SIEMENS

SINUMERIK

SINUMERIK 840D sl/828D
上級編

プログラミングマニュアル

適用

制御
SINUMERIK 840D sl / 840DE sl / 828D

CNC ソフトウェア　バージョン 4.8 SP3

08/2018
6FC5398-2BP40-6TA2

まえがき
基本的な安全に関する指示事項 1
フレキシブルな NC プログラミング 2
ファイルとプログラムの管理 3
保護領域 4
特殊動作命令 5
座標変換(フレーム) 6
座標変換 7
キネマテイック結合 8
キネマティック結合付きの衝突防止 9
キネマティック結合による座標変換 10
工具オフセット 11
軌跡の移動動作 12
軸連結 13
シンクロナイズドアクション 14
揺動 15
パンチングとニブリング 16
研削加工 17
その他の機能 18
ユーザー荒削りプログラム 19
サイクルを外部的にプログラミング 20
テーブル 21
付録 A

Промэлектроника, sales@prom-elec.com

法律上の注意

警告事項

本書には、ユーザーの安全性を確保し製品の損傷を防止するうえ守るべき注意事項が記載されています。ユーザーの安全性に関する注意事項は、安全警告サインで強調表示されています。このサインは、物的損傷に関する注意事項には表示されません。以下に表示された注意事項は、危険度によって等級分けされています。

⚠ 危険

回避しなければ、直接的な死または重傷に至る危険状態を示します。

⚠ 警告

回避しなければ、死または重傷に至るおそれのある危険な状況を示します。

⚠ 注意

回避しなければ、軽度または中度の人身傷害を引き起こすおそれのある危険な状況を示します。

通知

回避しなければ、物的損傷を引き起こすおそれのある危険な状況を示します。

複数の危険レベルに相当する場合は、通常、最も危険度の高い事項が表示されることになっています。安全警告サイン付きの人身傷害に関する注意事項があれば、物的損傷に関する警告が付加されます。

有資格者

本書が対象とする製品 / システムは必ず有資格者が取り扱うものとし、各操作内容に関連するドキュメント、特に安全上の注意及び警告が遵守されなければなりません。有資格者とは、訓練内容及び経験に基づきながら当該製品 / システムの取り扱いに伴う危険性を認識し、発生し得る危害を事前に回避できる者をいいます。

シーメンス製品を正しくお使いいただくために

以下の事項に注意してください。

⚠ 警告

シーメンス製品は、カタログおよび付属の技術説明書の指示に従ってお使いください。他社の製品または部品との併用は、弊社の推奨もしくは許可がある場合に限ります。製品を正しく安全にご使用いただくには、適切な運搬、保管、組み立て、据え付け、配線、始動、操作、保守を行ってください。ご使用になる場所は、許容された範囲を必ず守ってください。付属の技術説明書に記述されている指示を遵守してください。

商標

®マークのついた称号はすべて Siemens AG の商標です。本書に記載するその他の称号は商標であり、第三者が自己の目的において使用した場合、所有者の権利を侵害することになります。

免責事項

本書のハードウェアおよびソフトウェアに関する記述と、実際の製品内容との一致については検証済みです。 しかしなお、本書の記述が実際の製品内容と異なる可能性もあり、完全な一致が保証されているわけではありません。記載内容については定期的に検証し、訂正が必要な場合は次の版て更新いたします。

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
ドイツ

注文番号: 6FC5398-2BP40-6TA2
℗ 08/2018 変更する権利を留保

Copyright © Siemens AG 1995 - 2018.
無断複写・転載を禁じます

まえがき

SINUMERIK 取扱説明書

SINUMERIK 取扱説明書は以下のカテゴリに分類されます。

- 製品の取扱説明書/カタログ
- ユーザーマニュアル
- メーカ/サービスマニュアル

他の情報

次の項目に関する情報は、以下のアドレス (https://support.industry.siemens.com/cs/de/en/view/108464614)にあります:

- 取扱説明書の注文/取扱説明書の概要
- 説明書をダウンロードするその他のリンク
- オンラインでの説明書の利用(マニュアル/情報の検索)

ご提案や訂正など、本書に関するお問い合わせがございましたら、以下の電子メールアドレス (mailto:docu.motioncontrol@siemens.com)にご連絡ください。

mySupport/ドキュメンテーション

以下のアドレス (https://support.industry.siemens.com/My/ww/en/documentation)では、シーメンスのコンテンツに基づいてお客さま自身の文書を作成し、お客さまの機械装置の取扱説明書にご利用いただく方法を説明しています。

トレーニング

以下の "address (http://www.siemens.com/sitrain)" では、SITRAIN (製品、システム、およびオートメーションエンジニアリングソリューション用のシーメンスのトレーニング)に関する情報を提供しています。

FAQ

[Service&Support]ページの[Product Support (https://support.industry.siemens.com/cs/de/en/ps/faq)]の[Frequently Asked Questions]を参照してください。

SINUMERIK

SINUMERIK に関する情報は以下のアドレス (http://www.siemens.com/sinumerik)にあります。

対象

この文書は以下の方を対象にしています。

- プログラマ
- プロジェクトエンジニア

本書の目的

上記の対象読者は、プログラミング説明書を使用して、プログラムとソフトウェア ユーザーインタフェースの開発、プログラミング、テスト、デバッグをおこなうことができます。

記述の範囲

このプログラミングマニュアルは標準範囲の機能について記載しています。工作機械メーカが実施した拡張または変更箇所については，工作機械メーカ発行の説明書に記載されています。

その他本書で説明していない機能も、制御装置で実行できる場合があります。ただし、これは、そのような機能を新しい制御装置によって提供したり、サービス時に提供したりするということではありません。

さらに、単純化のために、本書にはすべてのタイプの製品に関するすべての詳細情報は記載されていません。また取り付け、運転および保守について考えられるすべてのケースを網羅しているわけではありません。

テクニカルサポート

テクニカルサポートの国別電話番号については、インターネットの [Contact (連絡先)] の下のアドレス (https://support.industry.siemens.com/sc/ww/en/sc/2090)を参照してください。

構造と内容に関する情報

プログラミングマニュアル、基本編/上級編

NC プログラミング説明書は、下記の 2 冊に分かれています。

1. **基本編**
   「基本偏」プログラミング説明書は、穴あけ、フライス、旋盤加工の経験をお持ちの熟練オペレータを対象としています。本書は、簡単なプログラミング例を使用して、命令と命令文について解説します。これらの定義は DIN 66025 にも準拠します。
2. **上級編**
   「上級編」プログラミング説明書は、詳細で包括的なプログラミング知識をお持ちの技術者を対象としています。SINUMERIK 制御装置は専用のプログラミング言語を使用するため、自由曲面やチャンネル協調などの複雑なワーク加工のプログラム指令が可能です。また、複雑な運転でも技術者が簡単にプログラム指令できるようにします。

本書で記述された NC 言語要素の適用範囲

本書で記述されたすべての NC 言語要素は SINUMERIK 840D sl で有効です。
SINUMERIK 828D に関する適用については、「命令:SINUMERIK 828D での適用 (ページ 1061)」の表を参照してください。

目次

まえがき 3

1 基本的な安全に関する指示事項 19

1.1 一般的な安全に関する指示事項 19

1.2 アプリケーション例に対する保証と責任 20

1.3 産業セキュリティ 21

2 フレキシブルな NC プログラミング 23

2.1 変数 23
2.1.1 システムデータ 23
2.1.2 予約ユーザー変数算術変数 26
2.1.2.1 チャネル別算術変数(R) 26
2.1.2.2 グローバル算術変数(RG) 28
2.1.3 予約ユーザー変数リンク変数 30
2.1.4 ユーザー変数の定義(DEF) 33
2.1.5 システムデータ、ユーザーデータ、NC 命令(REDEF)の再定義 40
2.1.6 属性:初期値 44
2.1.7 属性: 制限値(LLI、ULI) 47
2.1.8 属性: 物理単位(PHU) 49
2.1.9 属性: アクセス権(APR、APW、APRP、APWP、APRB、APWB) 51
2.1.10 属性:データクラス(DCM、DCI、DCU) - SINUMERIK 828D のみ 57
2.1.11 定義と再定義が可能な属性の概要 58
2.1.12 配列変数の定義と初期化(DEF、SET、REP) 60
2.1.13 配列変数の定義と初期化(DEF、SET、REP):詳細情報 65
2.1.14 データタイプ 66
2.1.15 変数の有効性確認(ISVAR) 67
2.1.16 属性値/データタイプ(GETVARPHU、GETVARAP、GETVARLIM、GETVARDIM、GETVARDFT、GETVARTYP)の読み取り 69

2.2 間接プログラミング 77
2.2.1 アドレスの間接プログラミング 77
2.2.2 G 命令の間接プログラム 79
2.2.3 位置属性の間接プログラミング(GP) 81
2.2.4 パートプログラム行の間接プログラミング(EXECSTRING) 84

2.3 算術機能 85

2.4 比較演算と論理演算 88

2.5 比較演算誤差の精度補正(TRUNC) 91

2.6 変数の最小値、最大値、および範囲(MINVAL、MAXVAL、および BOUND) 93

2.7 演算子の優先度....95
2.8 実行可能なタイプ変換....96
2.9 文字列演算子....97
2.9.1 STRING へのタイプ変換(AXSTRING)....97
2.9.2 STRING からのタイプ変換(NUMBER、ISNUMBER、AXNAME)....98
2.9.3 文字列の結合(<<)....100
2.9.4 小文字/大文字への変換(TOLOWER、TOUPPER)....101
2.9.5 文字列の長さの特定(STRLEN)....102
2.9.6 文字列中の文字/文字列の検索(INDEX、RINDEX、MINDEX、MATCH)....103
2.9.7 抽出文字列の選択(SUBSTR)....104
2.9.8 個々の文字の読み取りと書き込み....105
2.9.9 文字列のフォーマット(SPRINT)....106
2.10 プログラムのジャンプと分岐....117
2.10.1 プログラムの先頭への復帰ジャンプ(GOTOS)....117
2.10.2 ジャンプマークへのプログラムのジャンプ(GOTOB、GOTOF、GOTO、GOTOC)....118
2.10.3 プログラム分岐(CASE ... OF ... DEFAULT ...)....121
2.11 プログラム区間の繰り返し(REPEAT、REPEATB、ENDLABEL、P)....124
2.12 チェック命令....131
2.12.1 条件付き命令と分岐(IF、ELSE、ENDIF)....133
2.12.2 連続プログラムループ(LOOP、ENDLOOP)....134
2.12.3 カウントループ(FOR ... TO ...、ENDFOR)....135
2.12.4 ループの先頭に条件があるプログラムループ(WHILE、ENDWHILE)....137
2.12.5 ループの終了に条件があるプログラムループ(REPEAT、UNTIL)....138
2.12.6 入れ子のチェック命令を含むプログラム例....138
2.13 協調指令(INIT、START、WAITM、WAITMC、WAITE、SETM、CLEARM)....140
2.14 割り込みルーチン(ASUB)....147
2.14.1 割り込みルーチンの機能....147
2.14.2 割り込みルーチンの作成....148
2.14.3 割り込みルーチンの割り当てと起動(SETINT、PRIO、BLSYNC)....149
2.14.4 割り込みルーチンの割り当ての解除/再起動(DISABLE、ENABLE)....151
2.14.5 割り込みルーチンの割り当ての解除(CLRINT)....152
2.14.6 輪郭からの高速リトラクト(SETINT LIFTFAST、ALF)....153
2.14.7 輪郭からの高速リトラクトの移動方向 155
2.14.8 割り込みルーチンの動作順序....158
2.15 軸入れ替え、主軸入れ替え(RELEASE、GET、GETD)....160
2.16 別のチャネルへの軸の移行(AXTOCHAN)....165
2.17 マシンデータの起動(NEWCONF)....167
2.18 ファイルの書き込み(WRITE)....168
2.19 ファイルの削除(DELETE)....173

2.20 ファイルの行の読み取り(READ)....175
2.21 ファイルの存在の確認(ISFILE)....178
2.22 ファイル情報の読み出し(FILEDATE、FILETIME、FILESIZE、FILESTAT、FILEINFO)....180
2.23 切り上げ(ROUNDUP)....183
2.24 サブプログラム機能....184
2.24.1 概要....184
2.24.1.1 サブプログラム....184
2.24.1.2 サブプログラム名称....185
2.24.1.3 サブプログラムの入れ子....186
2.24.1.4 検索パス....188
2.24.1.5 仮パラメータと実パラメータ....188
2.24.1.6 パラメータ転送....189
2.24.2 サブプログラムの定義....190
2.24.2.1 パラメータ転送を含まないサブプログラム....190
2.24.2.2 値渡しパラメータ転送を含むサブプログラム(PROC)....191
2.24.2.3 参照渡しパラメータ転送を含むサブプログラム(PROC、VAR)....194
2.24.2.4 モーダル G 機能の保存(SAVE)....196
2.24.2.5 シングルブロック実行のマスク(SBLOF、SBLON)....197
2.24.2.6 実行中のブロック表示のマスク(DISPLOF、DISPLON、ACTBLOCNO)....203
2.24.2.7 解析によるサブプログラムの識別(PREPRO)....207
2.24.2.8 サブプログラム戻り M17....208
2.24.2.9 RET サブプログラム戻り....209
2.24.2.10 設定可能なサブプログラム復帰ジャンプ(RET ...)....210
2.24.2.11 パラメータ設定可能なサブプログラム復帰ジャンプ(RETB ...)....217
2.24.3 サブプログラムの呼び出し....221
2.24.3.1 パラメータ転送を含まないサブプログラム呼び出し....221
2.24.3.2 パラメータ転送を含むサブプログラム呼び出し(EXTERN)....224
2.24.3.3 プログラム繰り返し回数(P)....227
2.24.3.4 モーダルサブプログラム呼び出し(MCALL)....228
2.24.3.5 間接サブプログラム呼び出し(CALL)....230
2.24.3.6 呼び出しプログラム要素を指定した間接サブプログラム呼び出し(CALL BLOCK ... TO ...)....231
2.24.3.7 ISO 言語で作成したプログラムの間接呼び出し(ISOCALL)....233
2.24.3.8 パス指定とパラメータによるサブプログラム呼び出し(PCALL)....234
2.24.3.9 サブプログラム呼び出しの拡張検索パス(CALLPATH)....235
2.24.3.10 外部サブプログラムの実行(840D sl) (EXTCALL)....236
2.24.3.11 外部サブプログラムの実行(828D) (EXTCALL)....241
2.25 マクロ機能(DEFINE ... AS)....246
3 ファイルとプログラムの管理....251
3.1 プログラムメモリ....251

3.1.1 NC プログラムメモリ....251
3.1.2 外部プログラムメモリ....254
3.1.3 プログラムメモリファイルのアドレス指定....255
3.1.4 サブプログラム呼び出しの検索パス....261
3.1.5 パスとファイル名の問い合わせ....263
3.2 作業メモリ(CHANDATA、COMPLETE、INITIAL)....265
4 保護領域....269
4.1 プロテクションゾーンの定義(CPROTDEF、NPROTDEF)....269
4.2 プロテクションゾーンの起動/解除(CPROT、NPROT)....274
4.3 プロテクションゾーン違反、作業領域リミット、およびソフトウェアリミットスイッチのチェック(CALCPOSI)....279
5 特殊動作命令....293
5.1 符号化位置へのアプローチ(CAC、CIC、CDC、CACP、CACN)....293
5.2 スプライン補間(ASPLINE、BSPLINE、CSPLINE、BAUTO、BNAT、BTAN、EAUTO、ENAT、ETAN、PW、SD、PL)....295
5.3 スプライングループ(SPLINEPATH)....307
5.4 NC ブロック圧縮(COMPON、COMPCURV、COMPCAD、COMPSURF、COMPOF)の有効化/無効化....309
5.5 多項式補間(POLY、POLYPATH、PO、PL)....311
5.6 設定可能な軌跡基準(SPATH、UPATH)....318
5.7 タッチトリガプローブによる計測(MEAS、MEAW)....321
5.8 軸別計測(MEASA、MEAWA、MEAC) (オプション)....324
5.9 OEM ユーザー用の応用機能(OMA1 ... OMA5、OEMIPO1、OEMIPO2、G810 ... G829)....336
5.10 コーナ減速による送り速度低下(FENDNORM、G62、G621) 338
5.11 プログラム指令可能な動作終了条件(FINEA、COARSEA、IPOENDA、IPOBRKA、ADISPOSA)....340
6 座標変換(フレーム)....343
6.1 フレーム変数による座標変換....343
6.1.1 予約フレーム変数($P_CHBFRAME、$P_IFRAME、$P_PFRAME、$P_ACTFRAME)....345
6.2 フレームへの値の割り付け....350
6.2.1 値の直接割り当て(軸値、角度、スケール)....350
6.2.2 フレーム成分の読み取りと変更(TR、FI、RT、SC、MI)....352
6.2.3 フレームの接続....353
6.2.4 フレーム変数の定義(DEF FRAME)....355

6.3 荒削りオフセットと仕上げオフセット(CTRANS、CFINE)...357
6.4 外部ゼロオフセット($AA_ETRANS)...359
6.5 原点確立状態をリセットして現在値を設定(PRESETON)...361
6.6 原点確立状態をリセットせずに現在値を設定(PRESETONS)...363
6.7 空間の 3 つの計測点からのフレーム計算(MEAFRAME)...365
6.8 NCU グローバルフレーム...369
6.8.1 チャネル別フレーム($P_CHBFR、$P_UBFR)...370
6.8.2 チャネルの有効フレーム...371
7 座標変換...377
7.1 座標変換タイプの一般的なプログラミング...377
7.1.1 座標変換の旋回移動...379
7.1.2 方向座標変換 TRAORI の一覧...384
7.2 3 軸、4 軸、5 軸座標変換(TRAORI)...387
7.2.1 ユニバーサル工具ヘッドの一般的な関係...387
7.2.2 3 軸、4 軸、5 軸座標変換(TRAORI)...390
7.2.3 向きのプログラミングのタイプと初期設定(ORIRESET)...391
7.2.4 工具オリエンテーションのプログラミング(A...、B...、C...、LEAD、TILT):...393
7.2.5 正面削り(A4、B4、C4、A5、B5、C5)...400
7.2.6 旋回軸の基準(ORIWKS、ORIMKS):...401
7.2.7 旋回軸のプログラミング(ORIAXES、ORIVECT、ORIEULER、ORIRPY、ORIRPY2、ORIVIRT1、ORIVIRT2)...403
7.2.8 円錐面に沿った向きのプログラミング(ORIPLANE、ORICONCW、ORICONCCW、ORICONTO、ORICONIO)...406
7.2.9 2 つの接点の向きの指定(ORICURVE、PO[XH]=、PO[YH]=、PO[ZH]=)...410
7.3 旋回多項式(PO[角度]、PO[座標])...412
7.4 工具オリエンテーションの回転(ORIROTA、ORIROTR、ORIROTT、ORIROTC、THETA)...414
7.5 軌跡に対する向き...417
7.5.1 軌跡に対する向きのタイプ...417
7.5.2 軌跡に対する工具オリエンテーションの回転(ORIPATH、ORIPATHS、回転角度)...419
7.5.3 軌跡に対する工具回転の補間(ORIROTC、THETA)...420
7.5.4 旋回処理のスムージング(ORIPATHS A8=, B8=, C8=)...423
7.6 旋回の圧縮(COMPON、COMPCURV、COMPCAD、COMPOF、COMPSURF)...425
7.7 向きの特性の有効化と無効化(ORISON、ORISOF)...429
7.8 キネマティックトランスフォーメーション...431
7.8.1 面端座標変換(TRANSMIT)の起動...431
7.8.2 円筒補間(TRACYL)の起動...431
7.8.3 角度がプログラム指令可能な傾斜角座標変換(TRAANG)の有効化(TRAANG)...435

7.8.4 研削盤での傾斜プランジ切削(G5, G7)....436
7.9 座標変換重畳(TRACON)の起動....438
7.10 直交 PTP 移動....440
7.10.1 直交 PTP 移動(PTP、PTPG0、PTPWOC、CP)の有効化/無効化....440
7.10.2 ジョイント(STAT)の位置を指定します....441
7.10.3 軸角度の符号を指定します(TU)....446
7.10.4 例 1:ROBX 座標変換による 6 軸ロボットの PTP 移動....449
7.10.5 例 2:総合 5 軸座標変換の場合の PTP 移動....450
7.10.6 例 3:PTPG0 および TRANSMIT....451
7.11 座標変換の選択時の制約事項....453
7.12 座標変換の解除(TRAFOOF)....455
8 キネマテイック結合....457
8.1 要素の削除(DELOBJ)....457
8.2 名称を使用したインデックスの特定(NAMETOINT)....462
9 キネマティック結合付きの衝突防止....465
9.1 干渉ペアをチェックします(COLLPAIR)....466
9.2 干渉回避のマシンモデルの再計算要求(PROTA)....468
9.3 プロテクションゾーン状態の設定(PROTS)....470
9.4 :2 つのプロテクションゾーンの間隔の決定(PROTD)....471
10 キネマティック結合による座標変換....473
10.1 座標変換の有効化(TRAFOON)....473
10.2 機械計測後の方向座標変換の変更(CORRTRAFO)....475
11 工具オフセット....485
11.1 オフセットメモリ....485
11.2 追加オフセット....489
11.2.1 追加オフセットの選択(DL)....489
11.2.2 摩耗値とセットアップ値の指定($TC_SCPxy[t,d]、$TC_ECPxy[t,d])....490
11.2.3 追加オフセットの削除(DELDL)....491
11.3 工具補正の特殊処理....493
11.3.1 工具長のミラーリング....495
11.3.2 摩耗の符号評価....495
11.3.3 動作中の加工運転の座標系(TOWSTD、TOWMCS、TOWWCS、TOWBCS、TOWTCS、TOWKCS)....496
11.3.4 工具長と平面変更....500
11.4 オンライン工具補正....502
11.4.1 多項式機能(FCTDEF)の定義....502

11.4.2 オンライン工具補正の連続書き込み(PUTFTOCF)....504
11.4.3 オンライン工具補正の書き込み、個別(PUTFTOC)....505
11.4.4 オンライン工具オフセットの起動/解除(FTOCON/FTOCOF)....506

11.5 3 次元工具径補正....507
11.5.1 3 次元外周加工用の 3 次元工具径補正(CUT3DC、CUT3DCD、ISD)....507
11.5.2 3 次元正面削りのための 3 次元工具径補正の選択(CUT3DF、CUT3DFS、CUT3DFF、CUT3DFD)....511
11.5.3 3 次元外周加工、限界面の考慮(CUT3DCC、CUT3DCCD)....517

11.6 工具オリエンテーション(ORIC、ORID、OSOF、OSC、OSS、OSSE、ORIS、OSD、OST)....523

11.7 D 番号の任意割り当て、刃先番号....530
11.7.1 D 番号、刃先番号の任意割り当て(CE アドレス)....530
11.7.2 D 番号の任意割り当て: D 番号のチェック(CHKDNO)....530
11.7.3 D 番号の任意割り当て: D 番号の名称変更(GETDNO、SETDNO)....531
11.7.4 D 番号の任意割り当て: 指定した D 番号に対して T 番号の特定(GETACTTD)....532
11.7.5 D 番号の任意割り当て: D 番号の無効化(DZERO)....533

11.8 工具ホルダキネマティクス....534

11.9 旋回工具ホルダの工具長補正(TCARR、TCOABS、TCOFR、TCOFRX、TCOFRY、TCOFRZ)....542

11.10 オンライン工具長補正(TOFFON、TOFFOF)....545

11.11 旋回工具のオフセットデータ変更....548
11.11.1 オリエンテーションの計算(ORISOLH)....548
11.11.2 旋回工具のオフセットデータの変更の起動(CUTMOD、CUTMODK)....561

11.12 工具環境の使用....571
11.12.1 工具環境の保存(TOOLENV)....571
11.12.2 工具環境の削除(DELTOOLENV)....574
11.12.3 T 番号、D 番号、および DL 番号の読み出し (GETTENV)....575
11.12.4 保存した工具環境に関する情報の読み取り($P_TOOLENVN、$P_TOOLENV)....577
11.12.5 工具長、工具長成分、またはその両方の読み取り(GETTCOR)....577
11.12.6 工具成分の変更(SETTCOR)....585

11.13 座標軸への工具長 L1、L2、L3 の割り当ての読み取り(LENTOAX)....602

12 軌跡の移動動作....607

12.1 法線方向制御....607
12.1.1 連結の定義(TANG)....607
12.1.2 挿入ブロックの生成の有効化(TLIFT)....609
12.1.3 連結を起動(TANGON)....610
12.1.4 連結の無効化(TANGOF)....612
12.1.5 連結の解除(TANGDEL)....612

12.2 送り速度特性(FNORM、FLIN、FCUB、FPO)....615

12.3 加減速動作....621
12.3.1 加減速モード(BRISK、BRISKA、SOFT、SOFTA、DRIVE、DRIVEA)....621
12.3.2 スレーブ軸に対する加減速の動作(VELOLIMA、ACCLIMA、JERKLIMA)....623
12.3.3 テクノロジ用のダイナミック応答値の起動(DYNNORM、DYNPOS、DYNROUGH、DYNSEMIFIN、DYNFINISH)....625
12.4 フィードフォワード制御による移動(FFWON、FFWOF)....628
12.5 プログラム指令可能な輪郭精度(CPRECON、CPRECOF)....629
12.6 解析メモリによるプログラム手順(STOPFIFO、STARTFIFO、FIFOCTRL、STOPRE)....631
12.7 停止遅延範囲(DELAYFSTON、DELAYFSTOF)の定義....634
12.8 SERUPRO のプログラム位置のマスク(IPTRLOCK、IPTRUNLOCK)....637
12.9 輪郭への再位置決め(REPOSA、REPOSL、REPOSQ、REPOSQA、REPOSH、REPOSHA、DISR、DISPR、RMIBL、RMBBL、RMEBL、RMNBL)....640
12.10 モーションコントロールへの影響....649
12.10.1 加々速度補正の割合(JERKLIM)....649
12.10.2 速度補正の割合(VELOLIM)....650
12.10.3 JERKLIM と VELOLIM のプログラム例....652
12.11 輪郭/旋回の許容範囲のプログラミング(CTOL、OTOL、ATOL)....653
12.12 有効な連結によるブロック変更動作(CPBC)....659
13 軸連結....661
13.1 連結移動(TRAILON、TRAILOF)....661
13.2 カーブテーブル(CTAB)....666
13.2.1 カーブテーブルの定義(CTABDEF、CATBEND)....666
13.2.2 カーブテーブルの有無の確認(CTABEXISTS)....673
13.2.3 カーブテーブルの削除(CTABDEL)....673
13.2.4 カーブテーブルの削除、および上書きを防止するロック(CTABLOCK、CTABUNLOCK)....675
13.2.5 カーブテーブル: カーブテーブル機能の特定(CTABID、CTABISLOCK、CTABMEMTYP、CTABPERIOD)....677
13.2.6 カーブテーブル値の読み取り(CTABTSV、CTABTEV、CTABTSP、CTABTEP、CTABSSV、CTABSEV、CTAB、CTABINV、CTABTMIN、CTABTMAX)....679
13.2.7 カーブテーブル: メモリ使用の確認(CTABNO、CTABNOMEM、CTABFNO、CTABSEGID、CTABSEG、CTABFSEG、CTABMSEG、CTABPOLID、CTABPOL、CTABFPOL、CTABMPOL)....684
13.3 軸間連動機能(LEADON、LEADOF)....687
13.4 電子ギヤ(EG)....693
13.4.1 電子ギヤの定義(EGDEF)....693
13.4.2 電子ギヤの起動(EGON、EGONSYN、EGONSYNE)....694

