinematickém modelu stroje.
CORRTC	INT: Číslo chyby	REAL: Vektor korekce	INT: Prvek, který má být upraven	INT: Režim korekce	BOOL: Potlačení alarmů (nepovinné)			Modifikace vektorů offsetů nebo směrových vektorů orientovatelného držáku nástroje po měření stroje.

5.7 Aktuální jazyk v HMI

Následující tabulka obsahuje všechny jazyky, které jsou na uživatelském rozhraní k dispozici.

Momentálně nastavený jazyk je možné ve výrobním programu a v synchronních akcích zjistit pomocí následující systémové proměnné:

$AN_LANGUAGE_ON_HMI = <hodnota>

<hodnota>	Jazyk	Zkratka jazyka
1	Němčina (Německo)	DEU
2	Francouzština	FRA
3	Angličtina (Spojené království)	ENG
4	Španělština	ESP
6	Italština	ITA
7	Holandština	NLD
8	Čínština (zjednodušená)	CHS
9	Švédština	SVE
18	Maďarština	HUN
19	Finština	FIN
28	Čeština	CSY
50	Portugalština (Brazílie)	PTB
53	Polština	PLK
55	Dánština	DAN
57	Ruština	RUS
68	Slovenština	SKY
72	Rumunština	ROM
80	Čínština (tradiční)	CHT
85	Korejština	KOR
87	Japonština	JPN
89	Turečtina	TRK

Poznámka

Aktualizace proměnné $AN_LANGUAGE_ON_HMI se uskutečňuje:

- po náběhu systému
- po resetu NC systémů a/nebo PLC
- po přepnutí na jiný NC systém v rámci M2N
- po přepnutí jazyka na HMI

Přílohy

A

A.1 Seznam zkratek

A	
A	Výstup
AFIS	Automatické přepínání filtračního řetězce Automatic Filter Switch:
ASCII	American Standard Code for Information Interchange: Americká norma pro kódy při výměně informací.
ASIC	Application Specific Integrated Circuit: Integrovaný obvod pro specifickou aplikaci
ASUP	Asynchronní podprogram
AUTO	Provozní režim "Automatic" (Automaticky)
AUXFU	Auxiliary Function: Pomocná funkce
AWL	Seznam příkazů

B	
BAG	Skupiny provozních režimů
BCD	Binary Coded Decimals: Desítková čísla vyjádřená v binárním kódu.
BICO	Binector Connector
BIN	Binary Files: Binární soubory
BKS	Základní souřadný systém
BM	Provozní hlášení
BO	Binector Output
BTSS	Rozhraní ovládacího panelu

C	
CLC	Regulace vzdálenosti
CNC	Computerized Numerical Control: Computerunterstützte numerische Steuerung
COM	Communication
CP	Communication Processor
CPU	Central Processing Unit: Centrální procesorová jednotka
CST	Configured Stop: Konfigurovatelné zastavení

D	
DB	Datový modul (PLC)
DBB	Byte datového modulu (PLC)
DBD	Dvojité slovo datového modulu (PLC)
DBW	Slovo datového modulu (PLC)
DBX	Bit datového modulu (PLC)

D	
DDS	Drive Data Set: Datový blok pohonu
DIR	Directory: Adresář
DO	Drive Object
DRF	Differential Resolver Function: Funkce diferenčního otočného snímače (ruční kolečko)
DRY	Dry Run: Posuv při zkušebním zpracování
DW	Datové slovo
DWORD	Dvojité slovo (momentálně 32 bitů)

E	
E	Vstup
EES	Execution from External Storage
E/A	Vstupy / výstupy
ESR	Rozšířené zastavování a odjíždění
ETC	Tlačítko ETC ">" pro vyvolání rozšíření pruhu programových tlačítek v tomtéž menu

F	
FB	Funkční modul (PLC)
FC	Function Call: Funkční modul (PLC)
FDD	Feed Disable: Zablokování posuvu
FdStop	Feed Stop: Zastavení posuvu
FIFO	First In First Out: Paměť, která pracuje bez zadávání adres. Data, která jsou do ní uložená, jsou čtena ve stejné posloupnosti, v jaké byla uložena.
FM	Chybové hlášení
FUP	Funkční schéma (metoda programování pro PLC)
FW	Firmware

G	
GEO	Geometrie, např. geometrická osa
GP	Základní program (PLC)
GUD	Global User Data: Globální uživatelská data

H	
HEX	Zkratka pro hexadecimální formát
HiFu	Pomocná funkce
HMI	Human Machine Interface: Uživatelské rozhraní systému SINUMERIK
HSA	Pohon hlavního vřetena
HT	Handheld Terminal
HW	Hardware

I	
IBN	Uvádění do provozu
INC	Increment: Velikost kroku
INI	Initializing Data: Inicializační data
IPO	Interpolator

J	
JOG	Jogging: Seřizovací režim

K	
KOP	Kontaktní schéma (metoda programování pro PLC)

L	
LED	Light Emitting Diode: Světelná dioda
LMS	Systém pro měření polohy
LR	Regulátor polohy

M	
Main	Main program: Hlavní program (OB1, PLC)
MCP	Machine Control Panel: Řídící panel stroje
MD	Strojní parametr, příp. strojní parametry
MDA	Manual Data Automatic: Manuální zadávání
MDS	Motor Data Set: Datový blok motoru
MELDW	Slovo hlášení
MKS	Souřadný systém stroje
MM	Motor Module
MPF	Main Program File: Hlavní program (NC)
MPI	Multi Point Interface
MSTT	Řídící panel stroje

N	
NC	Numerical Control: Numerický řídící systém pro přípravu bloků, řízením posuvů atd.
NCU	Numerical Control Unit: Hardwarová jednotka NC systému
NCK	Numerical Control Kernel
NCSD	NC Start Disable
NST	Signál rozhraní
NV	Posunutí počátku
NX	Numerical Extension: Rozšiřovací modul osy

O	
OB	Organizační modul v PLC
OEM	Original Equipment Manufacturer
OP	Operation Panel: Zařízení pro obsluhu systému

P	
PCU	PC Unit: skříň PC (výpočetní jednotka)
PG	Programovací přístroj
PLC	Programmable Logic Control: Programovatelné logické řízení
PN	PROFINET
PO	Power On
POS	Poloha/pozice
PPO	Objekt parametrů a procesních dat; cyklický datový telegram při komunikaci po sběrnici PROFIBUS DP a s profilem "Pohony s měnitelnými otáčkami"
PPU	Panel Processing Unit (centrální hardware panelově orientovaného řídícího CNC systému, jako je např. SINUMERIK 828D)
PROFIBUS	Process Field Bus: Sériová datová sběrnice
PRT	Testování programu
PTP	Point to Point: od bodu k bodu
PZD	Data procesů: Složka procesních dat PPO

R	
REF	Funkce najíždění na referenční bod
REPOS	Funkce najíždění na původní polohu
RESU	Retrace support
RID	Read In Disable
RP	R-Parametry, početní parametry, předem definované uživatelské proměnné

S	
SA	Synchronní akce
SBL	Single Block: Zpracování blok po bloku
SBT	Safe Brake Test
SCC	Safety Control Channel
SCL	Structured Control Language
SD	Settingdatum, příp. nastavované parametry
SDI	Safe Direction
SERUPRO	Search-Run by Program Test: Vyhledávání bloku pomocí zkušebního zpracování programu
SIC	Safety Info Channel
SKP	Skip: Funkce pro přeskakování bloků výrobního programu
SLP	Safe Limited Position