13.4.3 電子ギヤの起動(EGOFS、EGOFC)...698
13.4.4 電子ギヤの定義の解除(EGDEL)...699
13.4.5 毎回転送り速度(G95) / 電子ギヤ(FPR)...700
13.5 主軸同期...701
13.5.1 主軸同期: プログラミング(COUPDEF、COUPDEL、COUPON、COUPONC、COUPOF、COUPOFS、COUPRES、WAITC)...701
13.6 汎用連結(CP...)...713
13.7 マスタ/スレーブ連結(MASLDEF、MASLDEL、MASLON、MASLOF、MASLOFS)...724
14 シンクロナイズドアクション...729
14.1 シンクロナイズドアクションの定義...729
15 揺動...731
15.1 非同期揺動(OS、OSP1、OSP2、OST1、OST2、OSCTRL、OSNSC、OSE、OSB)...731
15.2 シンクロナイズドアクションによる揺動制御(OSCILL)...738
16 パンチングとニブリング...747
16.1 起動と解除...747
16.1.1 パンチングとニブリングの起動または解除(SPOF、SON、PON、SONS、PONS、PDELAYON、PDELAYOF、PUNCHACC):...747
16.2 自動軌跡分割...753
16.2.1 軌跡軸の軌跡分割...756
16.2.2 単独軸の軌跡分割...757
17 研削加工...759
17.1 研削用工具監視の有効化/無効化(TMON、TMOF)...759
18 その他の機能...761
18.1 軸機能(AXNAME、AX、SPI、AXTOSPI、ISAXIS、AXSTRING、MODAXVAL)...761
18.2 置換可能なジオメトリ軸(GEOAX)...764
18.3 軸コンテナ(AXCTSWE、AXCTSWED、AXCTSWEC)...769
18.4 有効な軸位置を待機します(WAITENC)...772
18.5 プログラム指令可能なパラメータセット切り替え(SCPARA)...774
18.6 現在の NC 言語の適用範囲の確認(STRINGIS)...776
18.7 対話形式によるパートプログラムからのウィンドウの呼び出し(MMC)...781
18.8 プログラム実行時間/ワークカウンタ...787
18.8.1 プログラム実行時間...787
18.8.2 ワークカウンタ...791

18.9 Process DataShare - 外部機器/ファイルへの出力(EXTOPEN、WRITE、EXTCLOSE):...793
18.10 アラーム(SETAL)...799
18.11 停止延長と退避(ESR)...801
18.11.1 NC 制御 ESR...803
18.11.1.1 NC 制御による退避(POLF、POLFA、POLFMASK、POLFMLIN)...803
18.11.1.2 NC 制御による停止...807
18.11.2 ドライブ統合 ESR...808
18.11.2.1 ドライブ内蔵の停止の設定(ESRS)...808
18.11.2.2 ドライブ内蔵の後退の設定(ESRS)...809
18.12 Define blank (WORKPIECE)...811
18.13 言語切り替えモード(G290, G291)...816
19 ユーザー荒削りプログラム...819
19.1 荒削りの機能のサポート...819
19.2 輪郭テーブルの生成(CONTPRON)...820
19.3 コード化された輪郭テーブルの生成(CONTDCON)...827
19.4 2 個の輪郭要素の交点の特定(INTERSEC)...831
19.5 テーブルの輪郭要素をブロック毎に実行(EXECTAB)...833
19.6 円弧データの計算(CALCDAT)...834
19.7 輪郭解析の解除(EXECUTE)...836
20 サイクルを外部的にプログラミング...837
20.1 テクノロジサイクル...837
20.1.1 はじめに...837
20.1.2 テクノロジ別の概要...839
20.1.3 HOLES1 - 直線位置決めパターン...841
20.1.4 HOLES2 - 円弧位置決めパターン...842
20.1.5 POCKET3 - 長方形ポケットのフライス削り...845
20.1.6 POCKET4 - 円形ポケットのフライス削り...849
20.1.7 SLOT1- 直線溝...853
20.1.8 SLOT2 - 円周溝...857
20.1.9 LONGHOLE - 長穴...860
20.1.10 CYCLE60 - 彫刻サイクル...863
20.1.11 CYCLE61 - 正面削り...866
20.1.12 CYCLE62 - 輪郭の呼び出し...869
20.1.13 CYCLE63 - 輪郭ポケットのフライス削り...870
20.1.14 CYCLE64 - 輪郭ポケットの前加工...874
20.1.15 CYCLE70 - ねじフライス削り...876
20.1.16 CYCLE72 - 輪郭フライス削り...878

20.1.17 CYCLE76 - 長方形スピゴットのフライス削り ..882
20.1.18 CYCLE77 - 円形スピゴットのフライス削り ..885
20.1.19 CYCLE78 - ドリルねじフライス削り ..888
20.1.20 CYCLE79 - 多角形 ..892
20.1.21 CYCLE81 - 穴あけ、センタリング ..895
20.1.22 CYCLE82 - 穴あけ、座ぐり ..896
20.1.23 CYCLE83 - 深穴あけ ..899
20.1.24 CYCLE84 - フローティングチャックなしのタッピング ..903
20.1.25 CYCLE85 - リーマ加工 ..909
20.1.26 CYCLE86 - ボーリング ..910
20.1.27 CYCLE92 - 突切り ..912
20.1.28 CYCLE95 - 輪郭切削 ..914
20.1.29 CYCLE98 - 連続ねじ ..917
20.1.30 CYCLE99 - ねじの旋削 ..923
20.1.31 CYCLE435 - ドレッサー座標系の設定 ..928
20.1.32 CYCLE495 - 成形研削 ..929
20.1.33 CYCLE800 - 旋回 ..931
20.1.34 CYCLE801 - 格子またはフレーム ..937
20.1.35 CYCLE802 - 任意の位置 ..939
20.1.36 CYCLE830 - 深穴ドリル 2 ..943
20.1.37 CYCLE832 - 高速設定 ..951
20.1.38 CYCLE840 - フローティングチャックによるタッピング ..955
20.1.39 CYCLE899 - オープン溝のフライス削り ..959
20.1.40 CYCLE930 - 溝削り ..962
20.1.41 CYCLE940 - アンダーカット形状 ..965
20.1.42 CYCLE951 - 荒削り ..969
20.1.43 CYCLE952 - 輪郭の溝削り ..973
20.1.44 CYCLE4071 - 反転点での切り込みによる長手方向の研削 ..980
20.1.45 CYCLE4072 - 反転点での切り込みによる長手方向の研削とキャンセル信号 ..982
20.1.46 CYCLE4073 - 連続切り込みによる長手方向の研削 ..985
20.1.47 CYCLE4074 - 連続切り込みによる長手方向の研削とキャンセル信号 ..987
20.1.48 CYCLE4075 - 反転点での切り込みによる平面研削 ..991
20.1.49 CYCLE4077 - 反転点での切り込みによる平面研削とキャンセル信号 ..993
20.1.50 CYCLE4078 - 連続切り込みによる平面研削 ..997
20.1.51 CYCLE4079 - 間欠切り込みによる平面研削 ..999
20.1.52 GROUP_BEGIN - プログラムブロック開始 ..1001
20.1.53 GROUP_END - プログラムブロック終了 ..1002
20.1.54 GROUP_ADDEND - 試し切り追加終了 ..1003
20.1.55 必要条件 ..1004
20.1.55.1 サイクル画面の機能スケーリング ..1004
20.2 計測サイクル ..1006
21 テーブル ..1007
21.1 命令 ..1007

21.2 命令:SINUMERIK 828D での適用 1061
21.2.1 制御タイプ フライス削り / 旋削 1061
21.2.2 制御タイプ 研削 1103
21.3 HMI で現在設定されている言語 1141
A 付録 1143
A.1 略語の一覧 1143
A.2 本書の概要 1155
用語集 1157
索引 1183

基本的な安全に関する指示事項 1

1.1 一般的な安全に関する指示事項

⚠ 警告

安全に関する情報および残存危険性に注意しない場合の死亡の危険性

関連するハードウェアの資料/文書にある安全に関する情報の遵守や存在する危険性に対する注視がなされていない場合、重大な傷害または死亡事故が発生する可能性があります。

- ハードウェアドキュメントに記載された安全に関する指示事項を遵守してください。
- リスク評価では残存危険性を考慮してください。

⚠ 警告

不正なまたは変更されたパラメータ設定による機械の誤作動

不正なまたは変更されたパラメータ設定により、傷害や死亡に至る機械の誤動作が発生する場合があります。

- 承認されないアクセスに対するパラメータ設定変更 (パラメータ割り付け) を保護してください。
- 適切な対策を講じることで、考えられる誤作動に対応します (例: 非常停止または非常電源遮断)。

1.2 アプリケーション例に対する保証と責任

アプリケーション例に拘束力はなく、設定、機器、または起こり得る不測の事態に関する完全性を主張するものではありません。アプリケーション例は、特定のカスタマソリューションを示したものではなく、代表的なタスクを支援することのみを目的にしています。

ユーザー自身が責任を持って本製品の適切な運用を確実なものとしてください。アプリケーション例は、機器の使用、取り付け、操作、および保守を行うときの安全な取扱いに対する責任からお客様を解放するものではありません。

1.3 産業セキュリティ

注記

産業セキュリティ

シーメンスでは, プラント, システム, 機械装置およびネットワークの安全な運転をサポートする産業セキュリティ機能を備えた製品およびソリューションを提供しています。

サイバー攻撃に対して, プラント, システム, 機械装置およびネットワークを保護するために, 総合的で最新の産業セキュリティコンセプトを実装し, 継続的に維持することが必要です。当社の製品およびソリューションは、このようなコンセプトの一要素を構成するものです。

お客様には, プラント、システム、機械装置およびネットワークへの不正なアクセスを防止する責任があります。このようなシステム、機械装置およびコンポーネントは、このような接続が必要な場合にのみ、必要に応じて、十分なセキュリティ対策を講じた上で(例: ファイアウォールとネットワークの細分化)、企業ネットワークまたはインターネットに接続してください。

実装可能な産業セキュリティ対策に関する関連情報については、以下をご覧ください。

産業セキュリティ (http://www.siemens.com/industrialsecurity)

シーメンスの製品およびソリューションは, 更にセキュリティレベルを高めるために, 継続的な開発が行われています。当社では、製品の更新が利用できるようになったらすぐに適用すること、および最新の製品バージョンを使用することを強く推奨しています。サポートされてない製品バージョンの使用、最新版への更新適用失敗は、お客様へのサイバー攻撃の危険性を高めることがあります。

製品のアップデート情報を受け取るには, 以下で Siemens Industrial Security RSS Feed を申し込んでください:

産業セキュリティ (http://www.siemens.com/industrialsecurity)

関連情報はインターネットから入手できます。

産業セキュリティ設定マニュアル (https://support.industry.siemens.com/cs/ww/en/view/108862708)

⚠ 警告

ソフトウェアの誤動作による安全でない運転状態

ソフトウェアの誤動作 (例: ウィルス, トロイの木馬, マルウェアまたはウォーム) は, 死亡, 重傷や物損に至る場合があるシステムにおける安全ではない運転状態の原因となる場合があります。

- 最新のソフトウェアを使用して下さい。
- オートメーションおよびドライブコンポーネントを, 据えつけられた機器または機械装置に対する総合的で最先端の産業セキュリティコンセプトに組み込んでください。
- 据えつけられたすべての製品を総合的な産業セキュリティコンセプトに確実に組み込むようにしてください。
- 適切な保護対策で, 例えば, ウィルススキャンで悪意のあるソフトウェアから交換可能な記憶媒体上に保存されたファイルを保護してください。
- 「ノウハウプロテクト」ドライブ機能を有効にすることで、不正な変更からドライブを保護してください。

フレキシブルな NC プログラミング 2

2.1 変数

特に算術機能とチェック命令を組み合わせてシステムデータおよびユーザーデータの領域からの変数を使用すると、非常に高いレベルの柔軟性を持った NC プログラムとサイクルの作成が可能になります。

- システムデータ
  システムデータには、システムで事前定義された変数が含まれています。これらの変数には定義された意味があります。これらの変数はおもにシステムソフトウェアで使用されます。ユーザーは NC プログラムおよびサイクルでこれらの変数の読み取りと書き込みができます。例:マシンデータ、セッティングデータ、システム変数。
  システムデータアイテムの意味は固定されていますが、ユーザーは再定義によって特定の制限内で特性を変更することができます。
  「システムデータ、ユーザーデータ、NC 命令(REDEF)の再定義 (ページ 40)」を参照してください
- ユーザーデータ
  ユーザーデータには、ユーザーによって定義された変数が含まれています。変数の意味は、ユーザーだけが定義できます。変数はシステムに評価されません。
  ユーザーデータは以下に分けられます。
  - 予約ユーザー変数
    予約ユーザー変数はシステム内で定義済みの変数で、その番号はマシンデータでパラメータ設定されています。この変数の特性は、ユーザーが変更可能です。
    「システムデータ、ユーザーデータ、NC 命令(REDEF)の再定義 (ページ 40)」を参照してください。
  - ユーザー定義変数
    ユーザー定義変数は、ユーザーが定義し、実行時まではシステムが作成しない変数です。その番号、データタイプ、可視性、およびその他の機能は、ユーザーが定義します。
    「ユーザー変数の定義(DEF) (ページ 33)」を参照してください

2.1.1 システムデータ

システムデータには、システムで事前定義され、NC プログラムとサイクルのなかで、コントローラの現在のパラメータ設定へのアクセス、加えて機械状態、制御状態、および処理状態へのアクセスを可能にする変数が含まれています。

先読み変数

先読み変数は先読み中に、つまり、この変数が含まれているブロックが解釈されるときに読み取りおよび書き込みされるシステムデータです。先読み変数は、先読み停止をおこないません。

メインラン変数

先読み変数はメインラン中に、つまり、この変数が含まれているブロックが実行されるときに読み取りおよび書き込みされるシステムデータです。以下にメインラン変数を示します。

- シンクロナイズドアクションでプログラム指令できる変数(読み取り/書き込み)
- NC プログラムでプログラム指令して、先読み停止ができる変数です(読み取り/書き込み)。
- NC プログラムでプログラム指令でき、値は先読み中に計算されますが、メインランまでは書き込まれない変数です(メインランと同期:書き込み専用)

接頭語方式

システムデータとそれ以外のデータを区別するために、システムデータの前には通常、$記号の後に 1 文字または 2 文字とアンダーバーのついた接頭語が付けられます。

$ + 1 文字	**意味データタイプ**
先読みデータ(先読み中に読み取り/書き込みがおこなわれるシステムデータ)	
$M	マシンデータ [1]
$S	セッティングデータ、プロテクションゾーン [1]
$T	工具管理データ
$P	プログラム指令値
$C	ISO サイクルのサイクル変数
$O	オプションデータ
R	R 変数(算術変数) [2]
メインランデータ(メインラン中に読み取り/書き込みがおこなわれるシステムデータ)	
$$M	マシンデータ [1]
$$S	セッティングデータ [1]
$A	現在のメインランデータ
$V	位置コントローラデータ

$ + 1 文字	意味データタイプ
$R	R 変数(算術変数) [2]
[1] 機械データとセッティングデータを先読み変数とメインラン変数のどちらとして扱うかは、データに付けた$記号が 1 つか 2 つかによって決まります。表記はその用途に応じて自由に選択できます。 [2] パートプログラム/サイクルで R 変数を先読み変数として使用するときは、たとえば R10 のように接頭語を省略してください。シンクロナイズドアクションでメインラン変数として使用するときは、$R10 のように、$記号を接頭語として書いてください。	

2 番目の文字	意味可視性
N	NC グローバル変数(NC)
C	チャネル別変数(チャネル)
A	軸別変数(軸)

必要条件

接頭語方式での例外

以下の変数方式は、上記で指定した接頭語方式とは異なっています。

- $TC_...:ここでは、2 番目の文字 C はチャネル別システム変数ではなく、工具ホルダ用システム変数を表しています(TC=工具ホルダ)。
- $P_ ...:チャネル別システム変数

シンクロナイズドアクションでマシンデータとセッティングデータに使用

マシンデータとセッティングデータをシンクロナイズドアクションで使用するときは、接頭語を使用して、マシンデータまたはセッティングデータを、先読みの実行とメインランのどちらに同期して読み取るまたは書き込むかを定義できます。

加工中にデータを変更しない場合は、先読みの実行に同期して読み取ることができます。このために、マシンデータまたはセッティングデータの接頭語は、1 個の$記号で書いてください。

```
ID=1 WHENEVER $AA_IM[z] < $SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

加工中にデータを変更する場合は、メインランに同期して読み取り/書き込みをおこなってください。このために、マシンデータまたはセッティングデータの接頭語は、2 個の $記号で書いてください。

```
ID=1 WHENEVER $AA_IM[z] < $$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

注記

マシンデータとセッティングデータの書き込み

マシンデータまたはセッティングデータの項目を書き込むときは、パートプログラム/サイクルの実行時に有効なアクセスレベルで書き込みが可能であり、データが直ちに有効となるように「IMMEDIATE」設定されていることが重要です。

参照先

すべてのシステム変数の機能の一覧については、以下の資料を参照してください。

『リストマニュアル』、「システム変数」

下記も参照

変数 (ページ 23)

2.1.2 予約ユーザー変数算術変数

2.1.2.1 チャネル別算術変数(R)

チャネル別算術変数または R 変数は、予約された名称 R のユーザー変数で、REAL データタイプの配列として定義されています。互換性の理由から、R[10]などの配列インデックスによる表記、および R10 などの配列インデックスなしの表記の両方を、R 変数に使用できます。

シンクロナイズドアクションを使用する場合は、$R10 のように、$記号を含む接頭語を使用してください。

構文

先読み変数として使用する場合:
R<n>
R[<式>]

メインラン変数として使用する場合:
$R<n>
$R[<式>]

意味

R:	パートプログラムなどで、先読み変数として使用する場合の識別子	
$R:	シンクロナイズドアクションなどで、メインラン変数として使用する場合の識別子	
	タイプ:	REAL
	値の範囲:	非指数表記の場合: ± (0.000 0001 ... 9999 9999) **注:** 小数点以下 8 桁までの数値を使用できます。
		指数表記の場合: ± ($1*10^{-300}$ ... $1*10^{+300}$) **注:** ● 表記: <仮数>EX<指数> 例: 8.2EX-3 ● 符号と小数点を含め、10 文字までの文字を使用できます。
<n>:	R 変数の番号	
	タイプ:	INT
	値の範囲:	0~MAX_INDEX **注:** MAX_INDEX は、次のように、R 変数で設定した番号から計算します。 MAX_INDEX = (MD28050 $MN_MM_NUM_R_PARAM) - 1
<式>:	配列インデックス 式の結果が INT データタイプ(INT、REAL、BOOL、CHAR)に変換できるかぎり、任意の式を配列インデックスとして使用できます。	

例

R 変数の割り当ておよび数学関数での R 変数の使用:

プログラムコード	コメント
R0=3.5678	; 先読み変数の割り当て
R[1]=-37.3	; 先読み変数の割り当て
R3=-7	; 先読み変数の割り当て
$R4=-0.1EX-5	; メインプログラムの実行での割り当て:R4 = -0.1 * 10^-5
$R[6]=1.874EX8	; メインプログラムの実行での割り当て:R6 = 1.874 * 10^8
R7=SIN(25.3)	; 先読み変数の割り当て
R[R2]=R10	; R 変数を使用した間接アドレス指定
R[(R1+R2)*R3]=5	; 数式を使用した間接アドレス指定
X=(R1+R2)	; R1 と R2 の合計から得られる位置に軸 X を移動します
Z=SQRT(R1*R1+R2*R2)	; 平方根位置(R1^2 + R2^2)に Z 軸を移動します

下記も参照

変数 (ページ 23)

2.1.2.2 グローバル算術変数(RG)

機能

チャネル別 R 変数に加えて、グローバル R 変数にアクセスすることができます。これらはコントロールユニット内に存在し、すべてのチャネルから読み取り、書き込むことができます。

グローバル R 変数は、たとえばチャネル間で情報を転送するために使用されます。また、別の例として、主軸から張り出した未処理部分など、すべてのチャネルに対して評価する必要のあるグローバルな設定に使用されます。

グローバル R 変数は、操作画面からまたは NC プログラムで先読み時に読み取りおよび書き込みされます。シンクロナイズドアクションおよびテクノロジサイクルは使用できません。

注記

グローバル R 変数の読み取りおよび書き込み時には、チャネル間の同期はおこなわれ**ません**。

読み取りおよび書き込みは先読み時に実行されるため、1 つのチャネルからの書き込み値が別のチャネルで有効になるタイミングは定義されません。

例:

チャネル 1 で、グローバル R 変数をループカウンタとしてループを実行します。チャネル 2 がこのグローバル R 変数に値を書き込み、これによってチャネル 1 でループが中止されます。ただし、チャネル 1 で先読みで読み取ることのできるすべてのループは、引き続き実行されます。ループの数は定義されず、チャネルの読み込みなどに依存します。

チャネル間の同期は、WAIT フラグなどにより、アプリケーションとして実行してください。

構文

NC プログラムでの書き込み

```
RG[<n>]=<値>
RG[<式>]=<値>
```

NC プログラムでの読み取り

```
R...=RG[<n>]
R...=RG[<式>]
```

意味

RG :	**グローバル** R 変数の NC アドレスのデフォルト名 **注記:** NC アドレスの名前は MD15800 $MN_R_PARAM_NCK_NAME を使用して設定できます。	
<n>:	グローバル R 変数の番号	
	タイプ:	INT
	データ範囲:	0 ... MAX_INDEX **注:** MAX_INDEX は、次のように、グローバル R 変数で設定した番号から計算します。 MAX_INDEX = (MD18156 $MN_MM_NUM_R_PARAM_NCK) - 1
<式>:	式の結果が INT データタイプ(INT、REAL、BOOL、CHAR)に変換できるかぎり、任意の式を配列インデックスとして使用できます。	
<値>:	グローバル R 変数の値	
	タイプ:	REAL
	値の範囲:	非指数表記の場合: ± (0.000 0001 ... 9999 9999) **注:** 小数点以下 8 桁までの数値を使用できます。
		指数表記の場合: ± ($1*10^{-300}$ ... $1*10^{+300}$) **注**: ● 表記: **<**仮数**>EX<**指数**>** 例: 8.2EX-3 ● 符号と小数点を含め、10 文字までの文字を使用できます。

2.1.3 予約ユーザー変数リンク変数

ネットワークでリンクされた NCU 間で周期的にデータ交換をおこなう「NCU リンク」機能処理では、リンク変数を使用できます。この変数を使用すると、リンク変数メモリへのデータ形式別のアクセスが容易になります。リンク変数メモリは、ユーザー/工作機械メーカがシステム用基準に、サイズとデータ構成の両方を定義します。

リンク通信を構成している場合、リンク変数は、リンクに関与するすべての NCU によりパートプログラムとサイクルで読み取りと書き込みが可能な、システムグローバルユーザー変数です。グローバルユーザー変数(GUD)とは異なり、リンク変数は、シンクロナイズドアクションでも使用できます。

動作中の NCU リンクのないシステムでは、リンク変数を制御装置でローカルに、グローバルユーザー変数(GUD)と一緒に追加のグローバルユーザー変数として使用できます。

構文

```
$A_DLB[<インデックス>]
$A_DLW[<インデックス>]
$A_DLD[<インデックス>]
$A_DLR[<インデックス>]
```

意味

$A_DLB:	BYTE データ形式のリンク変数(1 バイト)	
	データタイプ:	UINT
	値の範囲:	0 ... 255
$A_DLW:	WORD データ形式のリンク変数(2 バイト)	
	データタイプ:	INT
	値の範囲:	-32768 ... 32767
$A_DLD:	DWORD データ形式のリンク変数(4 バイト)	
	データタイプ:	INT
	値の範囲:	-2147483648 ... 2147483647
$A_DLR:	REAL データ形式のリンク変数(8 バイト)	
	データタイプ:	REAL
	値の範囲:	±(2,2*10^{-308} … 1,8*10^{+308})

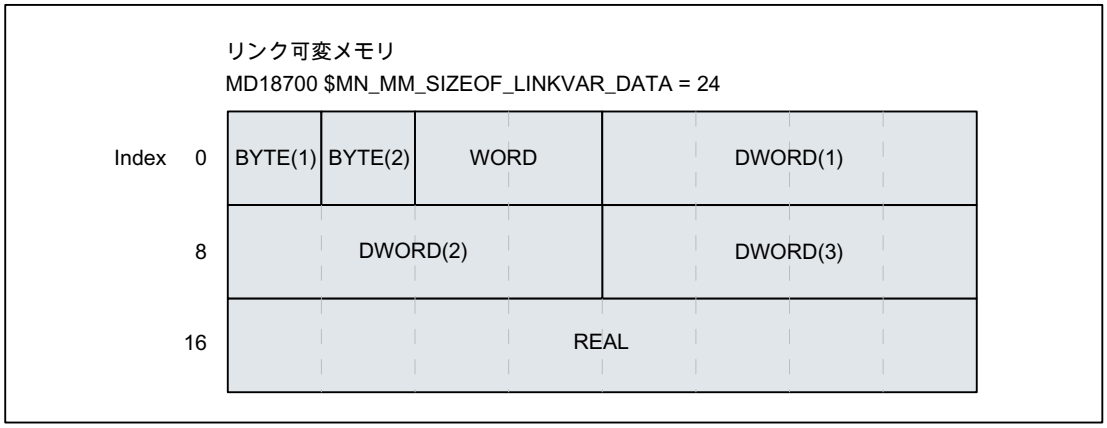
<インデックス>:	リンク変数メモリの先頭から数えたバイト単位のアドレスインデックス	
	データタイプ:	INT
	値の範囲:	0~MAX_INDEX **注** • MAX_INDEX は、次のように、パラメータ設定されたリンク変数メモリのサイズから計算します。 MAX_INDEX = (MD18700 $MN_MM_SIZEOF_LINKVAR_DATA) - 1 • リンク変数メモリにアドレス指定したバイト数がデータ形式の制限範囲内に配置されるよう、インデックスをプログラム指令してください。つまり、インデックス= n *バイト(ここでは n = 0、1、2、...)になります。 – $A_DLB[i]:i = 0、1、2、... – $A_DLW[i]:i = 0、2、4、... – $A_DLD[i]:i = 0、4、8、... – $A_DLR[i]:i = 0、8、16、...

例

オートメーションシステムに 2 基の NCU (NCU1 と NCU2)が含まれます。機械軸 AX2 が NCU1 に接続されています。この軸は、NCU2 のリンク軸として移動します。

NCU1 は、軸 AX2 の電流フィードバック値($VA_CURR)を周期的にリンク変数メモリに書き込みます。NCU2 は、リンク通信で周期的に伝送される電流フィードバック値を読み取り、制限値を超えた場合はアラーム 61000 を表示します。

リンク変数メモリのデータ構成を、次の図に示します。電流フィードバック値は、REAL 値で伝送されます。



NCU1

NCU1 はリンク変数$A_DLR[16]を使用して、軸 AX2 の電流フィードバック値を、内部的なシンクロナイズドアクションの補間サイクル中に周期的に、リンク変数メモリに書き込みます。

プログラムコード

```
N111 IDS=1 WHENEVER TRUE DO $A_DLR[16]=$VA_CURR[AX2]
```

NCU2

NCU2 はリンク変数$A_DLR[16]を使用して、軸 AX2 の電流フィードバック値を、内部的なシンクロナイズドアクションの補間サイクル中に周期的に、リンク変数メモリに読み込みます。電流フィードバック値が 23.0 A を超える場合は、アラーム 61000 が表示されます。

プログラムコード

```
N222 IDS=1 WHEN $A_DLR[16] > 23.0 DO SETAL(61000)
```

下記も参照

変数 (ページ 23)

2.1.4 ユーザー変数の定義(DEF)

DEF 命令を使用して、ユーザー固有の変数、すなわちユーザー変数(ユーザーデータ)を定義し、それに値を割り当てることができます。

有効範囲(つまり、変数を表示できる範囲)に応じて、ユーザー変数には次のカテゴリがあります。

- ローカルユーザー変数(LUD)
  ローカルユーザー変数(LUD)は、実行時にメインプログラム以外の NC プログラムで定義した変数です。この変数は、NC プログラムが呼び出されると作成され、プログラムリセットの終了時またはコントロールシステムの起動時に削除されます。ローカルユーザー変数には、この変数を定義した NC プログラム内でのみアクセスできます。
- プログラムユーザー変数(PUD)
  プログラムユーザー変数(PUD)は、メインプログラムとして使用される NC プログラムのなかで定義したユーザー変数です。この変数は、NC プログラムが呼び出されると作成され、プログラムリセットの終了時またはコントロールシステムの起動時に削除されます。PUD は、メインプログラム、およびメインプログラムのすべてのサブプログラムのなかでアクセスできます。

  注記

  プログラムグローバルユーザー変数(PUD)の使用可能性

  次のマシンデータを設定した場合は、メインプログラムで定義したプログラムグローバルユーザー変数(PUD)はサブプログラムにのみ表示されます。

  MD11120 $MN_LUD_EXTENDED_SCOPE = 1

  MD11120 = 0 の場合、メインプログラムで定義したプログラムグローバルユーザー変数はメインプログラムでのみ使用できます。
- グローバルユーザー変数(GUD)
  グローバルユーザー変数(GUD)は、データブロック(SGUD、MGUD、UGUD、GUD4 ～ GUD9)で定義され、プログラムリセット終了時、またはコントロールシステムの次回起動の後も保持される NC グローバル変数またはチャネルグローバル変数です。GUD は、すべての NC プログラムでアクセスできます。

ユーザー変数は、使用(読み取り/書き込み)前に定義してください。この場合は、以下の規則を守ってください。

- GUD は、定義ファイルで_N_DEF_DIR/_N_UGUD_DEF のように定義します。
- PUD と LUD は、NC プログラムの定義部で定義します。
- データは専用ブロックで定義してください。
- 各データ定義に使用できるのは、1 つのデータタイプのみです。
- 各データ定義で、複数の同じデータタイプの変数を定義できます。

構文

LUD と PUD
DEF <タイプ> <物理単位> <制限値> <名称>[<値 1>, <値 2>, <値 3>]=<初期値>

GUD
DEF <範囲> <先読み停止> <アクセス権> <データクラス> <タイプ> <物理単位> <制限値> <名称>[<値 1>, <値 2>, <値 3>]=<初期値>

意味

DEF:	GUD、PUD、LUD の各ユーザー変数の定義命令	
<範囲>:	適用範囲、GUD にのみ有効:	
	NC:	NC グローバルユーザー変数
	CHAN:	チャネルグローバルユーザー変数
先読み停止>:	先読み停止、GUD にのみ有効(任意選択)	
	SYNR:	読み取り時の先読み停止
	SYNW:	書き込み時の先読み停止
	SYNRW:	読み取り/書き込み時の先読み停止
アクセス権>:	NC プログラムまたは OPI を介した GUD の読み取り/書き込み保護レベル(任意選択)	
	APRP <保護レベル>:	読み取り:NC プログラム
	APWP <保護レベル>:	書き込み:NC プログラム
	APRB <保護レベル>:	読み取り:OPI
	APWB <保護レベル>:	書き込み:OPI
	<保護レベル>:	値の範囲:0 ... 7
	「属性: アクセス権(APR、APW、APRP、APWP、APRB、APWB) (ページ 51)」を参照してください	
<データクラス>:	データクラスの割り当て(SINUMERIK 828D のみ)	
	DCM:	データクラス M (= Manufacturer)
	DCI:	データクラス I (= Individual)
	DCU:	データクラス U (= User)
	「属性:データクラス(DCM、DCI、DCU) - SINUMERIK 828D のみ (ページ 57)」を参照してください	

<タイプ>:	データタイプ:	
	INT:	符合付き整数
	REAL:	実数(IEEE では LONG REAL)
	BOOL:	真偽値 TRUE (1)/FALSE (0)
	CHAR:	ASCII 文字
	STRING[<最大長>]:	定義された長さの文字列
	AXIS:	軸/主軸の識別子
	FRAME:	内部的な座標変換のジオメトリデータ
	「データタイプ (ページ 66)」を参照してください	
<物理単位>:	物理単位(任意選択)	
	PHU <単位>:	物理単位
	「属性: 物理単位(PHU) (ページ 49)」を参照してください	
<制限値>:	下限値/上限値(任意選択)	
	LLI <制限値>:	下限値(下限)
	ULI <制限値>:	上限値(上限)
	「属性: 制限値(LLI、ULI) (ページ 47)」を参照してください	
<名称>:	変数の名称 **注** ● 最大 31 文字 ● 先頭の 2 文字には、英字とアンダースコアのいずれかまたは両方を使用してください。 ● $記号はシステム変数として予約されているため、使用できません。	
[<値 1>, <値 2>, <値>]:	1 次元から 3 次元までの配列変数の配列サイズの指定(任意選択) 配列変数の初期化については、「配列変数の定義と初期化 (DEF、SET、REP) (ページ 60)」を参照してください。	
<初期値>:	初期値(任意選択) 「属性:初期値 (ページ 44)」を参照してください 配列変数の初期化については、「配列変数の定義と初期化 (DEF、SET、REP) (ページ 60)」を参照してください。	

例

例 1: 工作機械メーカ用のデータブロックのユーザー変数の定義

プログラムコード	コメント
%_N_MGUD_DEF	; GUD ブロック:工作機械メーカ
$PATH=/_N_DEF_DIR	
DEF CHAN REAL PHU 24 LLI 0 ULI 10 STROM_1, STROM_2	
; 概要	
; 2 つの GUD 項目の定義:STROM_1, STROM_2	
; 適用範囲:チャネル全体	
; データタイプ:REAL	
先読み停止:プログラム指令なし=>初期値=先読み停止なし	
; 物理単位:24 = [A]	
; 制限値:下限= 0.0、上限= 10.0	
; アクセス権:プログラム指令なし=>初期値= 7 =キー操作による切り替え位置 0	
; 初期値:プログラム指令なし=>初期値= 0.0	
DEF NCK REAL PHU 13 LLI 10 APWP 3 APRP 3 APWB 0 APRB 2 ZEIT_1=12, ZEIT_2=45	
; 概要	
; 2 つの GUD 項目の定義:ZEIT_1, ZEIT_2	
; 適用範囲:NC 全体	
; データタイプ:REAL	
先読み停止:プログラム指令なし=>初期値=先読み停止なし	
; 物理単位:13 = [s]	
; 制限値:下限= 10.0、上限=プログラム指令なし=>制限定義範囲の上限	
; アクセス権:	
; NC プログラム:書き込み/読み取り= 3 =エンドユーザー	
; OPI:書き込み= 0 =当社、読み取り= 3 =エンドユーザー	
; 初期値:ZEIT_1 = 12.0, ZEIT_2 = 45.0	
DEF NCK APWP 3 APRP 3 APWB 0 APRB 3 STRING[5] GUD5_NAME = "COUNTER"	
; 概要	
; 1 つの GUD 項目の定義:GUD5_NAME	
; 適用範囲:NC 全体	
; データタイプ:STRING、最大 5 文字	
先読み停止:プログラム指令なし=>初期値=先読み停止なし	
; 物理単位:プログラム指令なし=>初期値= 0 = 単位なし	
; 制限値:プログラム指令なし=>制限定義範囲:下限= 0、上限= 255	
; アクセス権:	
; NC プログラム:書き込み/読み取り= 3 =エンドユーザー	
; OPI:書き込み= 0 =当社、読み取り= 3 =エンドユーザー	
; 初期値:"COUNTER"	
M30	

例 2: グローバルプログラムとローカルユーザー変数(PUD/LUD)

プログラムコード	コメント
PROC MAIN	; メインプログラム
DEF INT VAR1	; PUD 定義
...	
SUB2	; サブプログラム呼び出し
...	
M30	

プログラムコード	コメント
PROC SUB2	; サブプログラム SUB2
DEF INT VAR2	; LUD 定義
...	
IF (VAR1==1)	; PUD を読み取ります
VAR1=VAR1+1	; PUD を読み取り、PUD に書き込みます
VAR2=1	; LUD に書き込みます
ENDIF	
SUB3	; サブプログラム呼び出し
...	
M17	

プログラムコード	コメント
PROC SUB3	; サブプログラム SUB3
...	
IF (VAR1==1)	; PUD を読み取ります
VAR1=VAR1+1	; PUD を読み取り、PUD に書き込みます
VAR2=1	; 異常:SUB2 からは LUD は未知です
ENDIF	
...	
M17	

例 3: データタイプ AXIS のユーザー変数の定義と用途

プログラムコード	コメント
DEF AXIS ABSCISSA	; 1 番目のジオメトリ軸
DEF AXIS SPINDLE	; 主軸
...	
IF ISAXIS(1) == FALSE GOTOF CONTINUE	
ABSCISSA = $P_AXN1	
CONTINUE:	
...	
SPINDLE=(S1)	; 1 番目の主軸

プログラムコード	コメント
OVRA[SPINDLE]=80	; 主軸オーバーライド = 80%
SPINDLE=(S3)	; 3 番目の主軸

必要条件

グローバルユーザー変数(GUD)

グローバルユーザー変数(GUD)を使用する場合は、次のマシンデータを考慮してください。

No.	識別子: $MN_	意味
11140	GUD_AREA_ SAVE_TAB	GUD ブロックの追加保存
18118 [1)]	MM_NUM_GUD_MODULES	アクティブファイルシステムの GUD ファイルの数
18120 [1)]	MM_NUM_GUD_NAMES_NCK	グローバル GUD の数
18130 [1)]	MM_NUM_GUD_NAMES_CHAN	チャネル別 GUD の数
18140 [1)]	MM_NUM_GUD_NAMES_AXIS	軸別 GUD の数
18150 [1)]	MM_GUD_VALUES_MEM	グローバル GUD 値のメモリロケーション
18660 [1)]	MM_NUM_SYNACT_GUD_REAL	REAL データタイプの設定可能な GUD の数
18661 [1)]	MM_NUM_SYNACT_GUD_INT	INT データタイプの設定可能な GUD の数
18662 [1)]	MM_NUM_SYNACT_GUD_BOOL	BOOL データタイプの設定可能な GUD の数
18663 [1)]	MM_NUM_SYNACT_GUD_AXIS	AXIS データタイプの設定可能な GUD の数
18664 [1)]	MM_NUM_SYNACT_GUD_CHAR	CHAR データタイプの設定可能な GUD の数
18665 [1)]	MM_NUM_SYNACT_GUD_STRING	STRING データタイプの設定可能な GUD の数

[1)] SINUMERIK 828D の場合は、MD は読み取りのみ可能です。

AXIS データタイプの NC グローバルユーザー変数のチャネル相互間の用途

軸識別子を使用したデータブロック内での定義のときに初期化された AXIS データタイプの NC グローバルユーザー変数は、この軸がチャネル間で同じチャネル軸番号を持つ場合にのみ、他の NC チャネルで使用できます。

そうでない場合は、この変数を、NC プログラムの開始時にロードするか、または次の例のように、AXNAME(...)機能(「軸機能(AXNAME、AX、SPI、AXTOSPI、ISAXIS、AXSTRING、MODAXVAL) (ページ 761)」を参照してください)を使用してください。

プログラムコード	コメント
DEF NCK STRING[5] ACHSE="X"	; データブロックで定義します。
...	
N100 AX[AXNAME(ACHSE)]=111 G00	; NC プログラムでの使用

2.1.5 システムデータ、ユーザーデータ、NC 命令(REDEF)の再定義

REDEF 命令は、システムデータ、ユーザーデータ、および NC 命令の属性を変更します。再定義の基本状態は、対応する定義の日付けが新しくなることです。

再定義のときは、複数の属性を同時に変更できません。変更する属性毎に、個別の REDEF 命令をプログラム指令してください。

複数の属性の同時変更をプログラム指令した場合は、常に最後の変更属性が有効になります。

属性値のリセット

REDEF によるアクセス権と初期化タイミングの変更の属性は、変数名称または NC 言語命令が後に続く REDEF を再プログラム指令することで初期値にリセットできます。

- アクセス権:保護レベル 7
- 初期化タイミング:現在値の初期化または保持はおこなわれません。

再定義可能な属性

「定義と再定義が可能な属性の概要 (ページ 58)」を参照してください

ローカルユーザー変数(PUD/LUD)

ローカルユーザー変数(PUD/LUD)は再定義できません。

構文

REDEF <名称> <先読み停止>

REDEF <名称> <物理単位>

REDEF <名称> <制限値>

REDEF <名称> <アクセス権>

REDEF <名称> <初期化タイミング>

```
REDEF <名称> <初期化タイミング> <初期値>
REDEF <名称> <データクラス>
REDEF <名称>
```

意味

REDEF:	特定の属性の再定義、または、システム変数、ユーザー変数、および NC 言語命令の「アクセス権」/「初期化タイミング」をリセットするための命令	
<名称>:	定義済み変数または NC 言語命令の名称	
<先読み停止>:	先読み停止	
	SYNR:	読み取り時の先読み停止
	SYNW:	書き込み時の先読み停止
	SYNRW:	読み取り/書き込み時の先読み停止
<物理単位>:	物理単位	
	PHU <単位>:	物理単位
	「属性: 物理単位(PHU) (ページ 49)」を参照してください **注:** 以下については再定義できません。 ● システム変数 ● 以下のデータタイプのグローバルユーザーデータ (GUD):BOOL、AXIS、STRING、FRAME	
<制限値>:	下限/上限	
	LLI <制限値>:	下限値(下限)
	ULI <制限値>:	上限値(上限)
	「属性: 制限値(LLI、ULI) (ページ 47)」を参照してください **注:** 以下については再定義できません。 ● システム変数 ● 以下のデータタイプのグローバルユーザーデータ (GUD):BOOL、AXIS、STRING、FRAME	

<アクセス権>:	パートプログラムまたは OPI による読み取り/書き込みアクセス権	
	APX <保護レベル>:	実行:NC 言語要素
	APRP <保護レベル>:	読み取り:パートプログラム
	APWP <保護レベル>:	書き込み:パートプログラム
	APRB <保護レベル>:	読み取り:OPI
	APWB <保護レベル>:	書き込み:OPI
	<保護レベル>:	値の範囲:0 ... 7
	「属性: アクセス権(APR、APW、APRP、APWP、APRB、APWB) (ページ 51)」を参照してください	
<初期化タイミング>:	変数が再初期化されるタイミング	
	INIPO:	電源投入
	INIRE:	メインプログラムの終了時、NC リセット時、または電源投入時
	INICF:	NEWCONF またはメインプログラム終了、NC リセットまたは電源投入
	PRLOC:	メインプログラムの終了時、部分的な変更後の NC リセット時、または電源投入時
	「属性:初期値 (ページ 44)」を参照してください	
<初期値>:	初期値 初期値を再定義するときは、初期化タイミングも常に指定してください(<初期化タイミング>を参照してください)。 「属性:初期値 (ページ 44)」を参照してください 配列変数の初期化については、「配列変数の定義と初期化(DEF、SET、REP) (ページ 60)」を参照してください。 **注** セッティングデータ以外のシステム変数に対しては再定義できません。	

<データクラス>:	データクラスの割り当て(SINUMERIK 828D のみ)	
	DCM:	データクラス M (= Manufacturer)
	DCI:	データクラス I (= Individual)
	DCU:	ータクラス U (= User)
	「属性:データクラス(DCM、DCI、DCU) - SINUMERIK 828D のみ (ページ 57)」を参照してください	

例

工作機械メーカ用のデータブロックのシステム変数$TC_DPCx の再定義

プログラムコード

```
%_N_MGUD_DEF                                    ; GUD ブロック:工作機械メーカ
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
```

N100 に 書き込みアクセス権:OPI =保護レベル 2、パートプログラム=保護レベル 3
ついて:

N200 に 物理単位[%]
ついて:

N300 に 下限値 = 0、上限値 = 200
ついて:

N400 に ; 配列変数は、POWER ON 時に 4 つの値で初期化されます。
ついて:

N800/ 「アクセス権」/「初期化タイミング」属性値のリセット
N900 に
ついて:

注記

ACCESS ファイルの使用

ACCESS ファイルを使用する場合、アクセス権の再定義を_N_MGUD_DEF から _N_MACCESS_DEF に再配置する必要があります。

必要条件

細分性

再定義は常に、名称で一義的に識別される変数全体に適用されます。たとえば、配列変数は、個々の配列要素への異なる属性の割り当てをサポートしていません。

2.1.6 属性:初期値

ユーザー変数の定義(DEF)

定義するときに、次のユーザー変数の初期値を事前に割り当てることができます。

- グローバルユーザー変数(GUD)
- プログラムユーザー変数(PUD)
- ローカルユーザー変数(LUD)

システム変数とユーザー変数の再定義(REDEF)

再定義するときに、次のユーザー変数の初期値を事前に割り当てることができます。

- システムデータ
  - セッティングデータ
- ユーザーデータ
  - R 変数
  - シンクロナイズドアクション変数($AC_MARKER、$AC_PARAM、$AC_TIMER)
  - シンクロナイズドアクション GUD (SYG_xy[ ]、ここで x=R、I、B、A、C、S および y=S、M、U、4 ～ 9)
  - EPS 変数
  - OEM 工具データ
  - OEM マガジンデータ
  - グローバルユーザー変数(GUD)

再初期化のタイミング

再定義のとき、変数を再初期化する(つまり、初期値にリセットする)タイミングを指定できます。

- INIPO (電源投入時)
  変数は電源投入時に初期化されます。
- INIRE (リセット)
  NC リセット時、モードグループリセット時、パートプログラム終了時(M02/M30)、または電源投入時に変数を最初期化します。
- INICF (NEWCONF)
  「マシンデータ有効設定」機能については、パートプログラム命令 NEWCONF またはNC リセット時、モードグループリセット時、パートプログラム終了時(M02/M30)、または電源投入時に HMI を介して変数が再初期化されます。
- PRLOC (プログラムの部分変更)
  パートプログラム実行中に変数が変更された場合に、NC リセット時、モードグループリセット時、パートプログラム終了時(M02/M30)にのみ変数を再初期化します。
  PRLOC 属性は、プログラム指令可能なセッティングデータと組み合わせた場合にのみ変更できます(以下の表を参照してください)。

表 2-1　プログラム指令可能なセッティングデータ

番号	識別子	G 命令 1)
42000	$SC_THREAD_START_ANGLE	SF
42010	$SC_THREAD_RAMP_DISP	DITS/DITE
42400	$SA_PUNCH_DWELLTIME	PDELAYON
42800	$SA_SPIND_ASSIGN_TAB	SETMS
43210	$SA_SPIND_MIN_VELO_G25	G25
43220	$SA_SPIND_MAX_VELO_G26	G26
43230	$SA_SPIND_MAX_VELO_LIMS	LIMS
43300	$SA_ASSIGN_FEED_PER_REV_SOURCE	FPRAON
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

番号	識別子	G 命令 1)
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)この G 命令は、セッティングデータをアドレス指定します。		

必要条件

初期値:グローバルユーザー変数(GUD)

- NC の適用範囲のグローバルユーザー変数(GUD)の初期化タイミングとして定義できるのは、INIPO (電源投入時)のみです。
- INIPO (電源投入時)に加えて、INIRE (リセット)または INICF (NEWCONF)を、CHAN の適用範囲のグローバルユーザー変数(GUD)の初期化タイミングとして定義できます。
- 適用範囲が CHAN で、初期化タイミングが INIRE(リセット)または INICF(NEWCONF)である、および NC リセット、モードグループリセット、および「マシンデータ有効化」のためのグローバルユーザー変数(GUD)では、名称を付けた事象が起動されたチャネルでのみ、変数を再初期化できます。

初期値:FRAME データタイプ

FRAME データタイプの変数には初期値を指定できません。FRAME データタイプの変数は、常に既定のフレームで自動的に初期化されます。

初期値:CHAR データタイプ

CHAR データタイプの変数の場合、ASCII コード(0... 255)ではなく、対応する ASCII 文字を引用符で囲んでプログラム指令できます(例: "A")。

初期値:データタイプ STRING

STRING データタイプの変数の場合、文字列は引用符で囲んでください(例: ...= "MACHINE_1")。

初期値:AXIS データタイプ

AXIS データタイプの変数の場合、拡張アドレス表記では、軸識別子を括弧で囲んでください(例: ...=(X3))。

初期値:システム変数

システム変数の場合、再定義を使用して、ユーザー用の初期値を定義することはできません。システム変数の初期値は、システムで指定されており、変更できません。ただし、再定義を使用して、システム変数を再初期化するタイミング(INIRE, INICF)を変更することはできます。

自動設定初期値:AXIS データタイプ

AXIS データタイプの変数の場合は、次の初期値を自動的に使用します。

- システムデータ:「1 番目のジオメトリ軸」
- シンクロナイズドアクション GUD(名称:SYG_A*)、PUD、LUD:
  マシンデータからの軸名称:MD20082 $MC_AXCONF_CHANAX_DEFAULT_NAME

自動設定初期値:工具とマガジンデータ

工具データとマガジンデータの初期値は、次のマシンデータを使用して定義できます。
MD17520 $MN_TOOL_DEFAULT_DATA_MASK

注記

同期制御

グローバル変数が異なる場所で読み取られる場合に、この変数の再初期化を起動する事象の同期制御については、そのユーザー/工作機械メーカの自己責任となります。

下記も参照

変数 (ページ 23)

2.1.7 属性: 制限値(LLI、ULI)

定義範囲の上限と下限は、次のデータタイプの場合にのみ定義できます。

- INT
- REAL
- CHAR

ユーザー変数の定義(DEF)制限値と自動設定初期値

上記のいずれかのデータタイプのユーザー変数を定義するときに、初期値を別途定義しない場合、変数はそのデータタイプの初期値へ自動的に設定されます。

- INT:0
- REAL:0.0
- CHAR:0

自動設定初期値が、プログラム指令した制限値で指定された定義範囲外にある場合は、自動設定初期値に最も近い制限値で変数が初期化されます。

- 自動設定初期値 **<** 下限値 (LLI) ⇒
  初期値 **=** 下限値
- 自動設定初期値 **>** 上限値 (ULI) ⇒
  初期値 **=** 上限値

例:

プログラムコード	コメント

`DEF REAL GUD1`	`; 下限値=制限定義範囲`
	`; 上限値=制限定義範囲`
	`; プログラム指令された初期値がありません`
	`; => 自動設定初期値 = 0.0`

`DEF REAL LLI 5.0 GUD2`	`; 下限値=5.0`
	`; 上限値=制限定義範囲`
	`; => 初期値 = 5.0`

`DEF REAL ULI -5 GUD3`	`; 下限値=制限定義範囲`
	`; 上限値= -5.0`
	`; => 初期値 = -5.0`

ユーザー変数の再定義(REDEF)制限値と現在値

ユーザー変数の制限値の再定義による変更のために、現在値が新しい定義範囲の範囲外となった場合は、アラームが発生して、制限値は拒否されます。

注記

ユーザー変数の再定義(REDEF)

ユーザー変数の制限値を再定義する場合、次の値と矛盾しないように変更してください。

- 制限値
- 現在値
- 再定義の初期値、および INIPO、INIRE、または INICF に基づいて自動再定義

下記も参照

変数 (ページ 23)

2.1.8 属性: 物理単位(PHU)

物的単位は、次のデータタイプの変数についてのみ指定できます。

- INT
- REAL

プログラム指令可能な物理単位(PHU)

物理単位は、固定小数点数形式で指定します。PHU <単位>

次の物理単位をプログラム指令できます。

<単位>	意味	物理単位
0	物理単位はありません	-
1	直線位置または回転位置 1)2)	[mm]、[inch]、[°]
2	直線位置 2)	[mm]、[inch]
3	回転位置	[°]
4	直線速度または回転速度 1) 2)	[mm/min]、[inch/min]、[rpm]
5	直線速度 2)	[mm/min]
6	回転速度	[rpm]
7	直線加減速度または回転加減速度 1)2)	[m/s^2]、[inch/s^2]、[rev/s^2]
8	直線加減速度 2)	[m/s^2]、[inch/s^2]
9	回転加減速度	[rev/s^2]
10	直線加々速度または回転加々速度 1)2)	[m/s^3]、[inch/s^3]、[rev/s^3]
11	直線加々速度 2)	[m/s^3]、[inch/s^3]
12	回転加々速度	[rev/s^3]
13	時間	[s]
14	位置制御ゲイン	[16.667/s]
15	毎回転送り速度 2)	[mm/rev]、[inch/rev]
16	温度補正 1)2)	[mm]、[inch]
18	推力	[N]