S	
SLS	Safely Limited Speed
SMI	Sensor Module Integrated
SOS	Safe Operating Stop
SPF	Sub Program File: Podprogram (NC)
SS1	Safe Stop 1
SS2	Safe Stop 2
STO	Safe Torque Off
STW	Řídící slovo
SUG	Obvodová rychlost kotouče
SW	Software

T	
TCU	Thin Client Unit
TM	Terminal Module (SINAMICS)
TO	Tool Offset: Korekční parametry nástroje
TOA	Tool Offset Active: Označení (typ souboru) pro korekční parametry nástroje
TOFF	On-line korekce délky nástroje
TRANSMIT	Transform Milling Into Turning: Transformace souřadného systému pro obrábění frézováním na soustruhu

U	
UP	Podprogram
USB	Universal Serial Bus

V	
VDI	Interní komunikační rozhraní mezi NC systémem a PLC

W	
WKS	Souřadný systém obrobku
WPD	Work Piece Directory: Adresář obrobku
WZ	Nástroj
WZV	Správa nástrojů

Z	
ZSW	Stavové slovo (pohonu)

Rejstřík

$

$A_PROBE, 620, 627
$A_PROBE_LIMITED, 627
$AA_ACC, 128
$AA_ATOL, 853
$AA_COUP_ACT
    při vlečení, 908
    v případě axiální vazby řídící hodnotou, 930
$AA_ESR_ENABLE, 988
$AA_FGREF, 113
$AA_FGROUP, 113
$AA_G0MODE, 184
$AA_LEAD_SP, 930
$AA_LEAD_SV, 930
$AA_MM, 621
$AA_MM1...4, 628
$AA_MW, 621
$AA_MW1...4, 628
$AC_ACT_PROG_NET_TIME, 995
$AC_ACTUAL_PARTS, 998
$AC_AXCTSWA, 894
$AC_AXCTSWE, 895
$AC_CTOL, 853
$AC_CTOL_G0_ABS, 187
$AC_CUT_INV, 809
$AC_CUTMOD, 809
$AC_CUTMOD_ANG, 809
$AC_CUTMODK, 809
$AC_CUTTING_TIME, 995
$AC_CYCLE_TIME, 995
$AC_DELAYFST, 863
$AC_ESR_TRIGGER, 988
$AC_F_TYPE, 144
$AC_FGROUP_MASK, 113
$AC_FZ, 144
$AC_MEA, 620, 627
$AC_OLD_PROG_NET_TIME, 995
$AC_OLD_PROG_NET_TIME_COUNT, 995
$AC_OPERATING_TIME, 995
$AC_OTOL, 853
$AC_OTOL_ G0_ABS, 188
$AC_PROG_NET_TIME_TRIGGER, 996
$AC_REPOS_PATH_MODE, 872
$AC_REQUIRED_PARTS, 998
$AC_S_TYPE, 98
$AC_SMAXVELO, 875
$AC_SMAXVELO_INFO, 875
$AC_SPECIAL_PARTS, 998
$AC_STOLF, 187
$AC_SVC, 98
$AC_TOFF, 88
$AC_TOFFCR, 88
$AC_TOFFL, 88
$AC_TOFFR, 88
$AC_TOTAL_PARTS, 998
$AC_TRAFO_CORR_ELEM_P, 740
$AC_TRAFO_CORR_ELEM_T, 740
$AC_TRAFO_ORIAX_LOC, 741
$AN_AXCTAS, 895
$AN_AXCTSWA, 894
$AN_ESR_TRIGGER, 988
$AN_LANGUAGE_ON_HMI, 1320
$AN_POWERON_TIME, 995
$AN_SETUP_TIME, 995
$NT_CLOSE_CHAIN_T, 741
$NT_CNTRL, 740
$NT_CORR_ELEM_P, 740
$NT_CORR_ELEM_T, 740
$NT_NAME, 733
$NT_ROT_AX_NAME, 806
$NT_TRAFO_INDEX, 733
$P_ACTBFRAME, 657
$P_AD, 808
$P_AEP, 284
$P_APDV, 284
$P_APR, 284
$P_BFRAME, 657
$P_CTOL, 853
$P_CTOL_ G0_ABS, 188
$P_CUT_INV, 809
$P_CUTMOD, 809
$P_CUTMOD_ANG, 809
$P_CUTMOD_ERR, 810
$P_CUTMODK, 809
$P_DELAYFST, 863
$P_F_TYPE, 144
$P_FGROUP_MASK, 114
$P_FZ, 144
$P_GWPS, 105
$P_CHBFRAME, 657
$P_CHBFRMASK, 658
$P_IFRAME, 658
$P_IS_EES_PATH, 567
$P_NCBFRAME, 657
$P_NCBFRMASK, 658
$P_ORI_DIFF, 801

$P_ORI_POS, 801
$P_ORI_SOL, 802
$P_ORI_STAT, 805
$P_OTOL, 853
$P_OTOL_ G0_ABS, 188
$P_PATH, 566
$P_PFRAME, 658
$P_PROG, 566
$P_PROGPATH, 566
$P_S_TYPE, 99
$P_SIM, 630, 904, 1000
$P_STACK, 566
$P_STOLF, 187
$P_SUBPAR, 498
$P_SVC, 99
$P_TOFF, 88
$P_TOFFCR, 88
$P_TOFFL, 88
$P_TOFFR, 88
$P_TOOLENV, 818
$P_TOOLENVN, 818
$P_WORKAREA_CS_COORD_SYSTEM, 356
$P_WORKAREA_CS_LIMIT_MINUS, 357
$P_WORKAREA_CS_LIMIT_PLUS, 356
$P_WORKAREA_CS_MINUS_ENABLE, 356
$P_WORKAREA_CS_PLUS_ENABLE, 356
$PA_ATOL, 853
$PA_FGREF, 113
$PA_FGROUP, 114
$SA_LEAD_TYPE, 929
$SC_PA_ACTIV_IMMED, 586
$SN_PA_ACTIV_IMMED, 586
$TC_CARR_CORR_ELEM, 793
$TC_CARR1...14, 781
$TC_CARR13, 788
$TC_CARR14, 788
$TC_CARR18...23, 781
$TC_CARR18[m], 785
$TC_DP1 ... 25, 742
$TC_TP_MAX_VELO, 95

*****

* (matematická funkce), 442

/

/ (matematická funkce), 442

+

+ (matematická funkce), 442

<

< (relační operátor), 444
<< (operátor zřetězení), 451
<= (relační operátor), 444
<> (relační operátor), 444

=

== (relační operátor), 444

>

> (relační operátor), 444
>= (relační operátor), 444

A

ABS, 442
Absolutní rozměry, 32
AC, 151
ACC, 127
ACCLIMA, 846
ACN, 158
ACOS, 442
ACP, 158
ACTBLOCNO, 510
ACTFRAME, 636
ADIS, 298
ADISPOS, 298
ADISPOSA, 631
Adresa
  Přiřazování hodnot, 48
Adresová písmena, 1251
Adresování, 560
Adresy, 380
Adresy OEM, 629
AFISOF, 856
AFISON, 856
Alarmy
  nastavení v NC programu, 1014
ALF
  pro rychlé pozvednutí od kontury, 551
  pro rychlý zpětný pohyb v průběhu řezání závitu, 233

AMIRROR, 330
AND, 444
ANG, 213
ANG1, 213
ANG2, 213
AP, 176
Aplikace „Siemens Industry Online Support", 25
APR, 413
APRB, 413
APRP, 413
APW, 413
APWB, 413
APWP, 413
AR
  Programování kruhu, 196
AROT, 318
AROTS, 324
Array, 419
AS, 490
ASCALE, 327
ASIN, 442
A-Spline, 605
ASPLINE, 598
Asynchronní oscilační pohyby, 965
ATAN2, 442
ATOL, 849
ATRANS, 312
Atributy polohy
  nepřímé programování, 438
Automatické rozdělení dráhy, 981
Automatický ukazatel vyhledávání, 865
AV, 937
AX, 887
AXCTSWE, 893
AXCTSWEC, 893
AXCTSWED, 893
AXIS, 397
AXNAME, 449
AXSTRING, 887
AXTOCHAN, 885
AXTOSPI, 887