<単位>	意味	物理単位
19	質量	[kg]
20	慣性モーメント 3)	[kgm^2]
21	パーセント	[%]
22	周波数	[Hz]
23	電圧	[V]
24	電流	[A]
25	温度	[°C]
26	角度	[°]
27	Kv	[1000/min]
28	直線位置または回転位置 3)	[mm]、[inch]、[°]
29	切削速度 2)	[m/min]、[feet/min]
30	周速度 2)	[m/s]、[feet/s]
31	抵抗	[Ω]
32	インダクタンス	[mH]
33	トルク 3)	[Nm]
34	トルク定数 3)	[Nm/A]
35	電流制御ゲイン	[V/A]
36	速度制御ゲイン 3)	[Nm/(rad*s)]
37	速度	[rpm]
42	電力	[kW]
43	電流、低	[µA]
46	トルク、低 3)	[µNm]
48	1000 分の 1	-
49	-	[Hz/s]
65	流量	[ℓ/min]
66	圧力	[bar]
67	体積 3)	[cm^3]
68	制御対象ゲイン 3)	[mm/(V*min)]
69	トルク制限制御対象ゲイン	[N/V]
155	ねじリード 3)	[mm/rev]、[inch/rev]
156	ねじリードの可変量 3)	[mm/rev/rev]、[inch/rev/rev]

<単位>	意味	物理単位
1) 物理単位は軸タイプ(直線軸または回転軸)に依存します。		
2) 単位系の切り替え G70/G71(inch/metric) 長さに関係するシステム変数とユーザー変数への読み取り/書き込み操作で、G70/G71により単位系(MD10240 $MN_SCALING_SYSTEM_IS_METRIC)を変更した後は、その値は変換**されません**(現在値、初期値、および制限値)。 G700/G710(inch/metric) 長さに関係するシステム変数とユーザー変数への読み取り/書き込み操作で、G700/G710により単位系(MD10240 $MN_SCALING_SYSTEM_IS_METRIC)を変更した後に、その値が変換**されます**(現在値、初期値、および制限値)。		
3) 変数は、NC の現在の単位系(インチ/メトリック)に自動的に変換**されません**。変換については、ユーザー/工作機械メーカの自己責任となります。		

注記

書式変換によるレベルオーバーフロー

長さの物理単位を含むユーザー変数(GUD/PUD/LUD)の内部保存書式はすべてメトリックです。このようなタイプの変数を、シンクロナイズドアクションなどの NCK のメインランで過度に使用すると、単位系が切り替わるときに補間レベルで CPU 処理時間オーバーフローが発生し、アラーム 4240 が発生する場合があります。

注記

単位の互換性

変数を使用する(割り当て、比較演算、計算など)ときに、関連する単位の互換性は確認されません。変換が必要な場合は、ユーザー/工作機械メーカの自己責任となります。

下記も参照

変数 (ページ 23)

2.1.9 属性: アクセス権(APR、APW、APRP、APWP、APRB、APWB)

名称

アクセス属性 AP...の名称には以下が含まれます。

1. A:**A**ccess (アクセス)
2. P:**P**rotection (プロテクション)

3. R / W:**R**ead / **W**rite (読み取り/書き込み)
4. P / O:**P**rogram / **B**TSS (OPI)

アクセス権/アクセスレベル

プログラミングのときに指定が必要な次のアクセスレベルは、アクセス権に対応しています。

アクセス権	保護レベル
システムパスワード	0
工作機械メーカのパスワード	1
サービスパスワード	2
エンドユーザーのパスワード	3
キー操作のスイッチ位置 3	4
キー操作のスイッチ位置 2	5
キー操作のスイッチ位置 1	6
キー操作のスイッチ位置 0	7

ユーザーデータの定義(DEF)

次のデータには、アクセス権(APR.../APW...)を定義できます。

- グローバルユーザーデータ(GUD)

システムデータとユーザーデータの再定義(REDEF)

次のデータには、アクセス権(APR.../APW...)を再定義できます。

- システムデータ
  - マシンデータ

    注記

    マシンデータの読み取り権限の再定義

    マシンデータ読み取りの保護レベルは、パートプログラムと OPI に共通するキーワード APR によってのみ設定できます。

    キーワード APRP および APRB は読み取り権限の再定義によってサポートされていないため、割り込み 12490 のメッセージ「アクセス権 APRP/APRB <保護レベル> が設定されていません」が表示されます。
  - セッティングデータ
  - システム変数
  - プロセスデータ
  - マガジンデータ
  - 工具データ
- ユーザーデータ
  - R 変数
  - シンクロナイズドアクション変数($AC_MARKER、$AC_PARAM、$AC_TIMER)
  - シンクロナイズドアクション GUD (SYG_xy[ ]、ここで x=R、I、B、A、C、S 、および y=S、M、U、4 ～ 9)
  - EPS 変数
  - OEM 工具データ
  - OEM マガジンデータ
  - グローバルユーザー変数(GUD)

    注記

    再定義のときは、最低保護レベルの 7 から、1(工作機械メーカ)などの専用保護レベルの間で、変数に、アクセス権を自由に割り当てることができます。

NC 言語命令の再定義(REDEF)

次の NC 言語命令では、アクセス権または実行権(APX)を再定義できます。

- G 命令/準備機能
  参照先
  プログラミングマニュアル 基本編、セクション:G 命令/準備機能
- 予約機能
  参照先
  プログラミングマニュアル 基本編、セクション:「予約機能」の章
- 予約サブプログラム呼び出し
  参照先
  プログラミングマニュアル 基本編、セクション:「予約サブプログラム呼び出し」の章
- シンクロナイズドアクションによる DO 命令
- サイクルプログラム識別子
  サイクルはサイクルディレクトリに格納し、PROC 命令を指令してください。

NC プログラムとサイクル関連のアクセス権(APRP、APWP)

さまざまなアクセス権により、NC プログラムまたはサイクルからのアクセスが次のように簡単になります。

- APRP 0/APWP 0
  - NC プログラムを処理するときに、システムパスワードを設定してください。
  - サイクルは、_N_CST_DIR ディレクトリ(システム)に格納してください。
  - システムに対して、_N_CST_DIR ディレクトリの実行権を MD11160 $MN_ACCESS_EXEC_CST で設定してください。
- APRP 1/APWP 1 または APRP 2/APWP 2
  - NC プログラムを処理するときに、工作機械メーカまたはサービスのパスワードを設定してください。
  - サイクルは、_N_CMA_DIR (工作機械メーカ)または_N_CST_DIR ディレクトリに格納してください。
  - _N_CMA_DIR または_N_CST_DIR のディレクトリの実行権が工作機械メーカ以上になるように、マシンデータ MD11161 $MN_ACCESS_EXEC_CMA または MD11160 $MN_ACCESS_EXEC_CST にそれぞれ設定してください。

- APRP 3/APWP 3
  - NC プログラムを実行するときに、エンドユーザーパスワードを設定してください。
  - サイクルは、_N_CUS_DIR (エンドユーザー)、_N_CMA_DIR、または_N_CST_DIR ディレクトリに格納してください。
  - _N_CUS_DIR、_N_CMA_DIR または_N_CST_DIR のディレクトリの実行権がエンドユーザー以上になるように、マシンデータ MD11162 $MN_ACCESS_EXEC_CUS、MD11161 $MN_ACCESS_EXEC_CMA または MD11160 $MN_ACCESS_EXEC_CST にそれぞれ設定してください。
- APRP 4 ~ 7/APWP 4 ~ 7
  - NC プログラムを処理するときに、キー操作スイッチを 3 ～ 0 に設定してください。
  - サイクルは、ディレクトリ_N_CUS_DIR、_N_CMA_DIR、またはディレクトリ _N_CST_DIR に格納してください。
  - _N_CUS_DIR、_N_CMA_DIR または_N_CST_DIR のディレクトリの実行権が該当するキー操作スイッチ位置以上になるように、マシンデータ MD11162 $MN_ACCESS_EXEC_CUS、MD11161 $MN_ACCESS_EXEC_CMA または MD11160 $MN_ACCESS_EXEC_CST にそれぞれ設定してください。

OPI に関連するアクセス権(APRB、APWB)

アクセス権(APRB、APWB)によって、OPI を介したシステム変数とユーザー変数へのアクセスを、すべてのシステムコンポーネント(HMI、PLC、外部コンピュータ、EPS サービスなど)に対して一様に制限します。

注記

ローカル HMI アクセス権

システムデータへのアクセス権を変更するときは、変更したアクセス権が、HMI 操作で定義されたアクセス権と矛盾しないようにしてください。

APR/APW アクセス属性

互換性の理由から、属性 APR および APW は、属性 APRP/APRB および APWP/APWB に自動的に割り当てられます。

- APR x ⇒ APRP x APRB x
- APW y ⇒ APWP y APWB y

ACCESS ファイルを使用したアクセス権

ACCESS ファイルを使用してアクセス権を割り当てる場合は、システムデータ、ユーザーデータ、および NC 言語命令のアクセス権は ACCESS ファイルでのみ再定義してください。グローバルユーザーデータ(GUD)は例外です。このデータの場合は、アクセス権は対応する定義ファイル*_DEF で再定義してください。

アクセス保護が継続的におこなわれるように、実行権用のマシンデータ、および対応するディレクトリのアクセス保護を、矛盾がないように変更してください。

原則として、以下の手順を実行します。

1. 以下の、必要な定義ファイルの作成をおこないます。
   - _N_DEF_DIR/_N_SACCESS_DEF
   - _N_DEF_DIR/_N_MACCESS_DEF
   - _N_DEF_DIR/_N_UACCESS_DEF
2. 下記の定義ファイルの書き込み権限のマシンデータを、再定義のために必要な値へ設定します:
   - MD11170 $MN_ACCESS_WRITE_SACCESS = <保護レベル>
   - MD11171 $MN_ACCESS_WRITE_MACCESS = <保護レベル>
   - MD11172 $MN_ACCESS_WRITE_UACCESS = <保護レベル>
3. サイクルから保護要素にアクセスする場合は、サイクルディレクトリ_N_CST_DIR、_N_CMA_DIR、および_N_CST_DIR に対する実行権と書き込み権を変更してください。
   実行権
   - MD11160 $MN_ACCESS_EXEC_CST = <保護レベル>
   - MD11161 $MN_ACCESS_EXEC_CMA = <保護レベル>
   - MD11162 $MN_ACCESS_EXEC_CUS = <保護レベル>

   書き込み権
   - MD11165 $MN_ACCESS_WRITE_CST = <保護レベル>
   - MD11166 $MN_ACCESS_WRITE_CMA = <保護レベル>
   - MD11167 MN_ACCESS_WRITE_CUS = <保護レベル>

   実行権は、少なくとも、使用する要素の最高の保護レベルと同じ保護レベルに設定してください。
   書き込み権は、少なくとも、実行権と同じ保護レベルに設定してください。
4. ローカル HMI サイクルディレクトリの書き込み権は、ローカル NC サイクルディレクトリと同じ保護レベルに設定してください。
   参照先
   操作マニュアル

ACCESS ファイルでのサブプログラム呼び出し

さらにアクセス保護をおこなうために、サブプログラム(SPF または MPF 識別子)を ACCESS ファイルで呼び出すことができます。サブプログラムは、呼び出しをおこなう ACCESS ファイルの実行権を継承します。

注記

ACCESS ファイルで再定義できるのは、アクセス権のみです。他のすべての属性のプログラム指令/再定義は、対応する定義ファイルで続行してください。

下記も参照

変数 (ページ 23)

2.1.10 属性:データクラス(DCM、DCI、DCU) - SINUMERIK 828D のみ

セットアップ時のデータ処理を簡素化するために、機械および機械シリーズ、すべてのシステム、および NC のユーザーデータの一括セットアップとアップグレードは、データクラスに分類されています。

データクラス	データ
S = System	Siemens が提供するシステムデータ。マシンデータ、セッティングデータ、標準サイクル、計測サイクル、定義(SGUD)、マクロ(SMAC)など。
M = Manufacturer (工作機械メーカ)	機械シリーズ用のセットアップデータ。メーカーサイクル、定義(MGUD)、マクロ(MMAC)、機械の対象機能を定義するマシンデータなど。
I = Individual (機械別)	機械用のセットアップデータ。補正データ、レファレンス点オフセットなど。
U = User	機械の操作時に生成される機械用のデータ。工具データ、セッティングデータ、パートプログラム、ユーザーサイクル、定義(UGUD)、マクロ(UMAC)など。

参照先:

SINUMERIK 828D 試運転マニュアル 旋削加工とフライス加工;「データクラスの概要と使用」の章

ユーザーデータの定義(DEF)

データ項目のデータクラスは、ユーザーデータが定義されているファイルまたはディレクトリのデータクラスにより自動的に指定されます。データ項目のデータクラスは変更できません。

ただし、ユーザーデータの定義(DEF)については、**データ値**に対してデータ項目のデータクラスと異なるデータクラスを指定することができます。

データ項目のデータクラスには次のことが適用されます。

データ値のデータクラスの優先度≤データ項目のデータクラスの優先度

例:

プローブを定義する GUD の定義は、メーカーサイクルを実行する必要があるため、データクラス M (= Manufacturer)にします。ただし、プローブタイプは機械によって異なるため、データ項目の値はデータクラス I (= Individual)にします。

MGUD.DEF (データクラス M)

```
...
DEF CHAN DCI INT CALIPER
...
```

システムデータの再定義(REDEF)

システムデータのデータクラスは、再定義(REDEF)により変更できます。再定義は、定義ファイルでデータクラス S または M を使用して実行してください。

ACCESS ファイルを使用している場合、再定義は ACCESS ファイルでのみ実行できます。

機械の各データクラス、設定、オプションデータ、およびシステム変数については、以下のマニュアルを参照してください。

- リストマニュアル、マシンデータの詳細説明、パラメータ:「クラス」
- リストマニュアル、システム変数

2.1.11 定義と再定義が可能な属性の概要

次の表は、どのデータタイプに対して、どの属性が定義(DEF)、および/または再定義(REDEF)可能かを示します。

システムデータ

データタイプ	初期値	制限値	物理単位	アクセス権	データクラス (828D のみ)
マシンデータ	---	---	---	REDEF	REDEF
セッティングデータ	REDEF	---	---	REDEF	---
FRAME データ	---	---	---	REDEF	---
プロセスデータ	---	---	---	REDEF	---
ピッチ誤差補正(EEC)	---	---	---	REDEF	---
真直度補正(CEC)	---	---	---	REDEF	---
象限突起補償(QEC)	---	---	---	REDEF	---
マガジンデータ	---	---	---	REDEF	---
工具データ	---	---	---	REDEF	---
プロテクションゾーン	---	---	---	REDEF	---
工具ホルダ、旋回	---	---	---	REDEF	---
キネマテイック結合	---	---	---	REDEF	---
3 次元プロテクションゾーン	---	---	---	REDEF	---
作業領域リミット	---	---	---	REDEF	---

ユーザーデータ

データタイプ	初期値	制限値	物理単位	アクセス権	データクラス
R 変数	REDEF	REDEF	REDEF	REDEF	---
シンクロナイズドアクション変数($AC_...)	REDEF	REDEF	REDEF	REDEF	---
シンクロナイズドアクション GUD(SYG_...)	REDEF	REDEF	REDEF	REDEF	---
EPS 変数	REDEF	REDEF	REDEF	REDEF	---
OEM 工具データ	REDEF	REDEF	REDEF	REDEF	---
OEM マガジンデータ	REDEF	REDEF	REDEF	REDEF	---
グローバルユーザー変数(GUD)	DEF/REDEF	DEF	DEF	DEF/REDEF	DEF/REDEF
ローカルユーザー変数(PUD/LUD)	DEF	DEF	DEF	---	---

下記も参照

変数 (ページ 23)

2.1.12 配列変数の定義と初期化(DEF、SET、REP)

ユーザー変数を、1 次元から 3 次元までの配列として定義できます。

- 1 次元:DEF <データタイプ> <変数名称>[<n>]
- 2 次元:DEF <データタイプ> <変数名称>[<n>,<m>]
- 3 次元:DEF <データタイプ> <変数名称>[<n>,<m>,<o>]

注記

STRING データタイプのユーザー変数は、2 次元までの配列として定義できます。

データタイプ

ユーザー変数を、次のデータタイプの配列として定義できます。BOOL、CHAR、INT、REAL、STRING、AXIS、FRAME

配列要素への値の割り当て

次のタイミングで、値を配列要素に割り当てることができます。

- 配列定義のとき(初期値)
- プログラム実行中

値は、次の方法で割り当てることができます。

- 配列要素の明示的な指定
- 開始要素としての配列要素の明示的な指定と値リストの指定(SET)
- 開始要素としての配列要素の明示的な指定、および値とその繰り返し頻度の指定(REP)

注記

FRAME データタイプのユーザー変数には、初期値を割り当てることはできません。

構文(DEF)

```
DEF <データタイプ> <変数名称>[<n>,<m>,<o>]
DEF  STRING[<文字列長>] <変数名称>[<n>,<m>]
```

構文(DEF...=SET...)

値リストの使用:

- 定義のとき:
  DEF <データタイプ> <変数名称>[<n>,<m>,<o>]=SET(<値 1>,<値 2>, ...)
  以下も同等です。
  DEF <データタイプ> <変数名称>[<n>,<m>,<o>]=(<値 1>,<値 2>, ...)

  注記
  値リストによる初期化の場合は、SET を指定する必要はありません。

- 値の割り当てのとき:
  <変数名称>[<n>,<m>,<o>]=SET(<値 1>,<値 2>, ...)

構文(DEF...=REP...)

繰り返し値の使用:

- 定義のとき:
  DEF <データタイプ> <変数名称>[<n>,<m>,<o>]=REP(<値>)
  DEF <データタイプ> <変数名称>[<n>,<m>,<o>]=REP(<値>, <配列要素数>)
- 値の割り当てのとき:
  <変数名称>[<n>,<m>,<o>]=REP(<値>)
  DEF <データタイプ> <変数名称>[<n>,<m>,<o>]=REP(<値>,<配列要素数>)

意味

DEF:	変数を定義する命令
<データタイプ>:	変数のデータタイプ
	値の範囲: ● システム変数の場合: BOOL、CHAR、INT、REAL、STRING、AXIS ● GUD 変数または LUD 変数の場合: BOOL、CHAR、INT、REAL、STRING、AXIS、FRAME
<文字列長>:	STRING データタイプの最大文字数
<変数名称>:	変数名称
[<n>,<m>,<o>]:	配列サイズまたは配列インデックス

<n>:	1 次元の配列サイズまたは配列インデックス	
	タイプ:	INT (システム変数の場合は AXIS も)
	値の範囲:	最大配列サイズ:65535 配列インデックス:0 ≤ n ≤ 65534
<m>:	2 次元の配列サイズまたは配列インデックス	
	タイプ:	INT (システム変数の場合は AXIS も)
	値の範囲:	最大配列サイズ:65535 配列インデックス:0 ≤ m ≤ 65534
<o>:	3 次元の配列サイズまたは配列インデックス	
	タイプ:	INT (システム変数の場合は AXIS も)
	値の範囲:	最大配列サイズ:65535 配列インデックス:0 ≤ o ≤ 65534
SET:	指定した値リストによる値の割り当て	
(<値 1>,<値 2>, ...):	値リスト	
REP:	指定した<値>を使用した値割り当て	
<値>:	REP による初期化時に配列要素に書き込まれる値。	
<配列要素数>:	指定した<値>で書き込む配列要素数残りの配列要素には、そのタイミングに応じて、次の規則が適用されます。 ● 配列定義時の初期化: →残りの配列要素にゼロが書き込まれます。 ● プログラム実行中の割り当ての場合: →配列要素の現在値は変更されません。 このパラメータをプログラム指令しない場合は、すべての配列要素に<値>が書き込まれます。 このパラメータがゼロの場合は、そのタイミングに応じて、次の規則が適用されます。 ● 配列定義時の初期化: →すべての要素にゼロが事前に割り当てられます。 ● プログラム実行中の割り当ての場合: →配列要素の現在値は変更されません。	

配列インデックス

SET または REP による値の割り当ての場合などでは、配列要素の順序は自動的に、配列インデックスが右から左への繰り返しになります。

例:24 個の配列要素による 3 次元配列の初期化:

```
DEF INT FELD[2,3,4] = REP(1,24)
  FELD[0,0,0] = 1     1 番目の配列エレメント
  FELD[0,0,1] = 1     2 番目の配列エレメント
  FELD[0,0,2] = 1     3 番目の配列エレメント
  FELD[0,0,3] = 1     4 番目の配列エレメント
  ...
  FELD[0,1,0] = 1     5 番目の配列エレメント
  FELD[0,1,1] = 1     6 番目の配列エレメント
  ...
  FELD[0,2,3] = 1     12 番目の配列エレメント
  FELD[1,0,0] = 1     13 番目の配列エレメント
  FELD[1,0,1] = 1     14 番目の配列エレメント
  ...
  FELD[1,2,3] = 1     24 番目の配列エレメント
```

これは、以下に対応します。

```
FOR n=0 TO 1
  FOR m=0 TO 2
    FOR o=0 TO 3
      FELD[n,m,o] = 1
    ENDFOR
  ENDFOR
ENDFOR
```

例:配列変数全体の初期化

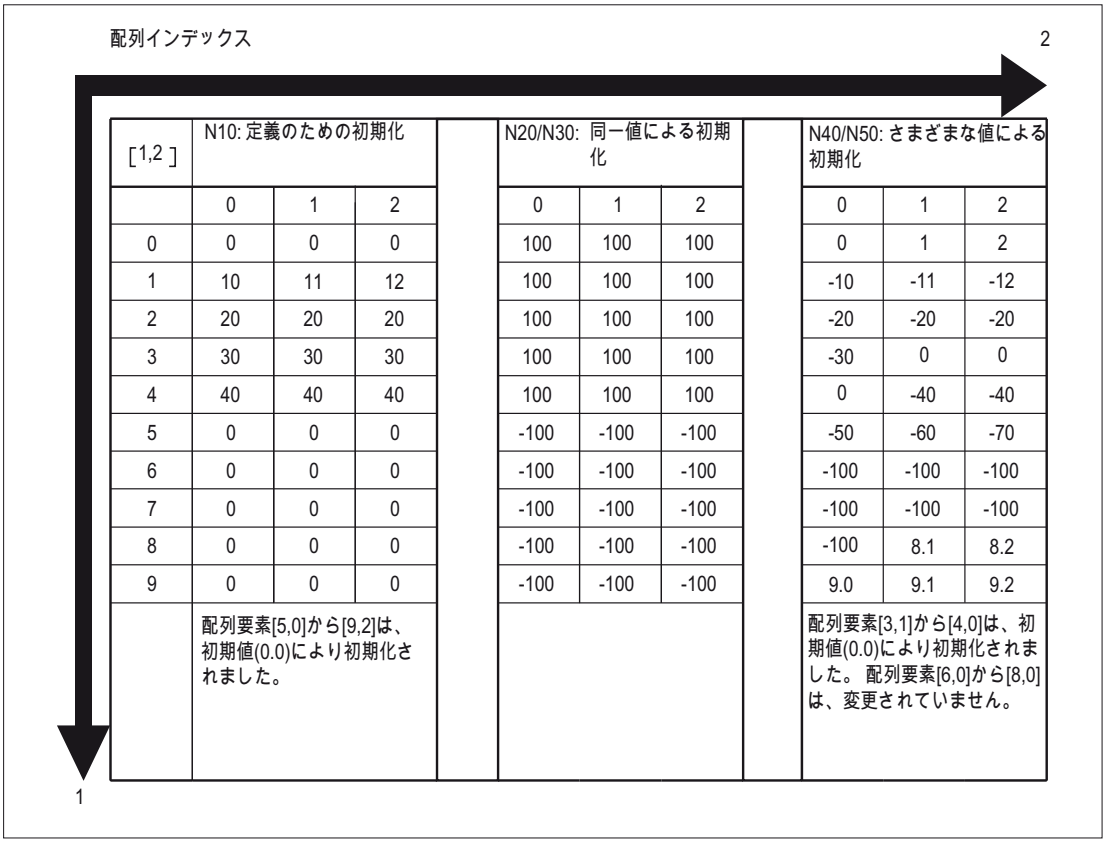
実際の割り当てについては、図を参照してください。

プログラムコード

```
N10 DEF REAL FELD1[10,3]=SET(0,0,0,10,11,12,20,20,20,30,30,30,40,40,40,)
```

プログラムコード

```
N20 ARRAY1[0,0] = REP(100)
N30 ARRAY1[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: 定義のための初期化		
	0	1	2
0	0	0	0
1	10	11	12
2	20	20	20
3	30	30	30
4	40	40	40
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
	配列要素[5,0]から[9,2]は、初期値(0.0)により初期化されました。		

N20/N30: 同一値による初期化		
0	1	2
100	100	100
100	100	100
100	100	100
100	100	100
100	100	100
-100	-100	-100
-100	-100	-100
-100	-100	-100
-100	-100	-100
-100	-100	-100

N40/N50: さまざまな値による初期化		
0	1	2
0	1	2
-10	-11	-12
-20	-20	-20
-30	0	0
0	-40	-40
-50	-60	-70
-100	-100	-100
-100	-100	-100
-100	8.1	8.2
9.0	9.1	9.2
配列要素[3,1]から[4,0]は、初期値(0.0)により初期化されました。 配列要素[6,0]から[8,0]は、変更されていません。		

下記も参照

配列変数の定義と初期化(DEF、SET、REP):詳細情報 (ページ 65)

変数 (ページ 23)

2.1.13 配列変数の定義と初期化(DEF、SET、REP):詳細情報

詳細情報(SET)

定義のときの初期化

- 1 番目の配列要素から、値リストにプログラム指令された要素と同じ配列要素のすべてに、値リストの値が割り当てられます。
- 値 0 は、値リストで明示的に値が宣言されずに配列要素に割り当てられます(値リストの隙間)。
- AXIS データタイプの変数の場合は、値リストの隙間は使用できません。
- 値リストに、定義した配列要素より多くの値が含まれる場合は、アラームが表示されます。

プログラム実行による値の割り当て

プログラム実行で値を割り当てる場合は、上記の定義の規則が適用されます。以下のオプションもサポートされています。

- 値リストでは、式も要素として使用できます。
- 値の割り当ては、プログラム指令した配列インデックスから開始されます。値を選択してサブ配列に割り当てることができます。

例:

プログラムコード	コメント
`DEF INT ARRAY[5,5]`	`; 配列を定義します。`
`ARRAY[0,0]=SET(1,2,3,4,5)`	`; 最初の 5 個の配列要素[0,0]～[0,4]への値の割り当て`
`FELD[0,0]=SET(1,2, , ,5)`	`; 最初の 5 個の配列要素[0,0]～[0,4]への隙間をもった値の割り当て、配列要素[0,2]、および[0,3] = 0`

`ARRAY[2,3]=SET(VARIABLE,4*5.6)`	`; 配列インデックス[2,3]から始まる、変数と式による値の割り当て:`
	`[2,3] = VARIABLE`
	`[2,4] = 4 * 5.6 = 22.4`

詳細情報(REP)

定義のときの初期化

- すべてまたは任意に選択した数の配列要素数を、指定した値(定数)で初期化します。
- FRAME データタイプの変数は初期化できません。

例:

プログラムコード	コメント
`DEF REAL varName[10]=REP(3.5,4)`	`; 配列を定義して、配列要素[0]～[3]を値 3.5 で初期化します`

プログラム実行による値の割り当て

プログラム実行で値を割り当てる場合は、上記の定義の規則が適用されます。以下のオプションもサポートされています。

- 値リストでは、式も要素として使用できます。
- 値の割り当ては、プログラム指令した配列インデックスから開始されます。値を選択してサブ配列に割り当てることができます。

例:

プログラムコード	コメント
`DEF REAL varName[10]`	`; 配列を定義します。`
`varName[5]=REP(4.5,3)`	`; 配列要素[5]～[7] = 4.5`
`R10=REP(2.4,3)`	`; R 変数 R10 ～ R12 = 2.4`
`DEF FRAME FRM[10]`	`; 配列を定義します。`
`FRM[5] = REP(CTRANS (X,5))`	`; 配列要素[5]～[9] = CTRANS(X,5)`

下記も参照

配列変数の定義と初期化(DEF、SET、REP) (ページ 60)

2.1.14 データタイプ

NC では、以下のデータタイプを使用できます。

データタイプ	意味	値の範囲
INT	符号付き整数	-2147483648 ... +2147483647
REAL	実数(IEEE では LONG REAL)	±(~2,2*10^{-308} … ~1,8*10^{+308})
BOOL	真偽値 TRUE (1)と FALSE (0)	1, 0
CHAR	ASCII 文字	ASCII コード 0 ～ 255
STRING	定義された長さの文字列	最大 200 文字の文字(特殊文字は使用できません)

AXIS	軸/主軸の識別子	チャネル軸識別子
FRAME	内部的な座標変換の形状パラメータ(平行移動、回転、スケーリング、ミラーリング)	---

自動データタイプ変換

割り当てのとき、およびパラメータ転送のときには、次のデータタイプ変換が可能で、自動的におこなわれます。

変換前↓/変換後 →	REAL	INT	BOOL
REAL	X	O	&
INT	X	X	&
BOOL	X	X	X
x:制限はなし O:; 変換後の値の範囲により、データ損失のおそれがあります⇒アラーム; 数値の四捨五入:小数点以下の値≥ 0.5 ⇒切り上げ、小数点以下の値< 0.5 ⇒切り捨て &:値≠ 0 ⇒ TRUE、値== 0 ⇒ FALSE			

下記も参照

変数 (ページ 23)

2.1.15 変数の有効性確認(ISVAR)

事前定義された ISVAR 機能を使用して、システム変数/ユーザー変数(マシンデータ、セッティングデータ、システム変数、GUD などの一般変数)が NC で既知であるかどうかを確認できます。

変数

確認対象の変数の構造は以下のようになります。

無次元変数:	<変数>
配列インデックス無しの 1 次元変数:	<変数>[ ]
配列インデックス n 有りの 1 次元変数:	<変数>[<n>]

配列インデックス無しの 2 次元変数: **<変数>[,]**

配列インデックス n および m 有りの 2 次元変数: **<変数>[<n>,<m>]**

構文

```
<結果>=ISVAR(<変数>[<n>,<m>])
```

意味

<結果>:	戻り値		
	データタイプ:	BOOL	
	値の範囲:	1	使用可能な変数
		0	未知の変数
ISVAR:	システム変数/ユーザー変数が NC で既知かどうかを確認します		
<変数>:	システム変数/ユーザー変数の名称		
	データタイプ:	STRING	
<n>:	第 1 次元の配列インデックス(**オプション**)		
	データタイプ:	INT	
<m>:	第 2 次元の配列インデックス(**オプション**)		
	データタイプ:	INT	

以下の確認が、転送パラメータに従っておこなわれます。

- 名称は既知か
- 変数は配列か
- 1 次元配列か、2 次元配列か
- それぞれの配列インデックスは許容範囲内か

すべての確認結果が「はい」の場合のみ、TRUE (1)が返されます。

確認結果が「いいえ」、またはシンタックスエラーが発生した場合、FALSE (0)が返されます。

例

プログラムコード	コメント
`DEF INT VAR1`	

プログラムコード	コメント
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("VAR1")	; この場合、IS_VAR は TRUE です。

プログラムコード	コメント
DEF REAL VARARRAY[10,10]	
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("VARARRAY[,]")	; この場合、IS_VAR は TRUE で 2 次元配列です。
N20 IS_VAR=ISVAR("VARARRAY")	; IS_VAR は TRUE で、変数が存在します。
N30 IS_VAR=ISVAR("VARARRAY[8,11]")	; IS_VAR は FALSE で、配列インデックスは使用できません。
N40 IS_VAR=ISVAR("VARARRAY[8,8")	; IS_VAR は FALSE で、「]」が抜けています (構文エラー)。
N50 IS_VAR=ISVAR("VARARRAY[,8]")	; IS_VAR は TRUE で、配列インデックスを使用できます
N60 IS_VAR=ISVAR("VARARRAY[8,]")	; ; IS_VAR は TRUE で、配列インデックスを使用できます

プログラムコード	コメント
DEF BOOL IS_VAR=FALSE	
N100 IS_VAR=ISVAR("$MC_GCODE_RESET_VALUES[1]"	; 転送パラメータはマシンデータ項目で、IS_VAR は TRUE です。

プログラムコード	コメント
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("$P_EP")	; この場合、IS_VAR は TRUE です。
N20 IS_VAR=ISVAR("$P_EP[X]")	; この場合、IS_VAR は TRUE です。

2.1.16 属性値/データタイプ(GETVARPHU、GETVARAP、GETVARLIM、GETVARDIM、GETVARDFT、GETVARTYP)の読み取り

システム変数/ユーザー変数の属性値は、予約機能 GETVARPHU、GETVARAP、GETVARLIM、および GETVARDFT を使用して、システム変数/ユーザー変数のデータタイプは、GETVARTYP を使用して読み取ることができます。

物理単位の読み取り

構文:
<結果>=GETVARPHU(<名称>)

意味

<結果>:	物理単位の数値		
	データタイプ:	INT	
	値の範囲:	「属性: 物理単位(PHU) (ページ 49)」の表を参照してください	
		障害時	
		- 2	指定した<名称>が、システムパラメータまたはユーザー変数に割り当てられていません。
GETVARPHU:	システム変数/ユーザー変数の物理単位の読み取り		
<名称>:	システム変数/ユーザー変数の名称		
	データタイプ:	STRING	

例:

NC は次の GUD 変数を含みます。

DEF CHAN REAL PHU 42 LLI 0 ULI 10000 electric

プログラムコード	コメント
DEF INT result=0	
result=GETVARPHU("electric")	; GUD 変数の物理単位を特定します。
IF (result < 0) GOTOF error	

結果として値 42 が返されます。これは物理単位[kW]に対応します。

注記

GETVARPHU を使用して、たとえば、2 つの変数が変数割り当て a = b で期待される物理単位を持つかどうかを確認できます。

読み取りアクセス権

構文:
<結果>=GETVARAP(<名称>,<アクセス>)

意味

<結果>:	指定した<アクセス>の保護レベル		
	データタイプ:	INT	
	値の範囲:	0 ... 7	「属性: アクセス権(APR、APW、APRP、APWP、APRB、APWB) (ページ 51)」を参照してください
		障害時	
		- 1	書き込みできません(<アクセス> "WP"および"WB"にのみ関連する)
		- 2	指定した<名称>が、システムパラメータまたはユーザー変数に割り当てられていません。
		- 3	<アクセス>の値が不正
GETVARAP:	システム変数/ユーザー変数へのアクセス権の読み取り		
<名称>:	システム変数/ユーザー変数の名称		
	データタイプ:	STRING	
<アクセス>:	アクセスのタイプ		
	データタイプ:	STRING	
	値の範囲:	"RP"	パートプログラムによる読み取り
		"WP"	パートプログラムによる書き込み
		"RB"	OPI による読み取り
		"WB"	OPI による書き込み

例:

プログラムコード	コメント
DEF INT result=0	
result=GETVARAP("$TC_MAP8","WB")	; OPI による書き込みに対するシステムパラメータ「マガジン位置」のアクセス保護を特定します。
IF (result < 0) GOTOF error	

結果として値 7 が返されます。これはキースイッチ位置 0(= アクセス保護なし)に対応します。

注記

GETVARAP を使用して、たとえば、アプリケーションで求められるアクセス権を確認する確認プログラムを実行できます。

制限値の読み取り

構文:
<ステータス>=GETVARLIM(<名称>,<制限値>,<結果>)

意味

<ステータス>:	機能状態		
	データタイプ:	INT	
	値の範囲:	1	OK
		-1	制限値が定義されていません (AXIS、STRING、FRAME タイプの変数に対して)
		-2	指定した<名称>が、システムパラメータまたはユーザー変数に割り当てられていません
		-3	<制限値>の値が不正です
GETVARLIM:	システム変数/ユーザー変数の下限値/上限値の読み取り		
<名称>:	システム変数/ユーザー変数の名称		
	データタイプ:	STRING	
<制限値>:	読み出す制限値を指定します		
	データタイプ:	CHAR	
	値の範囲:	"L":	= 下限値
		"U":	= 上限値
<結果>:	制限値を返します		
	データタイプ:	VAR REAL	

例:

プログラムコード	コメント
DEF INT state=0	
DEF REAL result=0	
state=GETVARLIM("$MA_MAX_AX_VELO","L",result)	MD32000 $MA_MAX_AX_VELO の下限値を特定します。
IF (result < 0) GOTOF error	

属性/データタイプの読み取り

構文:
<結果>=GETVARDIM(<名称>, インデックス)

意味

<結果>:	配列<インデックス>の次元/数	
	データタイプ:	INT
GETVARDIM:	システム変数/ユーザー変数の下限値/上限値の読み取り	
<名称>:	配列エレメントの数の読み取り	
	データタイプ:	STRING
<インデックス>:	配列の数、最大 3	
	データタイプ:	INT

例:

プログラムコード	コメント
N5 DEF REAL myReal[5,4]	
N10 R1 = GETVATDIM("myReal",1)	R1 = 5
N15 R2 = GETVATDIM("myReal",2)	R2 = 4

初期値の読み取り

構文:
<ステータス>=GETVARDFT(<名称>,<結果>[,<インデックス_1>,<インデックス_2>,<インデックス_3>])

意味

<ステータス>:	機能状態		
	データタイプ:	INT	
	値の範囲:	1	OK
		-1	使用可能な初期値がありません (例: <結果>の<名称>が間違ったタイプであるため)
		-2	指定した<名称>が、システムパラメータまたはユーザー変数に割り当てられていません
		-3	<インデックス_1>の値が不正、次元が 1 未満(= 配列なし = スカラー)
		-4	<インデックス_2>の値が不正
		-5	<インデックス_3>の値が不正
GETVARDFT:	システム変数/ユーザー変数の初期値の読み取り		
<名称>:	システム変数/ユーザー変数の名称		
	データタイプ:	STRING	
<結果>:	初期値を戻します		
	データタイプ:	VAR REAL (INT、REAL、BOOL、AXIS タイプの変数の初期値を読み取る場合)	
		VAR STRING (STRING と CHAR タイプの変数の初期値を読み取る場合)	
		VAR FRAME (FRAME タイプの変数の初期値を読み取る場合)	
<インデックス_1>:	1 次元へのインデックス(任意選択)		
	データタイプ:	INT	
	プログラム指令なし= 0		
<インデックス_2>:	2 次元へのインデックス(任意選択)		
	データタイプ:	INT	
	プログラム指令なし= 0		

<インデックス_3>:	3 次元へのインデックス(任意選択)	
	データタイプ:	INT
	プログラム指令なし= 0	

例:

プログラムコード	コメント
DEF INT state=0	
DEF REAL resultR=0	; INT、REAL、BOOL、AXIS タイプの初期値を確定する変数
DEF FRAME resultF=0	; FRAME タイプの初期値を確定する変数
IF (GETVARTYP("$MA_MAX_AX_VELO") <> 4) GOTOF error	
state=GETVARDFT("$MA_MAX_AX_VELO", resultR, AXTOINT(X))	; 「X」軸の初期値を特定します。
IF (result < 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)	

データタイプの読み取り

構文:
<結果>=GETVARTYP(<名称>)

意味

<結果>:	指定したシステム変数/ユーザー変数のデータタイプ		
	データタイプ:	INT	
	値の範囲:	1	= BOOL
		2	= CHAR
		3	= INT
		4	= REAL
		5	= STRING
		6	= AXIS
		7	= FRAME
		障害時	
		< 0	指定した<名称>が、システムパラメータまたはユーザー変数に割り当てられていません。
GETVARTYP:	システム変数/ユーザー変数のデータタイプの読み取り		
<名称>:	システム変数/ユーザー変数の名称		
	データタイプ:	STRING	

例:

プログラムコード	コメント
`DEF INT result=0`	
`DEF STRING name="R"`	
`result=GETVARTYP(name)`	`; R 変数のタイプを特定します。`
`IF (result < 0) GOTOF error`	

結果として値 4 が返されます。これは REAL データタイプに対応します。

2.2 間接プログラミング

2.2.1 アドレスの間接プログラミング

アドレスを間接的にプログラム指令すると、拡張アドレス(<インデックス>)が、適切なタイプの変数に置き換えられます。

注記

次のアドレスは間接的にはプログラム指令できません。

- N(ブロック番号)
- L (サブプログラム)
- 設定可能アドレス
  (例えば、X[1]の代わりに X1 は使用できません)

構文

```
<ADDRESS>[<インデックス>]
```

意味

<ADDRESS>[...]:	拡張子付き固定アドレス(インデックス)
<インデックス>:	変数、たとえば主軸番号、軸の変数など

例

例 1:主軸番号の間接プログラミング

直接プログラミング:

プログラムコード	コメント
S1=300	; 主軸番号 1 の速度(rpm 単位)。

間接プログラミング:

プログラムコード	コメント
DEF INT SPINU=1	; INT タイプの変数と値の割り当てを定義しています。
S[SPINU]=300	; SPINU 変数に保存された番号の主軸の速度が 300 rpm です(この例 1 では、番号が 1 の主軸です)。

例 2 :軸の間接プログラミング

直接プログラミング:

プログラムコード	コメント
FA[U]=300	; 軸「U」の送り速度 300。

間接プログラミング:

プログラムコード	コメント
DEF AXIS AXVAR2=U	; AXIS タイプの変数と値の割り当てを定義しています。
FA[AXVAR2]=300	; AXVAR2 という名称の変数に保存されたアドレス名称の軸の送り速度が、300 です。

例 3 :軸の間接プログラミング

直接プログラミング:

プログラムコード	コメント
$AA_MM[x]	; 軸「X」のプローブ計測値 (MCS) を読み取ります。

間接プログラミング:

プログラムコード	コメント
DEF AXIS AXVAR3=X	; AXIS タイプの変数と値の割り当てを定義しています。
$AA_MM[AXVAR3]	; 変数 AXVAR3 に保存された名称のプローブ計測値 (MCS) を読み取ります。

例 4 :軸の間接プログラミング

直接プログラミング:

プログラムコード
X1=100 X2=200

間接プログラミング:

プログラムコード	コメント
DEF AXIS AXVAR1 AXVAR2	; AXIS タイプの変数 2 個を定義しています。
AXVAR1=(X1) AXVAR2=(X2)	; 軸名称を割り当てています。
AX[AXVAR1]=100 AX[AXVAR2]=200	; AXVAR1 と AXVAR2 という名称の変数に保存されたアドレス名称の軸を移動しています。

例 5 :軸の間接プログラミング

直接プログラミング:

プログラムコード
G2 X100 I20

間接プログラミング:

プログラムコード	コメント
DEF AXIS AXVAR1=X	; AXIS タイプの変数と値の割り当てを定義しています。
G2 X100 IP[AXVAR1]=20	; 軸の中心点データの間接プログラミング。この軸のアドレス名称は AXVAR1 という名称の変数に保存されています。

例 6 :配列要素の間接プログラミング

直接プログラミング:

プログラムコード	コメント
DEF INT ARRAY1[4,5]	; 配列 1 を定義しています

間接プログラミング:

プログラムコード	コメント
DEFINE DIM1 AS 4	; 配列次元の場合は、配列サイズを固定値で指定してください。
DEFINE DIM2 AS 5	
DEF INT ARRAY[DIM1,DIM2]	
ARRAY[DIM1-1,DIM2-1]=5	

例 7 :間接サブプログラム呼び出し

プログラムコード	コメント
CALL "L" << R10	; R10 に置かれている番号のプログラムを呼び出します (文字列のカスケード化)。

2.2.2 G 命令の間接プログラム

G 命令の間接プログラムにより、サイクルを効率的にプログラム指令できます。

構文

G[<グループ>]=<番号>

意味

G[...]:	拡張子付き G 命令(インデックス)	
<グループ>:	インデックスパラメータ:G グループ	
	タイプ:	INT

<番号>:	G 命令番号の変数	
	タイプ:	INT または REAL

注記

通常、間接プログラム指令ができるのは、構文を特定しない G 命令のみです。

構文を特定する G 命令では、G グループ 1 のみが可能です。
構文が特定される G グループ 2、3、4 の G 命令は使用できません。

注記

間接 G 命令プログラム指令では、算術機能は使用できません。G 命令番号を計算する必要がある場合は、間接 G 命令プログラムの前に別のパートプログラム行で計算してください。

例

例 1:設定できるゼロオフセット(G グループ 8)

プログラムコード	コメント
N1010 DEF INT INT_VAR	
N1020 INT_VAR=2	
...	
N1090 G[8]=INT_VAR G1 X0 Y0	;G54
N1100 INT_VAR=INT_VAR+1	; G 命令演算
N1110 G[8]=INT_VAR G1 X0 Y0	;G55

例 2 : レベル選択(G グループ 6)

プログラムコード	コメント
N2010 R10=$P_GG[6]	; G グループ 6 の有効な G 命令を読み取り
...	
N2090 G[6]=R10	

参照先

G グループについて詳しくは、
プログラミングマニュアル基本編、「G 機能グループ」の章を参照してください。

2.2.3 位置属性の間接プログラミング(GP)

軸位置のインクレメンタル指令やアブソリュート指令などの位置属性は、キーワード BP と組み合わせて、変数として間接的にプログラム指令できます。

用途

位置属性の間接プログラミングは**置換サイクル**で使用します この場合、位置属性のキーワード(IC、AC、...)としてのプログラミングに、次の特長があります。

変数による間接プログラミングでは、**CASE** 命令が不要となります。間接プログラミング以外の場合は、この命令が、考えられるすべての位置属性への分岐のために必要になります。

構文

```
<POSITIONING COMMAND>[<軸/主軸>]=
BP(<位置>,<位置属性)
<軸/主軸>=BP(<位置>,<位置属性)
```

意味

<POSITIONING COMMAND>[]:	次の位置決め命令を、キーワード BP とともにプログラム指令できます。 POS、POSA、SPOS、SPOSA 以下もプログラムできます。 ● チャネルに存在するすべての軸と主軸の識別子: <軸/主軸> ● 可変軸/主軸識別子 AX
<軸/主軸>:	位置決めされる軸/主軸
GP():	位置決めのキーワード
<位置>:	パラメータ 1 軸/主軸位置を表わす定数または変数
<位置属性>:	パラメータ 2 位置属性(位置アプローチモードを表わす変数 ($P_SUB_SPOSMODE など)やキーワード(IC、AC、...)など)

変数で返される値には、次の意味があります。

値	意味	使用できる命令、他
0	位置属性への変更なし	
1	AC	POS、POSA、SPOS、SPOSA、AX、軸アドレス
2	IC	POS、POSA、SPOS、SPOSA、AX、軸アドレス
3	DC	POS、POSA、SPOS、SPOSA、AX、軸アドレス
4	ACP	POS、POSA、SPOS、SPOSA、AX、軸アドレス
5	ACN	POS、POSA、SPOS、SPOSA、AX、軸アドレス
6	OC	-
7	PC	-
8	DAC	POS、POSA、AX、軸アドレス
9	DIC	POS、POSA、AX、軸アドレス
10	RAC	POS、POSA、AX、軸アドレス
11	RIC	POS、POSA、AX、軸アドレス
12	CAC	POS、POSA
13	CIC	POS、POSA
14	CDC	POS、POSA
15	CACP	POS、POSA
16	CACN	POS、POSA

例

マスタ主軸 S1 とスレーブ主軸 S2 の間で動作中の主軸同期連結に対して、主軸を位置決めするために、次の置換サイクルがメインプログラムで SPOS 命令を使用して呼び出されます。

位置決めは、N2230 の命令を使用して実現されます。

```
SPOS[1]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE)
SPOS[2]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE)
```

アプローチする位置をシステム変数$P_SUB_SPOSIT から読み取ります。位置アプローチモードをシステム変数$P_SUB_SPOSMODE から読み取ります。

プログラムコード	コメント
N1000 PROC LANG_SUB DISPLOF SBLOF	
...	
N2100 IF($P_SUB_AXFCT==2)	
N2110	; 動作中の主軸同期連結に対する SPOS/SPOSA/M19 命令の置換
N2185 DELAYFSTON	; 停止遅延領域の始まり
N2190 COUPOF(S2,S1)	; 主軸同期連結の解除
N2200	; マスタ主軸とスレーブ主軸の位置決め
N2210 IF($P_SUB_SPOS==TRUE) OR ($P_SUB_SPOSA==TRUE)	
N2220	; 主軸を、次のように SPOS で位置決めします
N2230 SPOS[1]=GP($P_SUB_SPOSIT, $P_SUB_SPOSMODE)	
SPOS[2]=GP($P_SUB_SPOSIT, $P_SUB_SPOSMODE)	
N2250 ELSE	
N2260	; 主軸を M19 で位置決めします
N2270 M1=19 M2=19	; マスタ主軸とスレーブ主軸の位置決め
N2280 ENDIF	
N2285 DELAYFSTOF	; 停止遅延領域の終わり
N2290 COUPON(S2,S1)	; 主軸同期連結を起動します
N2410 ELSE	
N2420	; さらに置換をおこなうか確認します
...	
N3300 ENDIF	
...	
N9999 RET	

必要条件

- シンクロナイズドアクションでは、位置属性の間接プログラミングは使用できません。

参照先

機能マニュアル 基本機能; モードグループ、チャネル、プログラム運転、リセット応答(K1)、「サブプログラムによる NC 機能の置き換え」の章

2.2.4 パートプログラム行の間接プログラミング(EXECSTRING)

パートプログラム命令 EXECSTRING を使用すると、事前に生成した文字列変数をパートプログラム行として実行できます。

構文

EXECSTRING は、次のように個別のパートプログラム行にプログラム指令します。
EXECSTRING (<文字列変数>)

意味

EXECSTRING:	文字列変数をパートプログラム行として実行する命令
<文字列変数>:	実際に実行するパートプログラム行を含んだ STRING タイプの変数

注記

EXECSTRING を使用すると、**パートプログラムのプログラム区間**にプログラム指令できるパートプログラム構成は、フロー制御 (ページ 131)を除き、すべて抽出できます。これは、PROC と DEF 命令が、INI ファイルと DEF ファイルでの一般的な使用法と同じように除外されることを意味します。

例