B

B_AND, 444
B_NOT, 444
B_OR, 444
B_XOR, 444
BAUTO, 598
BCS, 40
BFRAME, 636
Binární konstanta, 385
BLOCK, 535
blok
  -číslo, 47
  -délka, 48
  -konec, 47
  -konec LF, 53
  Posloupnost příkazů, 48
Blok, 45
  přeskakovat, 50
Blok zastavení, 865
Bloková platnost, 47
Blokovací moment
  - Pevný doraz, 366
BLSYNC, 546
BNAT, 598
BNS, 41
Bod/úhel najíždění, 266
BOOL, 397
BOUND, 425
BRISK, 843
BRISKA, 843
Brusné nástroje, 74
Břity
  -číslo, 81
  -počet nástrojů pro obrábění kontury, 290
  -poloha, 70
  -rádius, 70
  -specifická poloha, 294
  -střed, 70
  -vztažný bod, 294
B-Spline, 606
BSPLINE, 598
BTAN, 598

C

CAC, 597
CACN, 597
CACP, 597
CADAPTOF, 897
CADAPTON, 897
CALL, 534
CALLPATH, 538
CASE, 468
CDC, 597
CDOF, 287
CDOF2, 287
CDON, 287
Cesta do adresáře, 562
Cesta pro vyhledávání
  Programovatelná cesta pro vyhledávání, 538
  při volání podprogramu, 565

CFC, 132
CFIN, 132
CFINE, 645
CFTCP, 132
CIC, 597
CIP, 199
CLEARM, 484
CLRINT, 548
COARSE, 937
COARSEA, 631
COMPCAD, 610
COMPCURV, 610
COMPLETE, 567
COMPOF, 610
COMPON, 610
COMPSURF, 610
CONTDCON, 1028
CONTPRON, 1023
CORROF, 339
CORRTC, 790
CORRTRAFO, 734
COS, 442
COUPDEF, 937
COUPDEL, 937
COUPOF, 937
COUPOFS, 937
COUPON, 937
COUPONC, 937
COUPRES, 937
CP, 713
CPBC, 949
CPDEF, 948
CPDEL, 948
CPFMOF, 951
CPFMON, 951
CPFMSON, 950
CPFPOS + CPOF, 951
CPFPOS + CPON, 949
CPFRS, 948
CPLA, 948
CPLCTID, 948
CPLDEF, 948
CPLDEL, 948
CPLDEN, 948
CPLINSC, 953
CPLINTR, 953
CPLNUM, 948
CPLOF, 948
CPLON, 948
CPLOUTSC, 953
CPLOUTTR, 953
CPLPOS, 949
CPLSETVAL, 948
CPMALARM, 954
CPMBRAKE, 954
CPMPRT, 953
CPMRESET, 952
CPMSTART, 953
CPMVDI, 954
CPOF, 948
CPON, 948
CPRECOF, 854
CPRECON, 854
CPROT, 583
CPROTDEF, 580
CPSETTYPE, 954
CPSYNCOP, 953
CPSYNCOP2, 953
CPSYNCOV, 954
CPSYNFIP, 953
CPSYNFIP2, 953
CPSYNFIV, 954
CR, 194
CROTS, 324
C-Spline, 607
CSPLINE, 598
CT, 202
CTAB, 919
CTABDEF, 909
CTABDEL, 915
CTABEND, 909
CTABEXISTS, 915
CTABFNO, 923
CTABFPOL, 923
CTABFSEG, 923
CTABID, 918
CTABINV, 919
CTABISLOCK, 918
CTABLOCK, 916
CTABMEMTYP, 918
CTABMPOL, 923
CTABMSEG, 923
CTABNO, 923
CTABNOMEM, 923
CTABPERIOD, 918
CTABPOL, 923
CTABPOLID, 923
CTABSEG, 923
CTABSEGID, 923
CTABSEV, 919
CTABSSV, 919
CTABTEP, 919
CTABTEV, 919
CTABTMAX, 919

CTABTMIN, 919
CTABTSP, 919
CTABTSV, 919
CTABUNLOCK, 916
CTOL, 849
CTOLG0, 184
CTRANS, 645
CUT2D, 289
CUT2DD, 289
CUT2DF, 289
CUT2DFD, 289
CUT3DC, 757
CUT3DCC, 767
CUT3DCCD, 767
CUT3DCD, 757
CUT3DF, 761
CUT3DFD, 761
CUT3DFF, 761
CUT3DFS, 761
CUTCONOF, 292
CUTCONON, 292
CUTMOD, 806
CUTMODK, 806
CYCLE4071
externí programování, 1146
CYCLE4072
externí programování, 1148
CYCLE4073
externí programování, 1152
CYCLE4074
externí programování, 1153
CYCLE4075
externí programování, 1156
CYCLE4077
externí programování, 1159
CYCLE4078
externí programování, 1163
CYCLE4079
externí programování, 1165
CYCLE435 - Nastavení souřadného systému orovnávače
externí programování, 1103
CYCLE495 - Profilování
externí programování, 1103
CYCLE60 - Gravírování
externí programování, 1055
CYCLE61 - Rovinné frézování
externí programování, 1057
CYCLE62 - Volání kontury
externí programování, 1059
CYCLE63 - Frézování konturové kapsy / odstranění zbytkového materiálu z konturové kapsy / frézování konturového čepu / odstranění zbytkového materiálu z konturového čepu
externí programování, 1060
CYCLE64 - Předvrtání konturové kapsy
externí programování, 1062
CYCLE70 - Frézování závitu
externí programování, 1064
CYCLE72 - Frézování po dráze
externí programování, 1066
CYCLE76 - Pravoúhlý čep
externí programování, 1070
CYCLE77 - Kruhový čep
externí programování, 1072
CYCLE78 - Frézování vrtaných závitů
externí programování, 1074
CYCLE782 - Přizpůsobení zatížení
externí programování, 1105
CYCLE79 - Mnohohran
externí programování, 1076
CYCLE800 - Naklápění roviny / naklápění nástroje / nastavování orientace nástroje
externí programování, 1108
CYCLE801 - Polohovací vzor Mřížka nebo Obdélník
externí programování, 1111
CYCLE802 - Libovolné pozice
externí programování, 1113
CYCLE806 - Interpolační soustružení
externí programování, 1117
CYCLE81 - Navrtávání středicích důlků
externí programování, 1078
CYCLE82 - Vrtání
externí programování, 1079
CYCLE83 - Vrtání hlubokých děr 1
externí programování, 1082
CYCLE830 - Vrtání hlubokých děr 2
externí programování, 1117
CYCLE832 - Vysokorychlostní obrábění
externí programování, 1123
CYCLE84 - Vrtání závitů bez vyrovnávací hlavičky
externí programování, 1085
CYCLE840 - Vrtání závitů s vyrovnávací hlavičkou
externí programování, 1127
CYCLE85 - Vystružování
externí programování, 1088
CYCLE86 - Vyvrtávání
externí programování, 1089
CYCLE899 - Otevřená drážka
externí programování, 1129
CYCLE92 - Upichování
externí programování, 1091

CYCLE930 - Zápich
externí programování, 1132
CYCLE940 - Odlehčovací zápich tvaru E a F / závitový zápich
externí programování, 1135
CYCLE95 - Oddělování třísky od kontury
externí programování, 1092
CYCLE951 - Oddělování třísky
externí programování, 1137
CYCLE952 - Oddělování třísky / odstraňování zbytkového materiálu oddělováním třísky / zapichování / zapichování zbytkového materiálu / soustružení zápichu / odstraňování zbytkového materiálu soustružením zápichu
externí programování, 1140
CYCLE98 - Řetězec závitů
externí programování, 1094
CYCLE99 - Soustružení závitu
externí programování, 1098
Čelní frézování, 680
Číslo břitu, 778
Čtení/zápis do parametrů pohonu, 902
CHAN, 397
CHANDATA, 567
CHAR, 397
CHF, 247
CHKDNO, 778
CHR, 247
Chráněné oblasti, 580

D

D..., 80
D0, 80
DAC, 166
DC, 158
D-čísla
přejmenovat, 779
zkontrolovat, 778
D-číslo
volné zadávání, 778
DEF, 397
DEFAULT, 468
DEFINE ... AS, 490
Definice surového obrobku, 1016
DELAYFSTOF, 861
DELAYFSTON, 861
DELDL, 745
DELETE, 573
DELOBJ, 725
DELTOOLENV, 816
Desítková konstanta, 384
DIACYCOFA, 166
DIAM90, 164
DIAM90A, 166
DIAMCYCOF, 164
DIAMCHAN, 166
DIAMCHANA, 166
DIAMOF, 164
DIAMOFA, 166
DIAMON, 164
DIAMONA, 166
DIC, 166
DILF, 233
DIN 66217, 38
DISABLE, 548
DISC, 270
DISCL, 273
DISPLOF, 510
DISPLON, 510
DISPR, 866
DISR, 273
DISRP, 273
DITE, 231
DITS, 231
DIV, 442
DL, 744
DO, 964
Doba prodlevy, 368
Doba zpracování, 995
-chování řídících struktur, 477
Doraz, 36
Dráhové osy, 374
Drážková pila, 79
DRFOF, 341
DRIVE, 843
DRIVEA, 843
Druh vazby, 940
DRVPRD, 902
DRVPWR, 902
Držák nástroje
- Orientovatelný, 786
-kinematika, 781
-vztažný bod, 37
DV, 937
DYNFINISH, 847
DYNNORM, 847
DYNPOS, 847
DYNPREC, 847
DYNROUGH, 847
DYNSEMIFIN, 847

E

Easy XML, 1005
EAUTO, 598
EES, 558
EG
    Elektronická převodovka, 930
EGDEF, 931
EGDEL, 936
EGOFC, 935
EGOFS, 935
EGON, 932
EGONSYN, 932
EGONSYNE, 932
Elektronická převodovka, 930
ELSE, 478, 964
ENABLE, 548
ENAT, 598
ENDFOR, 481
ENDIF, 478
ENDLABEL, 470
ENDLOOP, 480
ENDWHILE, 482
ENS, 41
ESR, 987
ESRR, 993
ESRS, 992
ETAN, 598
Eulerův úhel, 677
EVERY, 964
Evolventa, 208
EXECSTRING, 440
EXECTAB, 1033
EXECUTE, 1036
EXP, 442
EXTCALL
    pro SINUMERIK 840D sl, 539
EXTCLOSE, 1010
EXTERN, 528
Externí posunutí počátku, 646
externí programování, 1167
EXTOPEN, 1010

F

F...
    při posuvu, 106
    při přímkové interpolaci, 188
    při řezání závitů G34 G35, 230
FA, 123
FAD, 273
Faktor změny měřítka, 327
FALSE, 397
Faseta, 247
FB, 138
FCTDEF, 753
FCUB, 839
FD, 129
FDA, 129
FENDNORM, 631
FFWOF, 849
FFWON, 849
FGREF, 106
FGROUP, 106
FIFOCTRL, 859
FILEDATE, 577
FILEINFO, 577
FILESIZE, 577
FILESTAT, 577
FILETIME, 577
Filtrační řetězce pro požadovanou hodnotu polohy, 856
FINE, 937
FINEA, 631
FL, 106
FLIM, 877
FLIN, 839
FMA, 135
FNORM, 839
FOR, 481
Formát děrné pásky, 45
FP, 359
FPO, 839
FPR, 123
FPRAOF, 123
FPRAON, 123
Frame
    - změna měřítka, 327
    - Zrcadlové převrácení, 330
    - zřetězení, 660
    deaktivování, 338
    -příkazy, 309
    Vyvolat, 642
FRAME, 397
Framy, 307
    globální, 654
    Kanálové, 655
    Přiřadit, 643
    Řetězce framů, 644
    Systémové, 656
FRC, 247
FRCM, 247

frézovací nástroje, 71
Frézování konturové kapsy / odstranění zbytkového materiálu z konturové kapsy / frézování konturového čepu / odstranění zbytkového materiálu z konturového čepu – CYCLE63
    externí programování, 1060
Frézování po dráze - CYCLE72
    externí programování, 1066
Frézování vrtaných závitů - CYCLE78
    externí programování, 1074
Frézování závitu - CYCLE70
    externí programování, 1064
FROM, 964
FTOCOF, 756
FTOCON, 756
Funkce
    předem definovaná, 1307
Funkce Look Ahead, 303
Funkce OEM, 629

G

G1, 188
G110, 175
G111, 175
G112, 175
G140, 273
G141, 273
G142, 273
G143, 273
G147, 273
G148, 273
G153
    při deaktivování framu, 338
    při posunutí počátku (nuly), 144
G17, 147
G18, 147
G19, 147
G2, 191
G247, 273
G248, 273
G25
    Ohraničení pracovního pole, 352
    Omezení otáček vřetena, 105
G26
    Ohraničení pracovního pole, 352
    Omezení otáček vřetena, 105
G290, 1020
G291, 1020
G3, 191
G33, 223
G335, 237
G336, 237
G34, 230
G340, 273
G341, 273
G347, 273
G348, 273
G35, 230
G4, 368
G40, 253
G41, 253
G42, 253
G450, 270
G451, 270
G460, 284
G461, 284
G462, 284
G5, 709
G500
    při posunutí počátku (nuly), 144
G505 ... G599, 144
G53
    při deaktivování framu, 338
    při posunutí počátku (nuly), 144
G54 ... G57, 144
G58, 316
G59, 316
G60, 296
G601, 296
G602, 296
G603, 296
G62, 631
G621, 631
G64, 298
G641, 298
G642, 298
G643, 298
G644, 298
G645, 298
G7, 709
G70, 160
G700, 160
G71, 160
G710, 160
G74, 358
G75, 359
G810 ... G819, 630
G820 ... G829, 630
G9, 296
G90, 151
G91, 153
G93, 106
G94, 106

G95, 106
G96, 99
G961, 99
G962, 99
G97, 99
G971, 99
G972, 99
G973, 99
GEOAX, 889
Geometrická osa
    přepínání, 889
Geometrie
    -osy, 371
GET, 881
GETACTTD, 779
GETD, 881
GETDNO, 779
GETTCOR, 818
GETTENV, 817
GETVARAP, 429
GETVARDFT, 431
GETVARDIM, 431
GETVARLIM, 430
GETVARPHU, 428
GETVARTYP, 433
GFRAME0 ... GFRAME100, 343
G-kód
    nepřímé programování, 437
Globální paměť výrobních programů (GDIR), 558
GOTO, 465
GOTOB, 465
GOTOC, 465
GOTOF, 465
GOTOS, 464
GP, 438
Gravírování - CYCLE60
    externí programování, 1055
GROUP_ADDEND - konec doplňkového bloku najíždění
    externí programování, 1168
GROUP_BEGIN - Začátek programového bloku
    externí programování, 1167
GROUP_END - konec programového bloku
    externí programování, 1168
G-skupina
    Technologie, 847
GUD, 398
GWPSOF, 104
GWPSON, 104