```
プログラムコード                                    コメント
N100 DEF STRING[100] MY_BLOCK                      ; 実行するパートプログラム行を受け取る文字列変数の定義です。
N110 DEF STRING[10] MFCT1="M7"
...
N200 EXECSTRING(MFCT1 << "M4711")                  ; パートプログラム行「M7 M4711」を実行します。
...
N300 R10=1
N310 MY_BLOCK="M3"
N320 IF(R10)
N330 MY_BLOCK = MY_BLOCK << MFCT1
N340 ENDIF
N350 EXECSTRING(MY_BLOCK)                          ; パートプログラム行「M7 M3」を実行します。
```

2.3 算術機能

演算子/算術機能	意味
+	加算
-	減算
*	乗算
/ [1]	除算 [1]
DIV [1]	整数除算 [1]
MOD [1]	モジュロ除算(整数除算の余りを返します) [1]
:	FRAME 変数用結合演算子
Sin()	サイン
COS()	コサイン
TAN()	タンジェント
ASIN()	アークサイン
ACOS()	アークコサイン
ATAN2(,) [1]	アークタンジェント 2 [1]
SQRT()	平方根
ABS()	絶対値
POT()	2 乗(平方)
TRUNC()	整数要素 比較演算命令の精度は TRUNC で設定できます(「比較演算誤差の精度補正(TRUNC) (ページ 91)」を参照してください)
ROUND()	整数になるように四捨五入
LN()	自然対数
EXP()	指数関数
MINVAL ()	2 つの変数の小さい方の値 (「変数の最小値、最大値、および範囲(MINVAL、MAXVAL、および BOUND) (ページ 93)」を参照してください)

MAXVAL ()	2 つの変数の大きい方の値 (「変数の最小値、最大値、および範囲(MINVAL、MAXVAL、および BOUND) (ページ 93)」を参照してください)
BOUND ()	定義した数値の範囲内の変数値 (「変数の最小値、最大値、および範囲(MINVAL、MAXVAL、および BOUND) (ページ 93)」を参照してください)
CTRANS()	オフセット
CROT ()	回転
CSCALE()	スケーリングの変更
CMIRROR()	ミラーリング
1)「例」の節を参照してください	

プログラミング

算術機能には、通常の数学表記を使用します。実行の優先度を括弧で示します。角度は、三角関数と、その逆三角関数で指定します(直角= 90°)。

例

除算: /

(REAL タイプ) = INT タイプまたは REAL タイプ) / (INT タイプまたは REAL タイプ);

例:3 / 4 = 0.75

整数除算:DIV

(INT タイプ) = (INT タイプまたは REAL タイプ) / (INT タイプまたは REAL タイプ);

例:7 DIV 4.1 = 1

モジュロ除算(整数除算の余りを返します):MOD

(REAL タイプ) = (INT タイプまたは REAL タイプ) MOD (INT タイプまたは REAL タイプ);

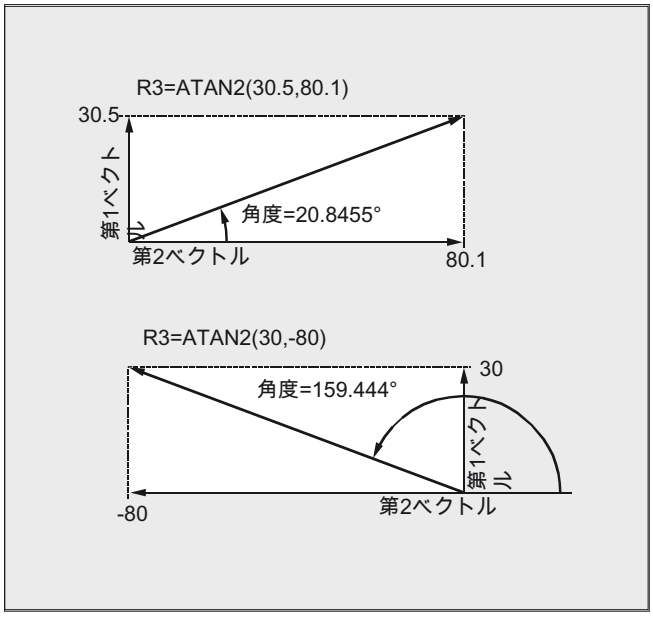
例:7 MOD 4.1 = 2.9

アークタンジェント 2:ATAN2

この関数は、2 つの相互に直交するベクトルの合計ベクトルの角度を計算します。

結果は、4 つの象限(--180° < 0 < +180°)のいずれかに該当します。

基準角度は常に、2 番目のベクトル値の正方向を基準にした角度です。



プログラミング例

プログラムコード	コメント
R1=R1+1	; 新しい R1 = 古い R1 + 1
R1=R2+R3 R4=R5-R6 R7=R8*R9	
R10=R11/R12 R13=SIN(25.3)	
R14=R1*R2+R3	; 乗算または除算は、加算または減算より先に実行します。
R14=(R1+R2)*R3	; 括弧内の式を最初に計算します。
R15=SQRT(POT(R1)+POT(R2))	; 内側の括弧内を最初に計算します。 R15 = ((R1^2 + R2^2))の平方根
RESFRAME= FRAME1:FRAME2	; 結合演算子による FRAME 論理演算
FRAME3=CTRANS(…):CROT(…)	FRAME 成分での値の割り当て

2.4 比較演算と論理演算

比較演算子を使用すると、ジャンプの条件などを定式化できます。複雑な式も比較できます。

比較演算子は、CHAR、INT、REAL、および BOOL タイプの変数に適用されます。コード値は CHAR タイプで比較されます。
タイプが STRING、AXIS、および FRAME の場合、STRING タイプの演算子に使用できる **==**と**<>**を、シンクロナイズドアクションでも使用できます。

比較演算の結果は常に BOOL タイプです。

真の値をリンクするには、**論理演算子**を使用します。

論理演算子は、BOOL タイプの変数にのみ使用できます。ただし、内部タイプ変換によって CHAR、INT、および REAL データタイプにも使用できます。

論理(ブール)演算の場合は、BOOL、CHAR、INT、および REAL データタイプには以下が適用されます。

- 0 の場合:FALSE
- 0 以外の場合:TRUE

ビット単位の論理演算子

論理演算子は、CHAR と INT タイプの単独ビットにも適用できます。タイプは自動的に変換されます。

プログラミング

関係演算子	意味
==	等しい
<>	等しくない
>	より大きい
<	より小さい
>=	等しいかそれより大きい
<=	等しいかそれより小さい

論理演算子	意味
および	論理積
OR	論理和

NOT	否定
XOR	排他的論理和

ビット単位の論理演算子	意味
B_AND	ビット毎の論理積
B_OR	ビット毎の論理和
B_NOT	ビット毎の否定
B_XOR	ビット毎の排他的論理和

注記

算術式では、通常の優先規則にかかわらず、括弧を使用して、すべての演算子の実行順序を指定できます。

注記

BOOLEAN オペランドと演算子の間はスペースを空けてください。

注記

演算子 B_NOT は 1 つのオペランドのみ参照します。このオペランドは、演算子の後に置かれます。

例

例 1:比較演算子

```
IF R10>=100 GOTOF DEST
```

または

```
R11=R10>=100
IF R11 GOTOF DEST
```

R10>=100 比較演算の結果は、まず、R11 に保持されます。

例 2:論理演算子

```
IF (R10<50) AND ($AA_IM[X]>=17.5) GOTOF DESTINATION
```

または

```
IF NOT R10 GOTOB START
```

NOT は 1 つのオペランドのみ参照します。

例 3:ビット単位の論理演算子

```
IF $MC_RESET_MODE_MASK B_AND 'B10000' GOTOF  ACT_PLANE
```

2.5 比較演算誤差の精度補正(TRUNC)

TRUNC 命令は、精度係数で乗算されたオペランドの切り捨てを実行します。

設定可能な比較演算命令の精度

REAL タイプのプログラムデータは、内部では IEEE 形式の 64 ビットで表わされます。この表示形式により、10 進数が不正確に表現され、最適な計算値と比較したときに予期しない結果となる場合があります。

相対等式値

この表示形式が原因による不正確さによってプログラムの流れが中断されないように、比較演算命令では、絶対等式値ではなく相対等式値でチェックします。

構文

比較演算誤差の精度補正

```
TRUNC (R1*1000)
```

意味

TRUNC:	小数点以下を切り捨て

10^{-12} の精度の相対等式値が考慮される演算子

- 等しい: (==)
- 等しくない: (<>)
- 以上: (>=)
- 以下: (<=)
- より大きい/より小さい: (><) (絶対等式値を使用して比較します)
- より大きい: (>)
- より小さい: (<)

互換性

互換性の理由により、(>)と(<)の相対等式のチェックは、マシンデータを MD10280 $MN_PROG_FUNCTION_MASK Bit0 = 1 に設定して無効にできます。

注記

REAL タイプのデータとの比較演算は、上記の理由により、不正確な結果となることがあります。誤差が大きすぎる場合は、オペランドを精度係数で乗算して、TRUNC で切り捨てをおこなう INTEGER 計算を使用してください。

シンクロナイズドアクション

比較演算命令の動作説明は、シンクロナイズドアクションにも該当します。

例

例 1:精度の配慮

プログラムコード	コメント
N40 R1=61.01 R2=61.02 R3=0.01	; 初期値の割り当て
N41 IF ABS(R2-R1) > R3 GOTOF ERROR	; 今まではジャンプが実行されていたはずです。
N42 M30	; プログラム終了
N43 ERROR:SETAL(66000)	
R1=61.01 R2=61.02 R3=0.01	; 初期値の割り当て
R11=TRUNC(R1*1000) R12=TRUNC(R2*1000)	; 精度補正
R13=TRUNC(R3*1000)	
IF ABS(R12-R11) > R13 GOTOF ERROR	; ジャンプは実行されません
M30	; プログラム終了
ERROR:SETAL(66000)	

例 2 :両方のオペランドの商の計算と評価

プログラムコード	コメント
R1=61.01 R2=61.02 R3=0.01	; 初期値の割り当て
IF ABS((R2-R1)/R3)-1) > 10EX-5 GOTOF ERROR	; ジャンプは実行されません
M30	; プログラム終了
ERROR:SETAL(66000)	

2.6 変数の最小値、最大値、および範囲(MINVAL、MAXVAL、および BOUND)

MINVAL および MAXVAL 命令は、2 つの変数の値を比較します。結果として、小さい方の値(MINVAL の場合)または大きい方の値(MAXVAL の場合)がそれぞれ得られます。

BOUND コマンドは、テスト変数の値が、定義した値の範囲内に入るかどうかをテストします。

構文

```
<小さい方の値>=MINVAL(<変数 1>,<変数 2>)
<大きい方の値>=MAXVAL(<変数 1>,<変数 2>)
<戻り値>=<BOUND>(<最小値>,<最大値>,<テスト変数>)
```

意味

MINVAL:	2 個の変数(<変数 1>、<変数 2>)のうち、**小さい方**の値が得られます。
<小さい方の値>:	MINVAL 命令の結果変数 小さい方の変数値に設定されます。
MAXVAL:	2 個の変数(<変数 1>、<変数 2>)のうち、**大きい方**の値が得られます。
<大きい方の値>:	MAXVAL 命令の結果変数 大きい方の変数値に設定されます。
BOUND:	変数(<テスト変数>)が、定義した値の範囲内にあるかどうかをテストします。
<最小値>:	値の範囲の最小値を定義する変数。
<最大値>:	値の範囲の最大値を定義する変数。
<戻り値>:	BOUND 命令の結果変数 テスト変数の値が定義した値の範囲内にある場合は、結果変数がテスト変数の値に設定されます。 テスト変数の値が最大値より大きい場合は、結果変数が定義範囲の最大値に設定されます。 テスト変数の値が最小値より小さい場合は、結果変数が定義範囲の最小値に設定されます。

注記

MINVAL、MAXVAL、および BOUND は、シンクロナイズドアクションでもプログラム指令できます。

注記

両方の値が等しい場合の動作

両方の値が等しい場合は、MINVAL/MAXVAL では、この等しい値に設定されます。BOUND の場合は、テストされる変数の値がそのまま戻されます。

例

プログラムコード	コメント
DEF REAL rVar1=10.5, rVar2=33.7, rVar3, rVar4, rVar5, rValMin, rValMax, rRetVar	
rValMin=MINVAL(rVar1,rVar2)	; rValMin を値 10.5 に設定します。
rValMax=MAXVAL(rVar1,rVar2)	; rValMax を値 33.7 に設定します。
rVar3=19.7	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3 は範囲内です。rRetVar を 19.7 に設定します。
rVar3=1.8	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3 は最小値を下回っています。rRetVar を 10.5 に設定します。
rVar3=45.2	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3 は最大値を上回っています。rRetVar を 33.7 に設定します。

2.7 演算子の優先度

各演算子には優先度が割り当てられます。式を解析するときには、常に最も優先度が高い演算子が最初に使用されます。複数の演算子の優先度が同じである場合は、左から右へと解析されます。

算術式では、通常の優先規則にかかわらず、括弧を使用して、すべての演算子の実行順序を指定できます。

演算の順序

優先度の高い方から低い方へ

1.	NOT, B_NOT	否定、ビット毎の否定
2.	*, /, DIV, MOD	乗算、除算
3.	+, –	加算、減算
4.	B_AND	ビット毎の論理積
5.	B_XOR	ビット毎の排他的論理和
6.	B_OR	ビット毎の論理和
7.	および	論理積
8.	XOR	排他的論理和
9.	OR	論理和
10.	<<	文字列の結合、結果は STRING タイプ
11.	==, <>, >, <, >=, <=	比較演算子

注記

フレーム用の結合演算子「:」は、他の演算子と同じような式で使用できません。したがって、この演算子には優先度はありません。

例:IF 命令

```
If (otto==10) and (anna==20) gotof end
```

2.8 実行可能なタイプ変換

機能

割り当て時のタイプ変換

変数に割り当てた定数値、変数、または式は、変数タイプを互換にしてください。この場合は、値を割り当てると、自動的にタイプが変換されます。

実行可能なタイプ変換

変換後	REAL	INT	BOOL	CHAR	STRING	AXIS	FRAME
変換前							
REAL	可	可*	可 1)	可*	–	–	–
INT	可	可	可 1)	可 2)	–	–	–
BOOL	可	可	可	可	可	–	–
CHAR	可	可	可 1)	可	可	–	–
STRING	–	–	可 4)	可 3)	可	–	–
AXIS	–	–	–	–	–	可	–
FRAME	–	–	–	–	–	–	可

意味

* REAL から INT へのタイプ変換では、0.5 以上の端数値は切り上げられ、それ以外の値は切り捨てられます(ROUND 機能を参照ください)。

1) 値が 0 以外の場合は TRUE、0 の場合は FALSE です。

2) 値が許容範囲内である場合です。

3) 1 文字のみの場合です。

4) 文字列長が 0 以下の場合は FALSE、それ以外の場合は TRUE です。

注記

変換により、値が目標範囲を超える場合は、エラーメッセージが出力されます。

式に複数のタイプが混合している場合は、タイプが自動変換されます。タイプ変換はシンクロナイズドアクションでもできます。「シンクロナイズドアクション、自動的なタイプ変換」の章を参照してください。

2.9 文字列演算子

文字列演算子

従来の演算子の他に、次の「割り当て」と「比較演算」の文字列演算子が実行できます。

- STRING へのタイプ変換(AXSTRING) (ページ 97)
- STRING からのタイプ変換(NUMBER、ISNUMBER、AXNAME) (ページ 98)
- 文字列の結合(<<) (ページ 100)
- 小文字/大文字への変換(TOLOWER、TOUPPER) (ページ 101)
- 文字列の長さの特定(STRLEN) (ページ 102)
- 文字列中の文字/文字列の検索(INDEX、RINDEX、MINDEX、MATCH) (ページ 103)
- 抽出文字列の選択(SUBSTR) (ページ 104)
- 個々の文字の読み取りと書き込み (ページ 105)
- 文字列のフォーマット(SPRINT) (ページ 106)

0 文字の特別な意味

内部では、0 文字は、文字列の終了識別子として解釈されます。文字を 0 文字に置き換えると、それ以降の文字列は切り捨てられます。

例:

プログラムコード	コメント
`DEF STRING[20] STRG="axis . stationary"`	
`STRG[6]="X"`	
`MSG(STRG)`	`; メッセージ「axis X stationary」を返します。`
`STRG[6]=0`	
`MSG(STRG)`	`; メッセージ「axis」を返します。`

2.9.1 STRING へのタイプ変換(AXSTRING)

「STRING へのタイプ変換」機能により、さまざまなタイプの変数を、メッセージの成分として使用できます(MSG)。

<<演算子を使用すると、データタイプ INT、REAL、CHAR、および BOOL については、この機能が自動的に実行されます(「 文字列の結合(**<<**) (ページ 100)」を参照してください)。

INT 値は、通常の読み取り可能な書式に変換されます。REAL 値は小数点以下 10 桁までの値に変換されます。

AXIS タイプの変数は、AXSTRING 命令を使用して STRING に変換できます。

構文

```
<STRING_RES> = << <任意のタイプ>
<STRING_RES> = AXSTRING(<軸識別子>)
```

意味

<STRING_RES>:	タイプ変換の結果の変数	
	タイプ:	STRING
<任意のタイプ>:	変数のタイプ INT、REAL、CHAR、STRING、および BOOL	
AXSTRING:	AXSTRING 命令は、指定した軸識別子を文字列として返します。	
<軸識別子>:	軸識別子の変数	
	タイプ:	AXIS

注記

FRAME 変数は変換できません。

2.9.2 STRING からのタイプ変換(NUMBER、ISNUMBER、AXNAME)

NUMBER 命令を使用して、STRING から REAL に変換します。変換できるかどうかは、ISNUMBER 命令を使用して確認できます。

文字列は、AXNAME 命令を使用して軸データタイプに変換します。

構文

```
<REAL_RES>=NUMBER("<文字列>")
<BOOL_RES>=ISNUMBER("<文字列>")
<AXIS_RES>=AXNAME("<文字列>")
```

意味

NUMBER:	NUMBER 命令は、<文字列>で示された数値を REAL 値として返します。		
<文字列>:	変換する STRING タイプの変数		
<REAL_RES>:	NUMBER によるタイプ変換の結果の変数		
	タイプ:	REAL	
ISNUMBER:	ISNUMBER 命令は、<文字列>を有効な数値に変換できるかどうかを確認します。		
<BOOL_RES>:	ISNUMBER による問い合わせ結果の変数		
	タイプ:	BOOL	
	値:	TRUE	ISNUMBER は、<文字列>が、言語規則に準じた有効な REAL 数値を表わす場合は、値 TRUE を返します。
		FALSE	NUMBER を同じ<文字列>で呼び出したときに、ISNUMBER が値 FALSE を返す場合は、アラームが発生します。
AXNAME:	AXNAME 命令は、指定した<文字列>を軸識別子へ変換します。 **注:** 設定した軸識別子に<文字列>が割り当てられない場合は、アラームが発生します。		
<AXIS_RES>:	AXNAME によるタイプ変換の結果の変数		
	タイプ:	AXIS	

例

```
DEF BOOL BOOL_RES
DEF REAL REAL_RES
DEF AXIS AXIS_RES
REAL_RES == 1234.9876Ex-7                ; BOOL_RES == TRUE
BOOL_RES=ISNUMBER("1234XYZ")             ; BOOL_RES == FALSE
REAL_RES=NUMBER("1234.9876Ex-7")         ; REAL_RES == 1234.9876Ex-7
AXIS_RES=AXNAME("X")                     ; AXIS_RES == X
```

2.9.3 文字列の結合(<<)

「文字列の結合」機能を使用すると、個々の成分で文字列を設定できます。

結合を実行するには、演算子「**<<**」を使用します。この演算子では、基本タイプである CHAR、BOOL、INT、REAL、および STRING のすべての組み合わせの目標タイプが STRING です。必要な変換はいずれも、既存の規則に従って実行されます。

構文

<任意のタイプ> << <任意のタイプ>

意味

<任意のタイプ>:	CHAR、BOOL、INT、REAL、または STRING タイプの変数
<< :	変数(**<**任意のタイプ**>**)を結合して、文字列(STRING タイプ)を作成する演算子。 この演算子はいわゆる「単一」タイプとして、単独でも使用できます。これは、STRING への明示的なタイプ変換(FRAME と AXIS は変換できません)に使用できます。 << <任意のタイプ>

たとえば、次のように、メッセージまたは命令をテキストリストから設定後に、パラメータを挿入できます(ブロック名称など)。

```
MSG(STRG_TAB[LOAD_IDX]<<BLOCK_NAME)
```

注記

文字列結合の中間の結果が、最大文字列長を超えないようにしてください。

注記

FRAME と AXIS タイプは、演算子「**<<**」と組み合わせて使用することはできません。

例

例 1:文字列の結合

プログラムコード	コメント
`DEF INT IDX=2`	
`DEF REAL VALUE=9.654`	

プログラムコード	コメント
DEF STRING[20] STRG="INDEX:2"	
IF STRG=="Index:"<<IDX GOTOF NO_MSG	
MSG("Index:"<<IDX<<"/value:"<<VALUE)	; 表示: "Index:2/value:9.654"
NO_MSG:	

例 2 :<<による明示的なタイプ変換

プログラムコード	コメント
DEF REAL VALUE=3.5	
<<VALUE	; 指定した REAL タイプの変数を STRING タイプに変換します。

2.9.4 小文字/大文字への変換(TOLOWER、TOUPPER)

「小文字/大文字の英字への変換」機能を使用すると、文字列のすべての英字を標準表記に変換できます。

構文

```
<STRING_RES>=TOUPPER("<文字列>")
<STRING_RES>=TOLOWER("<文字列>")
```

意味

TOUPPER:	TOUPPER 命令を使用すると、文字列のすべての英字が**大文字**の英字に変換されます。	
TOLOWER:	TOLOWER 命令を使用すると、文字列のすべての英字が**小文字**の英字に変換されます。	
<文字列>:	変換する文字列	
	タイプ:	STRING
<STRING_RES>:	変換結果の変数	
	タイプ:	STRING

例

ユーザーは操作画面で入力を開始できるため、標準の文字表記(大文字または小文字)を使用できます。

プログラムコード

```
DEF STRING [29] STRG
...
IF "LEARN.CNC"==TOUPPER(STRG) GOTOF LOAD_LEARN
```

2.9.5 文字列の長さの特定(STRLEN)

STRLEN 命令は、文字列の長さを特定します。

構文

```
<INT_RES>=STRLEN("<STRING>")
```

意味

STRLEN:	STRLEN 命令は、指定した文字列の長さを特定します。 文字列の先頭から数えた、0 文字でない文字数が返されます。	
<文字列>:	文字列の長さを特定します。	
	タイプ:	STRING
<INT_RES>:	特定結果の変数	
	タイプ:	INT

例

この機能を、単一文字へのアクセスと組み合わせて使用すると、文字列の終了を特定できます。

プログラムコード