H

Hexadecimální konstanta, 385
Hlášení, 349
Hlavní položka, 169
Hlavní směrnice pro ochranu dat, 26
Hloubka vnoření
    řídících struktur, 477
Hloubka zajíždění nástroje, 760
Hodnota S
    Interpretace, 91
HOLES1 - Polohovací vzor Přímka
    externí programování, 1039
HOLES2 - Polohovací vzor Kružnice nebo Kruhový oblouk
    externí programování, 1040

I

I...
    při kruhové interpolaci, 191
    při řezání závitů G33, 223
    při řezání závitů G34 G35, 230
IC, 153
ID, 964
Identifikátor, 44
IDS, 964
IF, 478
IFRAME, 636
INDEX, 453
Index pole, 421
INICF, 397
Inicializace
    polí, 419
Inicializační program, 568
INIPO, 397
INIRE, 397
INIT, 484
INITIAL, 567
INITIAL_INI, 567
Inkrementální rozměry, 34
INT, 397
Internetové stránky třetích stran, 21
Interpolace vektoru otočení, 692
Interpolační soustružení - CYCLE806
    externí programování, 1117
INTERSEC, 1032
INVCCW, 208
INVCW, 208
IPOBRKA, 631
IPOENDA, 631
IPOSTOP, 937
IPTRLOCK, 864
IPTRUNLOCK, 864
IR, 237

ISAXIS, 887
ISFILE, 576
ISNUMBER, 449
ISOCALL, 537
ISVAR, 426

J

J...
při kruhové interpolaci, 191
při řezání závitů G34 G35, 230
JERKLIM, 876
JERKLIMA, 846
JR, 237

K

K...
při kruhové interpolaci, 191
při řezání závitů G33, 223
při řezání závitů G34 G35, 230
Kanál
-osy, 373
Kartézské souřadnice, 30
Kinematika
S více stupni volnosti, 785
Koeficienty polynomu, 612
Komentáře, 49
Koncový bod, 172
Konec doplňkového bloku najíždění - GROUP_ADDEND
externí programování, 1168
Konec programového bloku - GROUP_END
externí programování, 1168
Konstanta typu INTEGER, 384
Konstanty, 384
Konstanty typu REAL, 384
KONT, 263
KONTC, 263
KONTT, 263
Kontura
-kódování, 1028
najíždění/odjíždění, 263
opětovné najetí, 866
-počítač, 214
-prvek, 172
-příprava, 1023
-tabulka, 1023
Konturový prvek
odjíždění, 1033
Korekce
Délka nástroje, 68
Rádius nástroje-, 69
-rovina, 291
Korekce rádiusu nástroje
CUT2DF, 291
na vnějších rozích, 270
Zpoždění v rozích, 630
Korekční parametry nástroje
aditivní, 744
-Offset, 84
Paměť korekcí, 741
Souřadný systém pro hodnoty opotřebení, 749
KR, 237
Kritérium konce pohybu
programovatelné, 631
Kruhová drážka - SLOT2
externí programování, 1050
Kruhová interpolace
pomocí vnitřního bodu a koncového bodu, 199
Spirální interpolace, 206
Kruhová kapsa - POCKET4
externí programování, 1045
Kruhové osy
Směrové vektory, 781
Úhel natočení, 781
Vektory vzdálenosti, 781
Kruhový čep - CYCLE77
externí programování, 1072
Kuželové závity, 229

L

L..., 526
Label, 470
LEAD, 674
LEADOF, 925
LEADON, 925
LENTOAX, 836
Levý závit, 225
LF, 47
LFOF, 233
LFON, 233
LFPOS, 233
LFTXT, 233
LFWP, 233
Libovolné pozice - CYCLE802
externí programování, 1113
LIFTFAST, 549
LIMS, 99
LINE FEED (konec řádku), 47
Linkové
- proměnné, 395
Lisování, 976

LLI, 410
LN, 442
Logické operátory, 444
LONGHOLE - Podlouhlá díra
externí programování, 1053
LOOP, 480
LUD, 398

M

M..., 346
M0, 346
M1, 346
M17, 513
M19
M-funkce, 346
při polohování vřetena, 118
M2, 346
M3, 89
M30, 513
M4, 89
M40, 346
M41, 346
M42, 346
M43, 346
M44, 346
M45, 346
M5, 89
M6, 66
M70, 118
Makro, 490
MASLDEF, 961
MASLDEL, 961
MASLOF, 961
MASLOFS, 961
MASLON, 961
MATCH, 453
MAXVAL, 425
MCALL, 532
MCS, 38
MD10010, 484
MD10210, 790
MD10240, 162
MD10260, 160
MD10280, 484
MD10651, 238
MD10710, 233
MD15800, 395
MD18104, 815
MD18116, 816
MD18156, 395
MD20360, 822
MD24558, 823
MD24658, 823
MD9440, 82
MEAC, 621
MEAFRAME, 651
MEAS, 618
MEASA, 621
MEAW, 618
MEAWA, 621
M-funkce, 346
MINDEX, 453
MINVAL, 425
MIRROR, 330
MMC, 1005
Mnohohran - CYCLE79
externí programování, 1076
MOD, 442
Modální platnost, 47
MODAXVAL, 887
Monitorování kolize, 287
MPF, 557
MSG, 349
mySupport-Dokumentation, 23

N

Najíždění na referenční bod, 358
Naklápění roviny / naklápění nástroje / nastavování orientace nástroje – CYCLE800
externí programování, 1108
NAMETOINT, 728
Nastavení souřadného systému orovnávače - CYCLE435
externí programování, 1103
Nástroj
- orientace:, 772
Bod pro výměnu, 37
-břit, 80
-číslo typu, 71
-korekce délky, 68
-korekce rádiusu, 69
-Paměť korekcí, 69
-parametry, 742
-skupina, 71
-špička, 70
-typ, 71
-výměna pomocí T-čísla, 65
-výměna s příkazem M6, 66
Návěští skoku
při opakování úseku programu, 470
při programových skocích, 466

Návratová dráha
-směr při řezání závitu, 234
Navrtávání středicích důlků - CYCLE81
externí programování, 1078
Název cesty, 561
Název jednotky, 561
Název souboru, 563
NC program
vytvořit, 52
NCK, 397
Nekonečná smyčka, 480
Nepřímé programování
adres, 434
atributů polohy, 438
G-kódů, 437
řádků výrobního programu, 440
NEWCONF, 1000
NOC, 937
NORM, 263
NOT, 444
NPROT, 583
NPROTDEF, 580
Nulový frame, 145
NUMBER, 449
Numerické rozšíření, 381
NUT, 685

O

Obráběcí bod frézy, 760
Obrobek
- Počátek souřadného systému, 36
-adresáře, 557
-hlavní adresář, 557
-kontura, 173
-počítadlo, 998
Obvodová rychlost, 104
Oddělování třísky
podporované funkce, 1022
Oddělování třísky - CYCLE951
externí programování, 1137
Oddělování třísky / odstraňování zbytkového materiálu oddělováním třísky / zapichování / zapichování zbytkového materiálu / soustružení zápichu / odstraňování zbytkového materiálu soustružením zápichu - CYCLE952
externí programování, 1140
Oddělování třísky od kontury - CYCLE95
externí programování, 1092
Odeslání zpětné vazby, 23
Odjíždění
nezávislé na pohonu, 993
řízené NC systémem, 988
Odlehčovací zápich tvaru E a F / závitový zápich - CYCLE940
externí programování, 1135
OEMIPO1/2, 630
OFFN, 253
Offset
Délka nástroje, 84
Rádius nástroje-, 84
Ohraničení pracovního pole
v BCS, 352
Okrajové podmínky u transformací, 723
OMA1 ... OMA5, 629
On-line korekce délky nástroje, 793
Opakování částí programu
s nepřímým programováním CALL, 535
OpenSSL, 25
OR, 444
ORIAXES, 683
ORIC, 772
ORICONCCW, 685
ORICONCW, 685
ORICONIO, 685
ORICONTO, 685
ORICURVE, 688
ORID, 772
Orientace nástroje
vzhledem ke dráze, 694
Orientace nástroje vztahující se ke dráze, 694
Orientační osy, 683
Orientovatelný držák nástroje, 781
ORIEULER, 683
ORIMKS, 681
ORIPATH, 695
ORIPATHS, 695
ORIPLANE, 685
ORIRESET(A, B, C), 673
ORIROTA, 692
ORIROTC
při interpolaci otáčení nástroje, 697
při otáčení orientace nástroje, 692
ORIROTR, 692
ORIROTT, 692
ORIRPY, 683
ORIRPY2, 683
ORIS, 772
ORISOF, 703
ORISOLH, 796
ORISON, 703
ORIVECT, 683

ORIVIRT1, 683
ORIVIRT2, 683
ORIWKS, 681
OS, 965
OSB, 965
OSC, 772
Oscilační pohyb
Bod obratu, 972
Oblast obratu, 972
Přísuv v bodě obratu, 974
Oscilační pohyby
Asynchronní, 965
Asynchronní oscilační pohyby, 965
Dílčí přísuv, 972
řízení prostřednictvím synchronních akcí, 969
Synchronní oscilační pohyby, 970
OSCILL, 969
OSCTRL, 965
OSD, 772
OSE, 965
OSNSC, 965
OSOF, 772
Osová vazba řídící hodnotou, 925
OSP1, 965
OSP2, 965
OSS, 772
OSSE, 772
OST, 772
OST1, 965
OST2, 965
Osy
*vyměnit, 881
Doplňkové, 372
Dráhové, 374
Geometrické, 371
Hlavní, 371
Kanálové, 373
PLC:-, 375
polohovací, 374
Příkazové, 375
Řídící spřažená osa, 377
Spřažené, 376
Stroj, 373
Synchronizované, 375
-typy, 370
Vlečení, 907
-Zásobník, 376
otáčky nástroje
maximální, 95
Otevřená drážka - CYCLE899
externí programování, 1129
Otočení
Programovatelné, 318
vektoru orientace, 692
OTOL, 849
OTOLG0, 184
OVR, 126
OVRA, 126
OVRRAP, 126
Označení
pro řetězec znaků, 53
pro speciální číselné hodnoty, 53
pro systémové proměnné, 53

P

P..., 531
P_ACTFRAME, 658
PACCLIM, 879
Paměť
Pracovní-, 567
Program, 555
Předběžné zpracování, 859
Paměť korekcí, 741
Paměť programů
Standardní adresáře, 556
Typy souborů, 557
Parametr
Nástroj, 742
-předávání při volání podprogramu, 528
Parametry
Formální-, 497
-předávání při volání podprogramu, 497
Skutečné-, 497
Parametry kruhu
vypočítat, 1034
Parametry měřicího cyklu
CYCLE961, 1190
CYCLE971, 1208
CYCLE973, 1171
CYCLE974, 1173
CYCLE976, 1179
CYCLE977, 1187
CYCLE978, 1181
CYCLE979, 1193
CYCLE982, 1206
CYCLE994, 1176
CYCLE995, 1199
CYCLE996, 1201
CYCLE9960, 1204
CYCLE997, 1196
CYCLE998, 1184
PAROT, 335

PAROTOF, 335
PCALL, 537
PDELAYOF, 976
PDELAYON, 976
Pevný bod
najíždění, 359
Pevný doraz, 363
PFRAME, 636
PHI
Polynom orientace, 691
při orientaci podél kuželové plášťové plochy, 685
PHU, 411
PL
při polynomické interpolaci, 611
při splinové interpolaci, 598
Platnost, 20
PLC
-Osy, 375
PM, 273
PO, 611
PO[PHI]
Polynom orientace, 691
při orientaci podél kuželové plášťové plochy, 685
při otáčení orientace nástroje, 695
PO[PSI]
Polynom orientace, 691
při orientaci podél kuželové plášťové plochy, 685
při otáčení orientace nástroje, 695
PO[THT]
Polynom orientace, 691
při otáčení orientace nástroje, 695
PO[XH]
Polynom orientace, 691
při udání orientace pomocí dvou styčných bodů, 688
PO[YH]
Polynom orientace, 691
při udání orientace pomocí dvou styčných bodů, 688
PO[ZH]
Polynom orientace, 691
při udání orientace pomocí dvou styčných bodů, 688
POCKET3 - Pravoúhlá kapsa
externí programování, 1042
POCKET4 - Kruhová kapsa
externí programování, 1045
Počáteční bod, 37
Počáteční bod – koncový bod, 172
Počátek souřadného systému
Nastavitelné, 41
Základní, 41
Počátky souřadného systému
při soustružení, 170
Početní parametry
Globální, 394
Kanálové, 392
Podélná drážka - SLOT1
externí programování, 1048
Podlouhlá díra - LONGHOLE
externí programování, 1053
Podpora pro produkt, 24
Podprogram
-návrat zpět, s nastavitelnými parametry (RET ...), 515
-návrat zpět, s nastavitelnými parametry (RETB...), 522
-název, 494
-opakování, 531
Použití, 492
Programovatelná cesta pro vyhledávání, 538
-volání bez předávání parametrů, 526
-volání modální, 532
-volání nepřímé, 534
-volání s předáváním parametrů, 528
Podprogramy podle normy DIN, 564
Pól, 175
Polární rádius, 31
Polární souřadnice, 31
Polární transformace, 665
Polární úhel, 32
Pole
-definice, 419
-prvek, 419
POLF
pro odjíždění řízené NC systémem, 988
pro rychlý zpětný pohyb v průběhu řezání závitu, 233
POLFA, 988
POLFMASK
pro odjíždění řízené NC systémem, 988
pro rychlý zpětný pohyb v průběhu řezání závitu, 233
POLFMLIN
pro odjíždění řízené NC systémem, 988
pro rychlý zpětný pohyb v průběhu řezání závitu, 233
Polohovací osy, 374
Polohovací vzor Kružnice nebo Kruhový oblouk - HOLES2
externí programování, 1040
Polohovací vzor Mřížka nebo Obdélník - CYCLE801
externí programování, 1111

Polohovací vzor Přímka - HOLES1
externí programování, 1039
POLY, 611
Polynom jmenovatele, 615
Polynomická interpolace, 611
POLYPATH, 611
Pomocné (doplňkové) osy, 372
Pomocný bod frézy, 760
PON, 984
PONS, 976
POS, 114
POSA, 114
POSFS, 937
POSP, 114
Posunutí počátečního bodu
při řezání závitu, 224
Posunutí počátku
Externí, 646
Nastavitelné, 144
Programovatelná, 312
Posuv
- rychlost, 188
Časově reciproční, 109
-korekce, 126
Měřicí jednotky, 111
-Override, 131
Pravidla, 106
pro dráhové osy, 108
pro polohovací osy, 123
pro synchronní osy, 110
s korekcí pomocí ručního kolečka, 129
Posuv od bodu k bodu, 712
Posuv PTP v kartézských souřadnicích, 712
POT, 442
PR, 273
Pracovní paměť, 567
Pracovní rovina, 35
Pravidlo tří prstů, 38
Pravoúhlá kapsa - POCKET3
externí programování, 1042
Pravoúhlý čep - CYCLE76
externí programování, 1070
Pravý závit, 225
PREPRO, 513
PRESETON, 648
PRESETONS, 649
PRIO, 546
PRLOC, 397
procedura
předem definovaná, 1284
Proces DataShare, 1010
Profilování - CYCLE495
externí programování, 1103
Program
- simulace, 999
-adresování, 560
-doba zpracování, 994
-hlavička, 54
Inicializační-, 568
-konec, 47
-název, 44
-opakování, 531
-paměť, 557
-skoky, 465
-větvení, 468
Programová smyčka
Konečná smyčka, 480
Smyčka IF, 478
Smyčka REPEAT, 483
Smyčka s počítadlem, 481
Smyčka WHILE, 482
Programování konturové křivky, 213
Programování kruhu
Druhy interpolace, 191
pomocí polárních souřadnic, 198
pomocí rádiusu a koncového bodu, 194
pomocí středu a koncového bodu, 191
pomocí úhlu kruhové výseče a středu, 196
Programování NC systémů
Sada znaků, 53
Programování orientace, 683
Programování průměrů, 164
Programování rádiusů, 164
Programovatelné zastavení, 348
Proměnná
Konvertování typu, 433
Proměnná typu FRAME
Předem definované proměnné typu FRAME, 636
Přiřazení hodnot, 639
Vyvolání transformace souřadného systému, 634
Proměnné
Konverze typu, 449
uživatelem definovaná, 397
Prostřihování, 976
PROTA, 729
PROTD, 731
PROTS, 730
Předběžné zpracování
-paměť, 859
Předvrtání konturové kapsy - CYCLE64
externí programování, 1062
Přepínatelné geometrické osy, 889
Přeskakované bloky, 50

Přeskakované úrovně, 50
Přesné najetí, 296
Přesnost kontury
    Programovatelná, 854
Převodový faktor, 905
Příčná osa, 171
Příkaz, 45
    -osy, 375
Příkaz posuvu, 172
Příkazy G-funkce, 1263
přímky
    -interpolace, 188
Příprava kontury
    Chybové hlášení, 1036
Přiřazování hodnot, 48
Přizpůsobení zatížení - CYCLE782
    externí programování, 1105
PSI
    Polynom orientace, 691
    při orientaci podél kuželové plášťové plochy, 685
PTP, 713
PTPG0, 713
PTPWOC, 713
PUD, 398
PUNCHACC, 976
PUTFTOC, 756
PUTFTOCF, 755
PW, 598

Q

QR kód, 25
QU, 345

R

RAC, 166
Rádius
    efektivní, 112
READ, 574
REAL, 397
REDEF, 403
Referenční bod, 37
Relační operátory, 444
RELEASE, 881
REP, 419
REPEAT, 470
REPEATB, 470
REPOSA, 866
REPOSH, 866
REPOSHA, 866
REPOSL, 866
REPOSQ, 866
REPOSQA, 866
RET, 514
RET (s nastavitelnými parametry), 515
RETB (s nastavitelnými parametry), 522
Režim jazyka, 1020
Režim při změnách zrychlení, 843
Režim řízení pohybu po dráze, 298
RG, 394
RIC, 166
RINDEX, 453
RMBBL, 866
RMEBL, 866
RMIBL, 866
RMNBL, 866
RND, 247
RNDM, 247
Rohy kontury
    srážení hran, 247
    zaoblení, 247
ROT, 318
ROTS, 324
ROUND, 442
ROUNDUP, 447
Rovinné frézování - CYCLE61
    externí programování, 1057
Rovinné závity, 228
Rozšířený způsob zápisu adresy, 381
RP, 176
RPL, 318
Ruční kolečko
    -korekce, 129
Run MyScreens, 1005
Rutina přerušení
    Aktivovat / deaktivovat, 548
    Nové přiřazení, 547
    Programovatelný směr pohybu, 551
    Rychlé pozvednutí od kontury, 549
    Vymazat, 548
    Zpětný pohyb, 551
Rychlé pozvednutí od kontury, 549
Rychlost pohybu po dráze
    Maximum, 877
Rychlý zpětný pohyb
    Řezání závitu, 233
Řetězec nástroje, 739
Řetězec obrobku, 739
Řetězec závitů - CYCLE98
    externí programování, 1094
Řetězové kótování, 34
Řezná rychlost, 93

Řezná rychlost (konstantní), 99
Řídící
    struktury, 476
Řídící osa
    v případě axiální vazby řídící hodnotou, 925
Řídící vřeteno, 373

S

S, 89
Sada znaků, 53
SAVE, 504
SBLOF, 505
SBLON, 505
SCALE, 327
SCC, 99
SCPARA, 896
SD, 598
SD41610, 741
SD41611, 741
SD42010, 232
SD42440, 154
SD42442, 154
SD42450$SC_CONTPREC, 855
SD42451 $SC_CONTPREC_G00_ABS, 855
SD42460 $SC_MINFEED (minimální posuv po dráze, když je aktivní příkaz CPRECON), 855
SD42465, 304
SD42466, 304
SD42475, 701
SD42476, 701
SD42477, 701
SD42900, 748
SD42910, 748
SD42920, 749
SD42930, 749
SD42935, 751
SD42940, 752
SD42984, 808
SD43240, 120
SD43250, 120
SET, 419
SETAL, 1014
SETDNO, 779
SETINT, 546
SETM, 484
SETMS, 89
SETTCOR, 824
SF, 223
Siemens Industry Online Support
    Aplikace, 25
Simulace, 999
Simulace řídící hodnoty, 929
SIN, 442
Singulární polohy, 682
SINUMERIK, 19
skok
    na návěští skoku, 465
    na začátek programu, 464
SLOT1 - Podélná drážka
    externí programování, 1048
SLOT2 - Kruhová drážka
    externí programování, 1050
Složka framu
    FI, 641
    MI, 641
    RT, 641
    SC, 641
    TR, 641
Směrový vektor, 678
Smyčka s počítadlem, 481
Snížení rychlosti na všech rozích, 631
SOFT, 843
SOFTA, 843
SON, 976
SONS, 976
Soubor
    -informace, 577
souřadnice
    Kartézské, 30
    Polární, 31
    Válcové, 177
Souřadný systém
    Přehled, 38
    Základní, 40
Souřadný systém obrobku, 42
Souřadný systém stroje, 38
Soustružení závitu - CYCLE99
    externí programování, 1098
Soustružnické nástroje, 76
SPCOF, 117
SPCON, 117
Speciální nástroje, 78
Speciální znaky, 53
SPF, 557
SPI, 887
SPIF1, 976
SPIF2, 976
Spirální interpolace, 206
Spline
    -interpolace, 598
    -Typy, 605
SPLINEPATH, 609
Splinový svazek, 609

SPN, 981
SPOF, 976
SPOS, 118
SPOSA, 118
SPP, 981
SPRINT, 456
Spřažené
-Osy, 376
Řídící spřažená osa, 377
SQRT, 442
SR, 135
SRA, 135
ST, 135
STA, 135
Standardní rozsah, 21
START, 484
STARTFIFO, 859
STAT, 713
Stav vazby
při vlečení, 908
v případě axiální vazby řídící hodnotou, 930
STOLF, 184
STOPFIFO, 859
STOPRE, 859
STRING, 397
-délka, 452
formátovat, 456
-operace, 448
-Zřetězení, 451
STRINGIS, 1001
STRLEN, 452
Stroj
- Počátek souřadného systému, 36
Stroje
-osy, 373
Struktura kapitol, 20
SUBSTR, 454
Sudovitý závit, 237
SUPA
při deaktivování framu, 338
při posunutí počátku (nuly), 144
SVC, 93
Synchronizace
hrubý, 940
jemné, 940
Synchronizace otáček, 937
Synchronizace polohy, 937
Synchronizace polohy s úhlovým offsetem, 937
synchronizované
-osy, 375
Synchronní oscilační pohyby
Definice přiřazení, 973
Následující dílčí přísuv, 976
Přiřazení oscilační a přísuvné osy, 973
Přísuv v oblasti obratu, 974
Přísuvný pohyb, 974
Synchronní akce, 973
Vyhodnocování, takt IPO, 975
Synchronní vřeteno
-definice páru, 943
Vazba, 936
SYNR, 397
SYNRW, 397
SYNW, 397
Systém měřicích jednotek, 160
Systémové framy, 656
Šikmé zapichovací broušení, 709
Špička frézy, 760

T

T0, 65
TAN, 442
TANG, 955
TANGDEL, 959
TANGOF, 959
TANGON, 957
TCARR, 786
TCOABS, 786
TCOFR, 786
TCOFRX, 786
TCOFRY, 786
TCOFRZ, 786
Technická podpora, 24
THETA
při interpolaci otáčení nástroje, 697
při otáčení orientace nástroje, 692
TILT, 674
TLIFT, 956
TMOF, 986
TMON, 986
TOFF, 84
TOFFL, 84
TOFFLR, 84
TOFFOF, 794
TOFFON, 794
TOFFR, 84
TOFRAME, 335
TOFRAMEX, 335
TOFRAMEY, 335
TOFRAMEZ, 335
Tolerance pohybů s příkazem G0, 184
TOLOWER, 452
TOOLENV, 813

TOROT, 335
TOROTOF, 335
TOROTX, 335
TOROTY, 335
TOROTZ, 335
TOUPPER, 452
TOWBCS, 750
TOWKCS, 750
TOWMCS, 750
TOWSTD, 750
TOWTCS, 750
TOWWCS, 750
TRAANG
    s naprogramovatelným úhlem, 707
TRACON, 710
TRACYL, 704
TRAFOOF, 724
TRAFOON, 733
TRAILOF, 905
TRAILON, 905
TRANS, 312
Transformace
    Kinematické transformace, 662
    Transformace orientace, 661
    Tří-, čtyř- a pětiosá transformace, 672
    zřetězené, 710
    Zřetězené transformace, 662
Transformace orientace TRAORI
    Generická 5/6-osá transformace, 665
    Kinematika stroje, 664
    Pohyby pro nastavení orientace, 663
    Programování orientace, 673
    Varianty programování orientace, 673
Transformace s naklápěcí lineární osou, 671
Transformace souřadného systému (Frame), 42
Transformace šikmého úhlu (TRAANG)
    s naprogramovatelným úhlem, 707
Transformace válcového pláště, 665
TRANSMIT, 704
TRAORI, 672
Trhavé pohyby
    -korekce, 876
    -omezení, 843
TRUE, 397
TRUNC, 442
TU, 717
TURN, 206
Typ kinematiky, 785

U

Údaje rozměrů
    Možnosti, 150
    pro kruhové osy a vřetena, 158
    v průměrech, 164
    v rádiusech, 164
Úhel RPY, 677
ULI, 410
UNTIL, 483
Upichování - CYCLE92
    externí programování, 1091
Úsek programu
    -opakování, 470
Uživatelské XML, 1005

V

Válcové souřadnice, 177
Válcové závity, 228
Vazba
    Generická, 947
Vazba pomocí požadované hodnoty, 940
Vazba pomocí rychlosti, 940
Vazba pomocí skutečné hodnoty, 940
Vazba řídící hodnotou
    Synchronizace řídící a vlečné osy., 928
    Vazba pomocí skutečné a požadované hodnoty, 928
Vektor orientace THETA, 692
velká a malá písmena nejsou rozlišována, 559
VELOLIM, 874
VELOLIMA, 846
Vlečené osy, 907
Vlečení, 905
Vlečná osa
    v případě axiální vazby řídící hodnotou, 925
Vlečné spojení os, 905
Volání kontury - CYCLE62
    externí programování, 1059
Volání podprogramu s udáním cesty a s parametry, 537
Volitelné zastavení, 348
Vrták, 73
Vrtání - CYCLE82
    externí programování, 1079
Vrtání hlubokých děr 1 - CYCLE83
    externí programování, 1082
Vrtání hlubokých děr 2 - CYCLE830
    externí programování, 1117

Vrtání závitů bez vyrovnávací hlavičky - CYCLE84
externí programování, 1085
Vrtání závitů s vyrovnávací hlavičkou - CYCLE840
externí programování, 1127
Vřeteno
*vyměnit, 881
Hlavní, 373
M-funkce, 349
Najeďte na požadovanou polohu, 118
-omezení otáček, 105
-otáčky, 89
-směr otáčení, 89
Vyhlazení
průběh orientace, 703
Vypisování bloku
potlačení, 510
Výpočet dráhy, 379
Výsledek
do externího zařízení/souboru, 1010
Vysokorychlostní obrábění - CYCLE832
externí programování, 1123
Vystružování - CYCLE85
externí programování, 1088
Výstup pomocné funkce
Rychlé, 345
v režimu řízení pohybu po dráze, 346
Vlastnosti pomocných funkcí, 344
Vyšší jazyk NC systému, 46
Vyvrtávání - CYCLE86
externí programování, 1089
Vzdělávání, 24
Vztažné body, 36
Vztažný rádius, 112

W

WAB, 273
WAITC, 937
WAITE, 484
WAITENC, 895
WAITM, 484
WAITMC, 484
WAITP, 114
WAITS, 118
WALCS<n>, 355
WALCS0, 355
WALIMOF, 352
WALIMON, 352
WCS, 42
Srovnání podle obrobku, 335
WHEN, 964
WHEN-DO, 973
WHENEVER, 964
WHENEVER-DO, 973
WHILE, 482
WORKPIECE, 1016
WRITE, 570
WRTPR, 351

X

X..., 173
XOR, 444

Y

Y..., 173

Z

Z..., 173
Začátek programového bloku - GROUP_BEGIN, 1167
Zadávání inkrementálních rozměrů, 153
Základní nastavení orientace nástroje - ORIRESET, 673
Základní posunutí, 41
Základní souřadný systém, 40
Základní souřadný systém počátku (nuly), 41
Zaoblení, 247
Zaokrouhlování, 447
Zápich - CYCLE930
externí programování, 1132
zastavení
nezávislé na pohonu, 992
řízené NC systémem, 992
Zastavení
na konci cyklu, 348
Programovatelné, 348
Volitelné, 348
Zastavení interního předběžného zpracování, 370
Zastavení předběžného zpracování
Interní, 370
Závit
-řetězec, 224
-řezání G33, 223
-řezání G34 G35, 230
-směr otáčení, 225
-stoupání, 230
-vícechodý, 224
Zbývající čas
pro daný obrobek, 996
Zjišťování a prohledávání oblastí určených pro vyhledávání, 865

Znak 0, 448
Zpoždění na vnitřních rozích, 631
Zpracování blok po bloku
    -potlačení, 505
Způsob zápisu typu EES, 560
Způsob zápisu typu NCK, 560
Zrychlení pohybu po dráze
    Maximum, 879
Zřetězení obráběcích operací
    proměnných typu String, 451