```
IF (STRLEN(BLOCK_NAME)>10) GOTOF ERROR
```

2.9.6 文字列中の文字/文字列の検索(INDEX、RINDEX、MINDEX、MATCH)

この機能は、単一文字または文字列を文字列で検索します。この機能の結果は、検索した文字列中の文字/文字列の位置を指定します。

構文

INT_RES=INDEX(STRING,CHAR) ; **結果のタイプ:INT**

INT_RES=RINDEX(STRING,CHAR) ; **結果のタイプ:INT**

INT_RES=MINDEX(STRING,STRING) ; **結果のタイプ:INT**

INT_RES=MINDEX(STRING,STRING) ; **結果のタイプ:INT**

意味

検索機能:検索が正常に終了した場合、文字列の位置(1 番目のパラメータ)を返します。文字/文字列が見つからない場合は、値-1 が返されます。最初の文字の位置は 0 です。

意味

INDEX:	2 番目のパラメータとして指定した文字を、(先頭から)1 番目のパラメータで検索します。
RINDEX:	2 番目のパラメータとして指定した文字を、(末尾から)1 番目のパラメータで検索します。
MINDEX:	INDEX 機能に対応します。ただし、最初に見つかった文字のインデックスを返す文字のリストが(文字列として)渡された場合を除きます。
MATCH:	文字列中で文字列を検索します。

これを使用すると、ブランクまたはパス区切り文字(「/」)の位置など、特定の条件に従って文字列が分割されます。

例

パス名称とブロック名称へ入力文字列を分割

プログラムコード	コメント
DEF INT PFADIDX, PROGIDX	
DEF STRING[26] INPUT	
DEF INT LISTIDX	
INPUT = "/_N_MPF_DIR/_N_EXECUTE_MPF"	

プログラムコード	コメント
LISTIDX = MINDEX (INPUT, "M,N,O,P") + 1	; LISTIDX で返される値は 3 です。これは、選択リストのパラメータ INPUT の「N」が先頭から開始して最初の文字のためです。
PFADIDX = INDEX (INPUT, "/") +1	; この結果、以下が適用されます。PFADIDX = 1
PROGIDX = RINDEX (INPUT, "/") +1	; この結果、以下が適用されます。PROGIDX = 12
	; 次の章で説明する SUBSTR 機能を使用すると、変数 INPUT を「パス」と「モジュール」成分に分割できます。
VARIABLE = SUBSTR (INPUT, PFADIDX, PROGIDX-PFADIDX-1)	; この時に、"_N_MPF_DIR"を返します。
VARIABLE = SUBSTR (INPUT, PROGIDX)	; この時に、"_N_EXECUTE_MPF"を返します。

2.9.7 抽出文字列の選択(SUBSTR)

SUBSTR 命令で文字列内の任意の部分を読み取ることができます。

構文

<STRING_RES>=SUBSTR(<文字列>,<インデックス>,<長さ>)

<STRING_RES>=SUBSTR(<文字列>,<インデックス>)

意味

SUBSTR:	この命令は、**<**文字列**>**の**<**インデックス**>**位置から指定した**<**長さ**>**の抽出文字列を返します。 パラメータ**<**長さ**>**を指定しない場合は、**<**インデックス**>**位置から文字列の末尾までの抽出文字列が返されます。
<インデックス>:	文字列内の抽出文字列の開始位置開始位置が文字列の末尾の後の場合は、空の文字列(" ")が返されます。文字列の最初の文字:インデックス = 0 値の範囲:0 ... (文字列長 - 1)
<長さ>:	抽出文字列の長さ。指定した長さが長すぎる場合は、文字列の末尾までの抽出文字列が返されます。 値の範囲:1 ... (文字列長 - 1)

例

プログラムコード	コメント
DEF STRING[29] RES	
; 1	
; 01234567890123456789	
RES = SUBSTR("QUITTUNG:10 to 99", 10, 2)	; RES == "10"
RES = SUBSTR("QUITTUNG:10 to 99", 10)	; RES == "10 to 99"

2.9.8 個々の文字の読み取りと書き込み

文字列内の個々の文字を読み取りと書き込みできます。

次の必要条件を遵守してください。

- ユーザー定義変数でのみ可能で、システム変数ではおこなえません。
- 文字列の個々の文字は、サブプログラム呼び出しで「値渡し」のみを転送します。

構文

<文字>=<文字列>[<インデックス>]
<文字>=<文字列配列>[<配列インデックス>,<インデックス>]
<文字列>[<インデックス>]=<文字>
<文字列配列>[<配列インデックス>,<インデックス>]=<文字>

意味

<文字列>:	任意の文字列
<文字>:	CHAR タイプの変数
<インデックス>:	文字列内の文字の位置 文字列の最初の文字:インデックス = 0 値の範囲:0 ... (文字列長 - 1)

例

例 1: 変数メッセージ

プログラムコード	コメント
; 0123456789	
DEF STRING [50] MESSAGE = "Axis n has reached position"	
MESSAGE [6] = "X"	

プログラムコード	コメント
MSG (MESSAGE)	; "軸 X は位置に到達しました"

例 2: システム変数の評価

プログラムコード	コメント
DEF STRING [50] STRG	;システム変数用バッファ
...	
STRG = $P_MMCA	; システム変数のロード
IF STRG[0] == "E" GOTO ...	; システム変数の評価

例 3: パラメータ転送「値渡し」と「参照渡し」

プログラムコード	コメント
; 0123456	
DEF STRING[50] STRG = "Axis X"	
DEF CHAR CHR	
...	
EXTERN UP_VAL(ACHSE)	;「値渡し」パラメータを含むサブプログラムの定義
EXTERN UP_REF(VAR ACHSE)	;「参照渡し」パラメータを含むサブプログラムの定義
...	
UP_VAL(STRG[6])	; パラメータ転送「値渡し」
...	
CHR = STRG[6]	; バッファ
UP_REF(CHR)	; パラメータ転送「参照渡し」

2.9.9 文字列のフォーマット(SPRINT)

定義済みの SPRINT 機能を使用して、文字列をフォーマットし、たとえば外部機器に出力するための準備を整えることができます(「Process DataShare - 外部機器/ファイルへの出力(EXTOPEN、WRITE、EXTCLOSE): (ページ 793)」も参照してください)。

構文

"<結果文字列>"=SPRINT("<フォーマット文字列>",<値_1>,<値_2>,..., <値_n>)

意味

SPRINT:	STRING タイプの値を提供する予約機能のための識別子。
"<フォーマット文字列>":	固定要素と可変要素を含む文字列。可変要素は、フォーマット制御文字%とそれに続くフォーマット記述を使用して定義します。
<値_1>,<値_2>,…,<値_n>:	定数形式または NC 変数形式の値。<フォーマット文字列>のフォーマット記述に対応して、n 番目のフォーマット制御文字%の位置する場所に挿入されます。
"<結果文字列>":	フォーマットされた文字列(最大 400 バイト)

使用可能なフォーマット記述

%B:	変換する値が以下の場合に、「TRUE」文字列に変換します。
	● 0 と等しくない。
	● 空の文字列でない(文字列値の場合)。
	変換する値が以下の場合に、「FALSE」文字列に変換します。
	● 0 と等しい。
	● 空の文字列。
	例:
	`N10 DEF BOOL BOOL_VAR=1`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF BOOL_VAR:%B", BOOL_VAR)`
	結果文字列「CONTENT OF BOOL_VAR:TRUE」が文字列変数 RESULT に書き込まれます。

%C:	ASCII 文字に変換します。
	例:
	`N10 DEF CHAR CHAR_VAR="X"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF CHAR_VAR:%C",CHAR_VAR)`
	結果文字列「CONTENT OF CHAR_VAR:X」が文字列変数 RESULT に書き込まれます。

%D:	整数値の文字列(INTEGER)に変換します。 **例:** N10 DEF INT INT_VAR=123 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF INT_VAR:%D",INT_VAR) 結果文字列「CONTENT OF INT_VAR:123」が文字列変数 RESULT に書き込まれます。
%<m>D:	整数値の文字列(INTEGER)に変換します。文字列は最低、<m>文字長となります。文字が欠落している場所には、スペースが左揃えで埋められます。 **例:** N10 DEF INT INT_VAR=-123 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF INT_VAR:%6D",INT_VAR) 結果文字列「CONTENT OF INT_VAR:xx-123」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。
%F:	小数点以下 6 桁の 10 進数の文字列に変換します。必要に応じて、小数位が四捨五入されるか 0 で埋められます。 **例:** N10 DEF REAL REAL_VAR=-1.2341234EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%F",REAL_VAR) 結果文字列変数 RESULT に文字列「CONTENT OF REAL_VAR:-1234.123400」が書き込まれます。
%<m>F:	小数点以下 6 桁で全体の長さが最低<m>文字の 10 進数の文字列に変換します。必要に応じて、小数位が四捨五入されるか 0 で埋められます。文字が欠落している場所には、全体の長さ<m>まで左詰でスペースが埋められます。 **例:** N10 DEF REAL REAL_VAR=-1.23412345678EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15F",REAL_VAR) 結果文字列変数 RESULT に文字列「CONTENT OF REAL_VAR:xxx-1234.123457」が書き込まれます(「x」はスペース用のプレースホルダーです)。

%.<n>F:	小数点以下<n>桁の 10 進数の文字列に変換します。必要に応じて、小数位が四捨五入されるか 0 で埋められます。 **例:** N10 DEF REAL REAL_VAR=-1.2345678EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3F",REAL_VAR) 結果文字列「CONTENT OF REAL_VAR:-1234.568」が文字列変数 RESULT に書き込まれます。
%<m>.<n>F:	小数点以下<n>桁で全体の長さが最低<m>文字の 10 進数の文字列に変換します。必要に応じて、小数位が四捨五入されるか 0 で埋められます。文字が欠落している場所には、全体の長さ<m>まで左詰でスペースが埋められます。 **例:** N10 DEF REAL REAL_VAR=-1.2341234567890EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%10.2F",REAL_VAR) 結果文字列「CONTENT OF REAL_VAR:xx-1234.12」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。
%E:	指数表記で表した 10 進数の文字列に変換します。仮数は保存され、小数点前の 1 桁と小数点以下 6 桁で正規化されます。必要に応じて、小数位が四捨五入されるか 0 で埋められます。指数はキーワード「EX」で始まります。その後に符号(「+」または「-」)と 2 桁または 3 桁の数字が続きます。 **例:** N10 DEF REAL REAL_VAR=-1234.567890 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%E",REAL_VAR) 結果文字列「CONTENT OF REAL_VAR:-1.234568EX+03」が文字列変数 RESULT に書き込まれます。

%<m>E:	指数表記で表した、全体の長さが最低<m>文字の 10 進数の文字列に変換します。文字が欠落している場所には、スペースが左揃えで埋められます。仮数は保存され、小数点前の 1 桁と小数点以下 6 桁で正規化されます。必要に応じて、小数位が四捨五入されるか 0 で埋められます。指数はキーワード「EX」で始まります。その後に符号(「+」または「-」)と 2 桁または 3 桁の数字が続きます。
	例:
	`N10 DEF REAL REAL_VAR=-1234.5`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%20E",REAL_VAR)`
	結果文字列「CONTENT OF REAL_VAR:xxxxxx-1.234500EX+03」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。

%.<n>E:	指数表記で表した 10 進数の文字列に変換します。仮数は保存され、小数点前の 1 桁と小数点以下<n>桁で正規化されます。必要に応じて、小数位が四捨五入されるか 0 で埋められます。指数はキーワード「EX」で始まります。その後に符号(「+」または「-」)と 2 桁または 3 桁の数字が続きます。
	例:
	`N10 DEF REAL REAL_VAR=-1234.5678`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.2E",REAL_VAR)`
	結果文字列「CONTENT OF REAL_VAR:-1.23EX+03」が文字列変数 RESULT に書き込まれます。

%<m>.<n>E:	指数表記で表した、全体の長さが最低<m>文字の 10 進数の文字列に変換します。文字が欠落している場所には、スペースが左揃えで埋められます。仮数は保存され、小数点前の 1 桁と小数点以下<n>桁で正規化されます。必要に応じて、小数位が四捨五入されるか 0 で埋められます。指数はキーワード「EX」で始まります。その後に符号(「+」または「-」)と 2 桁または 3 桁の数字が続きます。
	例:
	`N10 DEF REAL REAL_VAR=-1234.5678`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.2E", REAL_VAR)`
	結果文字列「CONTENT OF REAL_VAR:xx-1.23EX+03」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。

%G:	10 進数表記または指数表記で表した 10 進数 - 数値の範囲によって異なる - の文字列に変換します。表記する絶対値が 1.0EX-04 未満であるか 1.0EX+06 より大きい/等しい場合は、指数表記が選択されます。それ以外の場合は、10 進数表記となります。最大で上位 6 桁が表示されるか、必要に応じて四捨五入されます。
	10 進数表記の場合の例:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)`
	結果文字列「CONTENT OF REAL_VAR:0.000123457」が文字列変数 RESULT に書き込まれます。
	指数表記の場合の例:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+06`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)`
	結果文字列「CONTENT OF REAL_VAR:1.23457EX+06」が文字列変数 RESULT に書き込まれます。

%<m>G:	10 進数表記または指数表記で表した 10 進数 - 数値の範囲によって異なる - の文字列への変換(%G と同様)。文字列の全体の長さは最低、<m>文字となります。文字が欠落している場所には、スペースが左揃えで埋められます。
	10 進数表記の場合の例:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)`
	結果文字列「CONTENT OF REAL_VAR:xxxx0.000123457」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。
	指数表記の場合の例:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+06`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)`
	結果文字列「CONTENT OF REAL_VAR:xxx1.23457EX+06」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。

%.<n>G:	10 進数表記または指数表記で表した 10 進数 - 数値の範囲によって異なる - の文字列に変換します。最大で上位**<n>**桁が表示されるか、必要に応じて四捨五入されます。表記する絶対値が 1.0EX-04 未満であるか 1.0EX(+<n>)より大きい/等しい場合は、指数表記が選択されます。それ以外の場合は、10 進数表記となります。
	10 進数表記の場合の例:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)`
	結果文字列「CONTENT OF REAL_VAR:0.000123」が文字列変数 RESULT に書き込まれます。
	指数表記の場合の例:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT = SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)`
	結果文字列「CONTENT OF REAL_VAR:1.23EX+03」が文字列変数 RESULT に書き込まれます。

%<m>.<n>G:	10 進数表記または指数表記で表した 10 進数 - 数値の範囲によって異なる - の文字列への変換(%.<n>G と同様)。文字列の全体の長さは最低、**<m>**文字となります。文字が欠落している場所には、スペースが左揃えで埋められます。
	10 進数表記の場合の例:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)`
	結果文字列「CONTENT OF REAL_VAR:xxx0.0001235」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。
	指数表記の場合の例:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)`
	結果文字列「CONTENT OF REAL_VAR:xx1.235EX+06」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。

%.<n>P:	小数点以下<n>桁を考慮して、REAL 値を INTEGER 値に変換。
	INTEGER 値は 32 ビットのバイナリ値として出力されます。変換する値が 32 ビットで表せない場合、処理はアラームで中断されます。
	フォーマット命令%.<n>P を使用して生成されるバイトシーケンスにもバイナリゼロを入れることができるため、この方法で生成された文字列全体が NC データタイプ STRING の規約に対応しなくなります。その結果、STRING タイプの変数に保存することも、NC 言語の文字列命令を使用してさらに処理することもできません。唯一可能なのは、パラメータを該当する外部機器の出力で WRITE 命令に転送するのに使用することです(以下の例を参照してください)。
	<フォーマット文字列>にフォーマット記述、タイプ%P が入ると同時に、%.<n>P で生成された 2 進数を除く文字列全体が MD10750 $MN_SPRINT_FORMAT_P_CODE に対応して、ASCII 文字コード、ISO (DIN6024)、または EIA (RS244)で出力されます。変換できない文字がプログラム指令されている場合、処理はアラームで中断されます。
	例:
	`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`
	`... ; エラーの処理`
	`N200 WRITE(ERROR,EXT_DEVICE,SPRINT("INTEGER BINARY CODED:%.3P",REAL_VAR)`
	`N210 IF ERROR <> 0`
	`… ; エラーの処理`
	結果文字列「INTEGER BINARY CODED:'H0001E23A'」が出力機器/ext/dev/1 に転送されます。16 進数値 0x0001E23A は、10 進数値 123450 に対応しています。

%<m>.<n>P:	マシンデータ MD10751 $MN_SPRINT_FORMAT_P_DECIMAL の設定に対応する REAL 値を以下の文字列に変換します。
	● **<m> + <n>**桁の整数、**または**
	● 小数点前最大**<m>**桁および正確な小数点以下**<n>**桁の 10 進数
	フォーマット記述%.<n>P と同様に、文字列全体が MD10750 $MN_SPRINT_FORMAT_P_CODE で定義された文字コードで保存されます。
	MD10751 = 0 の場合の変換:
	REAL 値が**<m> + <n>**桁の整数の文字列に変換されます。必要に応じて、小数位が**<n>**桁に四捨五入されるか 0 で埋められます。小数点より前の桁の欠落している場所は、スペースで埋められます。-符号が左詰めで付けられます。+符号の代わりにスペースが入力されます。
	例:
	`N10 DEF REAL REAL_VAR=-123.45`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("PUNCHED TAPE FORMAT:%5.3P",REAL_VAR)`
	結果文字列「PUNCHED TAPE FORMAT:-xx123450」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。
	MD10751 = 1 の場合の変換:
	REAL 値が、小数点前最大**<m>**桁および正確な小数点以下**<n>**桁の 10 進数の文字列に変換されます。必要に応じて、小数点の前の桁が切り捨てられ、小数位が四捨五入されるか 0 で埋められます。**<n>**が 0 の場合、小数点も省略されます。
	例:
	`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)`
	結果文字列「PUNCHED TAPE FORMAT:-123.450 VAR2:23」が文字列変数 RESULT に書き込まれます。

%S:	文字列を挿入。
	例:
	`N10 DEF STRING[16] STRING_VAR="ABCDEFG"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%S",STRING_VAR)`
	結果文字列「CONTENT OF STRING_VAR:ABCDEFG」が文字列変数 RESULT に書き込まれます。

%<m>S:	最低**<m>**文字長の文字列を挿入。欠落している桁はスペースで埋められます。 **例:** N10 DEF STRING[16] STRING_VAR="ABCDEFG" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10S",STRING_VAR) 結果文字列「CONTENT OF STRING_VAR:xxxABCDEFG」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。
%.<n>S:	文字列の**<n>**個の文字(最初の文字から始まる)を挿入。 **例:** N10 DEF STRING[16] STRING_VAR="ABCDEFG" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%.3S",STRING_VAR) 結果文字列「CONTENT OF STRING_VAR:ABC」が文字列変数 RESULT に書き込まれます。
%<m>.<n>S:	文字列の**<n>**個の文字(最初の文字から始まる)を挿入。生成された文字列の全体長は最低、**<m>**文字となります。欠落している桁はスペースで埋められます。 **例:** N10 DEF STRING[16] STRING_VAR="ABCDEFG" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10.5S", STRING_VAR) 結果文字列「CONTENT OF STRING_VAR:xxxxxABCDE」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。
%X:	INTEGER 値を 16 進数表記の文字列に変換。 **例:** N10 DEF INT INT_VAR='HA5B8' N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("INTEGER HEXADECIMAL:%X",INT_VAR) 結果文字列「INTEGER HEXADECIMAL:A5B8」が文字列変数 RESULT に書き込まれます。

注記

識別子とキーワードの大文字と小文字を区別しないという NC 言語の特性が、フォーマット記述にも適用されます。その結果、大文字と小文字のどちらを使用してプログラム指令しても、機能的な違いはありません。

組み合わせのオプション

下の表に、どの NC データタイプをどのフォーマット記述と組み合わせられるかについての情報を記載します。自動的なデータタイプ変換に関する規則が適用されます(「データタイプ (ページ 66)」を参照してください)。

	NC データタイプ						
	BOOL	CHAR	INT	REAL	STRING	AXIS	FRAME
%B	+	+	+	+	+	-	-
%C	-	+	-	-	+	-	-
%D	+	+	+	+	-	-	-
%F	-	-	+	+	-	-	-
%E	-	-	+	+	-	-	-
%G	-	-	+	+	-	-	-
%S	-	+	-	-	+	-	-
%x	+	+	+	-	-	-	-
%P	-	-	+	+	-	-	-

注記

表では、NC データタイプ AXIS および FRAME を SPRINT 機能で直接使用できないことが示されています。ただし、以下は可能です。

- AXSTRING 機能を使用して AXIS データタイプを文字列に変換 – その後、SPRINT で処理することができます。
- フレーム成分アクセス毎に、FRAME データタイプの個々の値を読み取り。その結果、REAL データタイプを入手して SPRINT で処理することができます。

2.10 プログラムのジャンプと分岐

2.10.1 プログラムの先頭への復帰ジャンプ(GOTOS)

GOTOS 命令を使用すると、プログラムを繰り返すために、メインプログラムまたはサブプログラムの先頭にジャンプして戻ることができます。

マシンデータを使用して、プログラムの先頭への全ての復帰ジャンプに以下を設定できます。

- プログラムの実行時間を「0」に設定します。
- ワーク数のカウントが値「1」ずつ増加します。

構文

GOTOS

意味

GOTOS:	プログラムの先頭にジャンプするジャンプ命令。	
	実行は、次の NC/PLC インタフェース信号によって制御されます。 DB21 の DBX384.0(プログラム分岐の制御)	
	値:	意味
	0	プログラムの先頭へ復帰ジャンプしません。プログラムの実行は、GOTOS の後の次のパートプログラムブロックで再開します。
	1	プログラムの先頭へ復帰ジャンプします。パートプログラムは繰り返されます。

必要条件

- GOTOS は内部で STOPRE (先読み停止)を開始します。
- GOTOS で、データ定義(LUD 変数)を含むサブプログラムにジャンプした場合は、定義区間の後の最初のプログラムブロックにジャンプします。つまり、データ定義は再実行されません。定義した変数が、GOTOS ブロックでの設定値を保持し、定義区間でプログラム指令した標準値へリセットされないのはこのためです。
- GOTOS 命令は、シンクロナイズドアクションとテクノロジサイクルでは使用できません。

例

プログラムコード	コメント
N10 ...	; プログラムの開始
...	
N90 GOTOS	; プログラムの先頭へジャンプします
...	

2.10.2 ジャンプマークへのプログラムのジャンプ(GOTOB、GOTOF、GOTO、GOTOC)

ジャンプラベルをプログラムに設定し、同じプログラム内の別の場所からそこに、GOTOF、GOTOB、GOTO、または GOTOC 命令を使用してジャンプできます。プログラムの実行は、ジャンプラベルの直後にある命令で再開されます。これは、プログラム内で分岐を実行できることを意味します。

ジャンプラベルの他に、メインブロック番号とサブブロック番号をジャンプ先にすることができます。

ジャンプ命令の前にジャンプ条件(IF ...)の式を指定した場合は、その条件を満たす場合にのみ、プログラムジャンプが実行されます。

構文

```
GOTOB <ジャンプ先>
IF <ジャンプ条件> == TRUE GOTOB <ジャンプ先>

GOTOF <ジャンプ先>
IF <ジャンプ条件> == TRUE GOTOF <ジャンプ先>

GOTO <ジャンプ先>
IF <ジャンプ条件> == TRUE GOTO <ジャンプ先>

GOTOC <ジャンプ先>
IF <ジャンプ条件> == TRUE GOTOC <ジャンプ先>
```

意味

GOTOB:	プログラムの先頭方向へジャンプするジャンプ命令。
GOTOF:	プログラムの末尾方向へジャンプするジャンプ命令。
GOTO:	ジャンプ先を検索するジャンプ命令。検索は、最初にプログラムの末尾方向へ、次にプログラムの先頭方向へ実行されます。

GOTOC:	GOTO と同じ結果となりますが、この場合は、アラーム 14080「Jump destination not found」(ジャンプ先が見つかりません)がマスクされます。 つまり、ジャンプ先の検索が正常に終了しなかった場合でも、プログラムの実行が中断されず、GOTOC 命令の後のプログラム行が続行されます。	
<ジャンプ先>:	ジャンプ先パラメータ 指定できるデータには以下のものがあります。	
	<ジャンプラベル**>**:	ジャンプ先は、次のように、ユーザーが定義した名称でプログラムに設定されたジャンプラベルです。**<**ジャンプラベル**>:**
	<ブロック番号**>**:	ジャンプ先は、メインブロック番号またはサブブロック番号です(例:200, N300)。
	STRING タイプ変数:	ジャンプ先となる変数です。変数は、ジャンプラベルまたはブロック番号を表わします。
IF:	条件ジャンプの式に使用するキーワード。 条件ジャンプには、すべての比較演算子と論理演算子を使用できます(結果:TRUE または FALSE)。プログラムジャンプは、この演算の結果が TRUE である場合に実行されます。	

注記

ジャンプラベル

ジャンプラベルは常に、ブロックの先頭に配置します。プログラム番号がある場合は、ブロック番号の直後にジャンプラベルを配置します。

ジャンプラベルの名称を付けるときは、次の規則が適用されます。

- 文字数:
  - 最小 2
  - 最大 32
- 次の文字を使用できます。
  - 英字
  - 数字
  - アンダースコア
- 最初の 2 文字は、英字またはアンダースコアにしてください。
- ジャンプラベルの名称の後にはコロン(「:」)を付けます。

必要条件

- ジャンプ先となるのは、プログラムの**中**ある、ジャンプラベルまたはブロック番号が付いているブロックのみです。
- ジャンプ条件のないジャンプ命令は、個別のブロックにプログラム指令してください。この制限は、ジャンプ条件のあるジャンプ命令には適用されません。この場合は、複数のジャンプ命令を同じブロックの式に使用できます。
- ジャンプ条件のないジャンプ命令を含むプログラムの場合、プログラムの終了 M2/M30 は必ずしも、プログラムの末尾にある必要はありません。

例

例 1:ジャンプラベルへのジャンプ

プログラムコード	コメント
N10 …	
N20 GOTOF Label_1	; プログラムの末尾方向の、
	; ジャンプラベル「Label_1」にジャンプします。
N30 …	
N40 Label_0:R1=R2+R3	; ジャンプラベル「Label_0」が設定されています。
N50 …	
N60 Label_1:	; ジャンプラベル「Label_1」が設定されています。
N70 …	
N80 GOTOB Label_0	; プログラム先頭方向の、
	; ジャンプラベル「Label_0」にジャンプします。
N90 …	

例 2 :ブロック番号へ間接ジャンプ

プログラムコード	コメント
IF <条件> == TRUE	
R10=100	; ジャンプ先を割り当てます
ELSE	
R10=110	; ジャンプ先を割り当てます
ENDIF	
; プログラム末尾方向の、ブロック番号が R10 にあるブロックへジャンプします	
N10 GOTOF "N"<<R10	
...	
N90 ...	
N100 ...	; ジャンプ先
N110 ...	
...	

例 3 :ジャンプ先変数へジャンプ

プログラムコード	コメント
DEF STRING[20] DESTINATION	
IF <条件> == TRUE	
DESTINATION = "Label1"	; ジャンプ先を割り当てます
ELSE	
DESTINATION = "Label2"	; ジャンプ先を割り当てます
ENDIF	
; プログラムの末尾方向の、ジャンプ先となる変数「DESTINATION の内容」にジャンプします。	
GOTOF DESTINATION	
Label1:T="Drill1"	; ジャンプ先 1
...	
Label2:T="Drill2"	; ジャンプ先 2
...	

例 4 :ジャンプ条件によるジャンプ

プログラムコード	コメント
N40 R1=30 R2=60 R3=10 R4=11 R5=50 R6=20	; 初期値を割り当てます。
N41 LA1:G0 X=R2*COS(R1)+R5 Y=R2*SIN(R1)+R6	; ジャンプラベル LA1
N42 R1=R1+R3 R4=R4-1	
; IF ジャンプ条件 == TRUE	
; THEN プログラム先頭方向のジャンプラベル LA1 にジャンプします	
N43 IF R4>0 GOTOB LA1	
N44 M30	; プログラム終了

2.10.3 プログラム分岐(CASE ... OF ... DEFAULT ...)

CASE 機能を使用すると、変数または算術機能の現在値(タイプ:INT)を確認して、結果によっては、プログラムの別の位置へジャンプできます。

構文

```
CASE(<式>) OF <定数_1> GOTOF <ジャンプ先_1> <定数_2> GOTOF <ジャンプ先_2> ... DEFAULT GOTOF <ジャンプ先_n>
```

意味

CASE:	ジャンプ命令です。
<式>:	変数または算術機能です。
OF:	条件付きプログラム分岐の式を作成するためのキーワードです。

<定数_1>:	変数または算術機能に対して、1 番目に指定する定数値です。	
	タイプ:	INT
<定数_2>:	変数または算術機能に対して、2 番目に指定する定数値です。	
	タイプ:	INT
DEFAULT:	変数または算術機能が、指定した定数値を使用しない場合は、DEFAULT 命令を使用してジャンプ先を特定できます。 **注:** DEFAULT 命令をプログラム指令しない場合は、CASE 命令の次のブロックがジャンプ先になります。	
GOTOF:	プログラムの末尾方向へジャンプするジャンプ命令。 GOTOF の代わりに、すべての他の GOTO 命令をプログラム指令できます(「ジャンプマークへのプログラムジャンプ」を参照してください)。	
<ジャンプ先_1>:	変数または算術機能の値が、指定した 1 番目の定数に対応する場合は、分岐はこのジャンプ先となります。 ジャンプ先は、次のように指定できます。	
	<ジャンプマーク**>**:	ジャンプ先は、次のように、ユーザーが定義した名称でプログラムに設定されたジャンプマーク(ラベル)です。**<**ジャンプマーク**>:**
	<ブロック番号**>**:	ジャンプ先は、メインブロック番号またはサブブロック番号です(例:200, N300)。
	STRING タイプの変数:	ジャンプ先となる変数です。変数は、ジャンプマークまたはブロック番号を表わします。
<ジャンプ先_2>:	変数または算術機能の値が、指定した 2 番目の定数に対応する場合は、分岐はこのジャンプ先となります。	
<ジャンプ先_n>:	変数の値が、指定した定数値でない場合は、分岐はこのジャンプ先となります。	

例

プログラムコード

```
...
N20 DEF INT VAR1 VAR2 VAR3
```

プログラムコード

```
N30 CASE(VAR1+VAR2-VAR3) OF 7 GOTOF Label_1 9 GOTOF
Label_2 DEFAULT GOTOF Label_3
N40 Label_1:G0 X1 Y1
N50 Label_2:G0 X2 Y2
N60 Label_3:G0 X3 Y3
...
```

N30 の CASE 命令は、次のプログラム分岐の方法を定義します。

1. 算術機能 VAR1+VAR2-VAR3 の値が 7 の場合は、ジャンプマーク定義「Label_1」のあるブロックにジャンプします(→ N40)。
2. 算術機能 VAR1+VAR2-VAR3 の値が 9 の場合は、ジャンプマーク定義「Label_2」のあるブロックにジャンプします(→ N50)。
3. 算術機能 VAR1+VAR2-VAR3 の値が 7 でも 9 でもない場合は、ジャンプマーク定義「Label_3」のあるブロックにジャンプします(→ N60)。

2.11 プログラム区間の繰り返し(REPEAT、REPEATB、ENDLABEL、P)

プログラム区間の繰り返しを使用すると、1 つのプログラム内で既存のプログラム区間を任意の順序で繰り返すことができます。

繰り返すプログラム行またはプログラム区間は、ジャンプマーク(ラベル)が付けられます。

注記

ジャンプマーク(ラベル)

ジャンプマークは常に、ブロックの先頭に配置します。プログラム番号がある場合は、ブロック番号の直後にジャンプマークを配置します。

ジャンプマークの名称を付けるときは、次の規則が適用されます。

- 文字数:
  - 最小 2
  - 最大 32
- 次の文字を使用できます。
  - 英字
  - 数字
  - アンダースコア
- 最初の 2 文字は、英字またはアンダースコアにしてください。
- ジャンプマークの名称の後にはコロン(「:」)を付けます。

構文

1.個々のプログラム行の繰り返し:

```
<ジャンプマーク>: ...
...
REPEATB <ジャンプマーク> P=<n>
...
```

2.ジャンプマークと REPEAT 命令の間のプログラム区間の繰り返し:

```
<ジャンプマーク>: ...
...
REPEAT <ジャンプマーク> P=<n>
...
```

3.2 個のジャンプマーク間の区間の繰り返し:

```
<開始ジャンプマーク>: ...
...
```

```
<終了ジャンプマーク>: ...
...
REPEAT <開始ジャンプマーク> <終了ジャンプマーク> P=<n>
...
```

注記

括弧で 2 個のジャンプマークを含む REPEAT 命令を入れ子にすることはできません。<開始ジャンプマーク>が REPEAT 命令の前にあり、<終了ジャンプマーク>に到達する前に REPEAT 命令がある場合は、<開始ジャンプマーク>と REPEAT 命令の間の区間が繰り返されます。

4.ジャンプマークと ENDLABEL の間の区間の繰り返し:

```
<ジャンプマーク>: ...
...
ENDLABEL: ...
...
REPEAT <ジャンプマーク> P=<n>
...
```

注記

括弧で<ジャンプマーク>と ENDLABEL を含む REPEAT 命令を入れ子にすることはできません。<ジャンプマーク>が REPEAT 命令の前にあり、ENDLABEL に到達する前に REPEAT 命令がある場合は、<ジャンプマーク>と REPEAT 命令の間の区間が繰り返されます。

意味

REPEATB:	プログラム行を繰り返す命令です。
REPEAT:	プログラム区間を繰り返す命令です。

<ジャンプマーク>:	<ジャンプマーク>は、以下を示します。 ● 繰り返すプログラム行(REPEATB の場合) または ● 繰り返すプログラム区間の先頭(REPEAT の場合)
	<ジャンプマーク>で示されたプログラム行は、REPEAT/REPEATB 命令の前にある場合も、後にある場合もあります。検索はまず、プログラムの先頭方向へ開始されます。この方向にジャンプマークが見つからない場合は、検索が、プログラムの末尾方向へ続行されます。 **例外:** ジャンプマークと REPEAT 命令の間のプログラム区間を繰り返す(「構文」の 2 を参照してください)場合は、<ジャンプマーク>で示されたプログラム行が REPEAT 命令の**前**にあるように作成してください。これは、この場合には、プログラムの先頭方向**のみ**検索が実行されるためです。 <ジャンプマーク>を含む行に、さらに命令が含まれる場合は、繰り返し毎にそれらの命令が実行されます。
ENDLABEL:	繰り返すプログラム区間の末尾をマークするキーワードです。 ENDLABEL を含む行に、さらに命令が含まれる場合は、繰り返し毎にそれらの命令が実行されます。 ENDLABEL はプログラムで複数回、使用できます。
P:	繰り返し回数を指定するアドレスです。
<n>:	プログラム区間の繰り返し回数です。
	タイプ: INT
	繰り返すプログラム区間を<n>回、繰り返します。最後の繰り返しの後に、REPEAT/REPEATB 行の次の行からプログラムを再開します。 **注:** 回数を P=<n>で指定しない場合は、プログラム区間を 1 回だけ繰り返します。

例

例 1:個々のプログラム行の繰り返し

プログラムコード	コメント
N10 POSITION1:X10 Y20	
N20 POSITION2:CYCLE(0,,9,8)	; サイクルを位置決めします。
N30 ...	
N40 REPEATB POSITION1 P=5	; BLOCK N10 を 5 回実行します。
N50 REPEATB POSITION2	; N20 ブロックを 1 回実行します。
N60 ...	
N70 M30	

例 2 :ジャンプマークと REPEAT 命令の間のプログラム区間の繰り返し:

プログラムコード	コメント
N5 R10=15	
N10 Begin:R10=R10+1	; 幅
N20 Z=10-R10	
N30 G1 X=R10 F200	
N40 Y7=R10	
N50 X=-R10	
N60 Y=-R10	
N70 Z=10+R10	
N80 REPEAT BEGIN P=4	; N10 ～ N70 の区間を 4 回実行します。
N90 Z10	
N100 M30	

例 3 :2 個のジャンプマーク間の区間の繰り返し

プログラムコード	コメント
N5 R10=15	
N10 Begin:R10=R10+1	; 幅
N20 Z=10-R10	
N30 G1 X=R10 F200	
N40 Y7=R10	
N50 X=-R10	
N60 Y=-R10	
N70 END:Z = 10	
N80 Z10	
N90 CYCLE(10,20,30)	
N100 REPEAT BEGIN END P=3	; N10 ～ N70 の区間を 3 回実行します。
N110 Z10	
N120 M30	

例 4 :ジャンプマークと ENDLABEL の間の区間の繰り返し

プログラムコード	コメント
N10 G1 F300 Z-10	
N20 BEGIN1:	
N30 X10	
N40 Y10	
N50 BEGIN2:	
N60 X20	
N70 Y30	
N80 ENDLABEL:Z10	
N90 X0 Y0 Z0	
N100 Z-10	
N110 BEGIN3:X20	
N120 Y30	
N130 REPEAT BEGIN3 P=3	; N110 ～ N120 の区間を 3 回実行します。
N140 REPEAT BEGIN2 P=2	; N50 ～ N80 の区間を 2 回実行します。
N150 M100	
N160 REPEAT BEGIN1 P=2	; N20 ～ N80 の区間を 2 回実行します。
N170 Z10	
N180 X0 Y0	
N190 M30	

例 5 :フライス加工、さまざまなテクノロジによるドリル位置決め加工

プログラムコード	コメント
N10 CENTER DRILL()	; センタードリルを取り付けます。
N20 POS_1:	; 穴あけ位置 1
N30 X1 Y1	
N40 X2	
N50 Y2	
N60 X3 Y3	
N70 ENDLABEL:	
N80 POS_2:	; 穴あけ位置 2
N90 X10 Y5	
N100 X9 Y-5	
N110 X3 Y3	
N120 ENDLABEL:	
N130 DRILL()	; ドリルと穴あけサイクルを変更します。
N140 THREAD(6)	; タップ M6 とねじ切りサイクルをロードします。
N150 REPEAT POS_1	; POS_1 から ENDLABEL のプログラム区間を 1 回、繰り返します。
N160 DRILL()	; ドリルと穴あけサイクルを変更します。
N170 THREAD(8)	; タップ M8 とねじ切りサイクルをロードします。
N180 REPEAT POS_2	; POS_2 から ENDLABEL のプログラム区間を 1 回、繰り返します。
N190 M30	

詳細情報

- プログラム区間繰り返しは、入れ子にすることができます。各呼び出しは、サブプログラムレベルを使用します。
- プログラム区間繰り返しのときに M17 または RET をプログラム指令した場合は、繰り返しがキャンセルされます。プログラムは、REPEAT 行の次のブロックから再開します。
- 実際のプログラム表示では、プログラム区間繰り返しが別のサブプログラムレベルとして表示されます。
- プログラム区間繰り返しのときにレベルがキャンセルされた場合は、プログラム区間繰り返しを呼び出した後の地点からプログラムを再開します。
  例:

プログラムコード	コメント
N5 R10=15	
N10 Begin:R10=R10+1	; 幅
N20 Z=10-R10	
N30 G1 X=R10 F200	
N40 Y7=R10	; 繰り返しレベル中断
N50 X=-R10	
N60 Y=-R10	
N70 END:Z10	
N80 Z10	
N90 CYCLE(10,20,30)	
N100 REPEAT BEGIN END P=3	
N120 Z10	; プログラムの実行を再開します。
N130 M30	

- チェック命令とプログラム区間繰り返しは、組み合わせて使用できます。ただし、2 つの間に重複がないようにしてください。プログラム区間繰り返しがチェック命令分岐内にあるか、またはチェック命令がプログラム区間繰り返しの中にあるように作成してください。
- ジャンプとプログラム区間繰り返しが混在している場合は、ブロックは単純に順番どおりに実行されます。たとえば、ジャンプがプログラム区間繰り返し部から実行された場合は、プログラム指令したプログラム区間終了が見つかるまで、処理がおこなわれます。
  例:

プログラムコード
N10 G1 F300 Z-10
N20 BEGIN1:

プログラムコード

```
N30 X=10
N40 Y=10
N50 GOTOF BEGIN2
N60 ENDLABEL:
N70 BEGIN2:
N80 X20
N90 Y30
N100 ENDLABEL:Z10
N110 X0 Y0 Z0
N120 Z-10
N130 REPEAT BEGIN1 P=2
N140 Z10
N150 X0 Y0
N160 M30
```

注記

REPEAT 命令は、移動ブロックの後に指令してください。

2.12 チェック命令

コントローラは、標準では、プログラム指令順に処理します。

この順序は、プログラムブロック選択とプログラムループをプログラム指令して、可変にすることができます。これらのチェック命令は、キーワード IF、ELSE、ENDIF、LOOP、FOR、WHILE、REPEAT を使用してプログラム指令します。

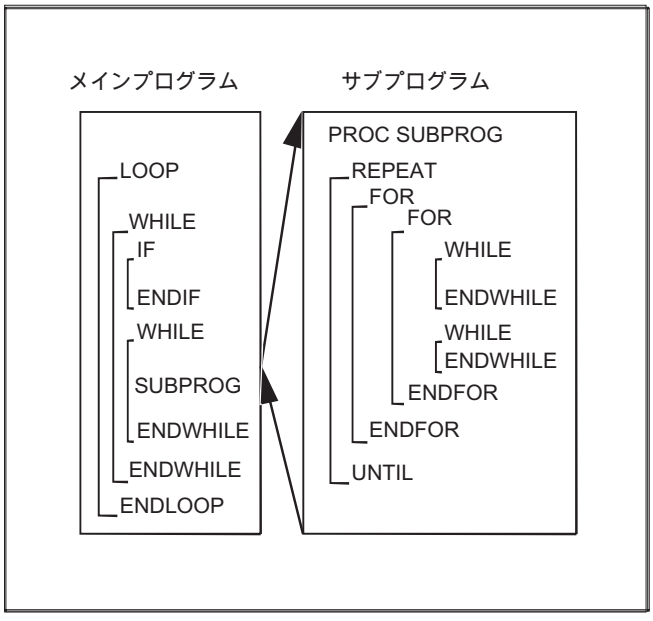
通知
プログラム指令エラー チェック命令は、プログラムの命令区間にのみに挿入できます。プログラムヘッダーの定義は、条件付きで実行することも、繰り返し実行することもできません。 チェック命令のキーワードにもジャンプ先にも、マクロを重ね合わせることはできません。マクロを定義した場合は、このチェックはおこなわれません。

有効性

チェック命令はプログラム全体で使用することはできません。

ネストレベル

各サブプログラムレベルには、入れ子の深さが 16 層までのチェック命令を設定できます。



動作の実行時間

インタプリタモード(標準で有効)では、プログラム分岐を使用すると、チェック命令より効率よく、プログラムの処理時間を短縮できます。

事前にコンパイルされたサイクルでは、プログラム分岐とチェック命令の間の差はありません。

プログラムループの実行中のブロックの表示

選択されたブロックだけがプログラムループ内で実行されている場合、プログラムループの**前の**最後のメインランブロックが実行中のブロックの表示に表示されます。

したがって、処理済みの選択されたブロックも実行中のブロックの表示に表示されます。たとえば、診断のために、デコードシングルブロック SBL2 を有効にする必要があります。

参照先

機能マニュアル、基本機能:モードグループ、チャネル、プログラム運転、リセット応答 (K1) > シングルブロック > 自動先読み停止によるデコードシングルブロック SBL2 の章

メインラインブロックなしでの研削

プログラムループ内で、メインランブロックがプログラムされている場合、ループ条件が満たされるまでループが先読みされます。

その結果、負荷率が高くなり、表示に悪影響が生じる場合があります。

対策として、STOPRE 命令、または 0 秒のドウェル時間 G04 を、ループに**挿入**できます。

必要条件

- チェック命令要素を含むブロックはマスクできません。
- ジャンプマーク(ラベル)は、チェック命令要素を含むブロックでは使用できません。
- チェック命令はインタプリタ的に処理されます。ループ終了を検出すると、ループ先頭の検索がおこなわれ、処理のなかに見つけたチェック命令を可能にします。このため、プログラムのブロック構成は、インタプリタモードで完全にはチェックされません。

- 一般的に、チェック命令とプログラム分岐を混ぜて使用することは望ましくありません。
- サイクルの解析時に、チェック命令が正しく入れ子になっていることを確認できます。

2.12.1 条件付き命令と分岐(IF、ELSE、ENDIF)

条件付き命令 IF - プログラムブロック - ENDIF

条件付き命令では、条件が満たされたときにだけ、IF と ENDIF の間のプログラムブロックが実行されます。

分岐:IF - プログラムブロック 1 - ELSE - プログラムブロック 2 - ENDIF

分岐では、2 つのプログラムブロックのうちの 1 つが常に実行されます。

条件が満たされた場合、IF と ELSE の間のプログラムブロック 1 が実行されます。

条件が**満たされなかった**場合、ELSE と ENDIF の間のプログラムブロック 2 が実行されます。

構文

条件付き命令

```
IF <条件>
プログラムブロック                    ; <条件> == TRUE の場合に実行
ENDIF
```

分岐

```
IF <条件>
   プログラムブロック 1               ; <条件> == TRUE の場合に実行
ELSE
   プログラムブロック 2               ; <条件> == FALSE の場合に実行
ENDIF
```

意味

IF:	条件付き命令または分岐を通知します。
ELSE:	プログラムブロック選択を開始します。

ENDIF:	条件付き命令または分岐の終了をマークします。
<条件>:	TRUE または FALSE として評価される論理式です。

例:工具交換サブプログラム

プログラムコード	コメント
PROC L6	工具交換プログラムです。
N500 DEF INT TNR_AKTUELL	動作中の T 番号の変数です。
N510 DEF INT TNR_VORWAHL	事前選択された T 番号用の変数
	現在の工具を特定します。
N520 STOPRE	
N530 IF $P_ISTEST	プログラムテストモードの場合...
N540 TNR_AKTUELL = $P_TOOLNO	...「現在の」工具がプログラム指令結果から読み取られます。
N550 ELSE	そうでない場合は...
N560 TNR_AKTUELL = $TC_MPP6[9998,1]	...主軸の工具が読み出されます。
N570 ENDIF	
N580 GETSELT(TNR_VORWAHL)	主軸で事前に選択した工具の T 番号を読み取ります。
N590 IF TNR_AKTUELL <> TNR_VORWAHL	事前に選択した工具がまだ、現在の工具ではない場合は、...
N600 G0 G40 G60 G90 SUPA X450 Y300 Z300 D0	...工具交換位置へアプローチして、...
N610 M206	...工具交換を実行します。
N620 ENDIF	
N630 M17	

2.12.2 連続プログラムループ(LOOP、ENDLOOP)

無限ループが無限プログラムで使用されます。ループの終了には常に、先頭へ戻る分岐があります。

構文

```
LOOP
...
ENDLOOP
```

意味

LOOP:	無限ループを開始します。
ENDLOOP:	ループの終了を示し、そして、ループの先頭へ復帰ジャンプします。

例

プログラムコード

```
...
LOOP
MSG ("no tool cutting edge active")
M0
Stopre
ENDLOOP
...
```

2.12.3 カウントループ(FOR ... TO ...、ENDFOR)

カウントループは、運転を固定の実行回数だけ繰り返して実行する場合に使用します。

構文

```
FOR <変数> = <開始値> TO <終了値>
...
ENDFOR
```

意味

FOR:	カウントループを開始します。
ENDFOR:	ループの終了を示し、そして、カウントの終了値に達していないかぎり、ループの先頭へ復帰ジャンプします。

<変数>:	開始値から終了値へと、実行のたびに値「1」ずつ増えていくカウント変数です。	
	タイプ	INT または REAL **注:** R 変数などをカウントループにプログラム指令している場合は、REAL タイプを使用します。カウント変数が REAL タイプの場合は、その値が整数に丸められます。
<開始値>:	カウントの開始値 条件:この開始値は、終了値より小さい数にしてください。	
<終了値>:	カウントの終了値	

例

例 1:INTEGER 変数、または R 変数のカウント変数

INTEGER 変数のカウント変数:

プログラムコード	コメント
DEF INT iVARIABLE1	
R10=R12-R20*R1 R11=6	
FOR iVARIABLE1 = R10 TO R11	; カウント変数= INTEGER 変数
R20=R21*R22+R33	
ENDFOR	
M30	

R 変数のカウント変数:

プログラムコード	コメント
R11=6	
FOR R10=R12-R20*R1 TO R11	; カウント変数= R 変数 (実数タイプの変数)
R20=R21*R22+R33	
ENDFOR	
M30	

例 2 :固定数の部品の製造

プログラムコード	コメント
DEF INT WKPCCOUNT	; INT タイプの変数を「WKPCCOUNT」という名称で定義します。

プログラムコード	コメント
FOR WKPCCOUNT = 0 TO 100	; カウントループを開始します。「WKPCCOUNT」変数は、開始値「0」から終了値「100」まで増えていきます。
G01 …	
ENDFOR	; カウントループの終了
M30	

2.12.4 ループの先頭に条件があるプログラムループ(WHILE、ENDWHILE)

WHILE ループの場合は、ループの先頭に条件があります。WHILE ループは、条件を満たしているかぎり、実行されます。

構文

```
WHILE <条件>
...
ENDWHILE
```

意味

WHILE:	プログラムループを開始します。
ENDWHILE:	ループの終了を示し、そして、ループの先頭へ復帰ジャンプします。
<条件>:	WHILE ループが実行されるように、条件を満たしてください。

例

プログラムコード	コメント
...	
WHILE $AA_IW[DRILL_AXIS] > -10	; 次の条件下で、WHILE ループを呼び出します。「穴あけ軸の実際の WCS 指令値が-10 より大きい」
G1 G91 F250 AX[DRILL_AXIS] = -1	
ENDWHILE	
...	

2.12.5 ループの終了に条件があるプログラムループ(REPEAT、UNTIL)

REPEAT ループの場合は、ループの終了で条件があります。REPEAT ループを実行したあと、条件が満たされるまで継続的に繰り返されます。

構文

```
REPEAT
...
UNTIL <条件>
```

意味

REPEAT:	プログラムループを開始します。
UNTIL:	ループの終了を示し、そして、ループの先頭へ復帰ジャンプします。
<条件>:	REPEAT ループが実行されないように、条件を満たします。

例

プログラムコード	コメント
...	
REPEAT	; REPEAT ループを呼び出します。
...	
UNTIL ...	; 条件が満たされているかどうかを確認します。
...	

2.12.6 入れ子のチェック命令を含むプログラム例

プログラムコード	コメント
LOOP	
IF NOT $P_SEARCH	; ブロック検索がおこなわれない場合
G1 G90 X0 Z10 F1000	
WHILE $AA_IM[X] <= 100	; WHILE (指令値 X 軸 <= 100)
G1 G91 X10 F500	; 穴あけパターン
Z-5 F100	
Z5	
ENDWHILE	

プログラムコード	コメント
ELSE	; ELSE ブロック検索
MSG("No drilling during block search")	
ENDIF	; ENDIF
$A_OUT[1] = 1	; 次のプレートの穴あけ
G4 F2	
ENDLOOP	
M30	

2.13 協調指令(INIT、START、WAITM、WAITMC、WAITE、SETM、CLEARM)

原則として、NC のチャネルは、その中で開始されたプログラムを、そのモードグループ内の他のチャネルとは無関係に処理できます。ただし、モードグループの複数のチャネルにある複数のプログラムがワークの加工に関与している場合は、異なるチャネルのプログラム処理を以下の協調コマンドによって協調させる必要があります。

必要条件

プログラム協調に使用するすべてのチャネルは、 **同じ** モードグループに属します。

MD10010 $MC_ASSIGN_CHAN_TO_MODE_GROUP[<チャネル>] = <モードグループ番号>

チャネル番号ではなくチャネル名

チャネル番号の代わりに MD20000 $MC_CHAN_NAME[<チャネルインデックス>]に入力されたチャネル名は、プログラム協調のための予約処理のパラメータとして使用することもできます。まず NC プログラムでチャネル名の使用を有効にする必要があります。

MD10280 $MN_PROG_FUNCTION_MASK, bit 1 = TRUE

構文

```
INIT(<ChanNr>, <Prog>, <AckMode>)
START(<ChanNr>, <ChanNr>, ...)
WAITM(<MarkNr>, <ChanNr>, <ChanNr>, ...)
WAITE(<ChanNr>, <ChanNr>, ...)
WAITMC(<MarkNr>, <ChanNr>, <ChanNr>, ...)
SETM(<MarkNr>, <MarkNr>, ...)
CLEARM(<MarkNr>, <MarkNr>, ...)
```

意味

INIT():	指定したチャネルで処理する NC プログラムを選択する予約処理
START():	それぞれのチャネルで選択したプログラムを開始するための予約機能
WAITM():	指定したチャネルで待機マークに到達するまで待機する予約処理 指定した待機マークは同じチャネルの WAITM で設定します。その前のブロックをイグザクトストップで終了します。待機マークは、同期制御が終わると解除されます。 チャネルごとに 10 個のマークを同時に設定できます。
WAITE():	複数の他のチャネルのプログラムの終了まで待機する予約処理

WAITMC(): [1]	指定したチャネルで待機マークに到達するまで待機する予約処理 WAITM とは異なり、イグザクトストップの減速は他のチャネルが待機マークに到達していない場合のみ開始します。		
SETM(): [1]	チャネル協調用の複数の待機マークを設定する予約処理 これはそのチャネル内の実行には影響しません SETM はチャネルリセットと NC スタート後も有効です。		
CLEARM(): [1]	チャネル協調用の複数の待機マークを削除する予約処理 これはそのチャネル内の実行には影響しません CLEARM() はチャネル内のすべての待機マークを削除します。 CLEARM(0) は待機マーク「0」のみを削除します。 CLEARM はチャネルリセットと NC スタート後も有効です。		
<ChanNr>:	チャネル番号 自身のチャネルの番号を指定する必要はありません。		
	タイプ:	INT	
<Prog>:	絶対または相対パス指定(**オプション**) + プログラム名称		
	タイプ:	STRING	
	パス指定については、以下を参照してください。 **参照先** 『プログラミングマニュアル、加工スケジュール』、「ファイルとプログラムの管理」>「プログラムメモリ」>「プログラムメモリのファイルのアドレス指定」のセクション (ページ 255)		
<AckMode>:	応答モード(**オプション**)		
	タイプ:	CHAR	
	規格値:	"n"	応答なし 命令の送信後にプログラムの実行を続行します。命令を正常に実行できない場合でも、送信側には通知されません。
		"S"	同期確認応答 プログラムの実行は、受信コンポーネントが命令を応答するまで中止します。確認の応答があった場合は、次の命令を実行します。確認の応答がない場合は、故障メッセージを出力します。

<MarkNr>:	待機マークの番号 **注** マルチチャネルシステムでは、最大 100 個の待機マーク(待機マーク 0 ～ 99)を使用できます。 シングルチャネルシステムで使用できるのは、待機マーク 0 のみです。
1) ユーザー別の通信/チャネル協調では、SETM / CLEARM を使用して待機マークを配置できます。また、条件付き待機指令 WAITMC を使用する必要もありません。チャネルリセットおよび NC スタートの後も、待機マークの値は保持されます。	

例

MD20000 からチャネル名の使用を開始する

- パラメータ設定

```
MD10280 $MN_PROG_FUNCTION_MASK, bit 1 = TRUE
$MC_CHAN_NAME[ 0 ] = "MACHINING"  ; チャネル 1 の名前
$MC_CHAN_NAME[ 1 ] = "INFEED"   ; チャネル 2 の名前
```

- プログラミング

プログラムコード	コメント
START(MACHINING)	; チャネル 1 の開始
START(INFEED)	; チャネル 2 の開始

START でローカルの「チャネル名称」と「ユーザー変数」を使用

プログラムコード	コメント
DEF INT MACHINE = 1	; チャネル 1 のユーザー変数の定義
DEF INT LOADER = 2	; チャネル 2 のユーザー変数の定義
...	
START(MACHINE)	; チャネル 1 の開始
START(LOADER)	; チャネル 2 の開始

START でローカルの「チャネル名称」、ユーザー変数およびパラメータ設定したチャネル名称を使用

プログラムコード	コメント
DEF INT chanNo1	; チャネル 1 のユーザー変数の定義
DEF INT chanNo2	; チャネル 2 のユーザー変数の定義
chanNo1 = CHAN_1	; パラメータ設定したチャネル名称チャネル 1 の割り当て
chanNo2 = CHAN_2	; パラメータ設定したチャネル名称チャネル 2 の割り当て
...	
START(chanNo1)	; チャネル 1 の開始

プログラムコード	コメント
START(chanNo2)	; チャネル 2 の開始

絶対パス指定による INIT 命令

チャネル 2 でのプログラム/_N_MPF_DIR/_N_ABSPAN1_MPF の選択

プログラムコード

```
INIT(2,"/_N_WCS_DIR/_N_SHAFT1_WPD/_N_CUT1_MPF")
```

プログラム名称を使用した INIT 命令

名称"MYPROG"のプログラムの選択検索パスを使用してプログラムが検索されます。

プログラムコード

```
INIT(2,"MYPROG")
```

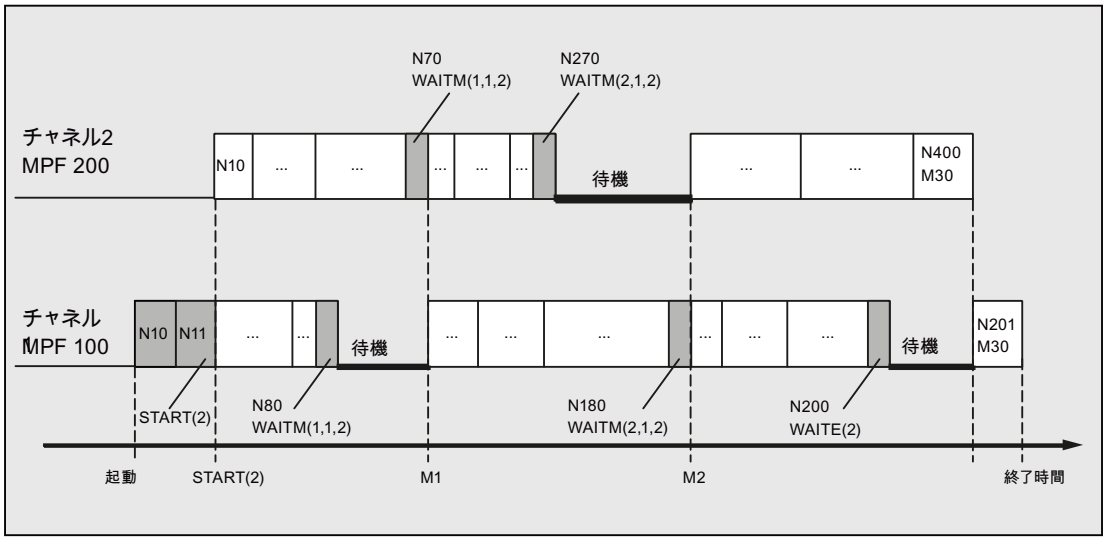
WAITM によるプログラム協調

- チャネル 1:プログラム/_N_MPF_DIR/_N_MPF100_MPF はすでに選択され、開始されています。

プログラムコード	コメント
	; プログラム MPF100
N10 INIT(2,"MPF200","N")	; プログラム MPF200 の選択、チャネル 2
N11 START(2)	; チャネル 2 の開始
...	
N80 WAITM(1,1,2)	; チャネル 1 と 2 で WAIT マーク 1 を待機
N81 ...	; チャネル 1、N81 とチャネル 2、N71 は ; 同時に開始されます
...	
N180 WAITM(2,1,2)	; チャネル 1 と 2 で WAIT マーク 2 を待機
N181 ...	; チャネル 1、N181 とチャネル 2、N271 は ; 同時に開始されます
...	
N200 WAITE(2)	;チャネル 2 のプログラムの終了を待機
N201 ...	; N201 はプログラム終了後にのみ開始されます ; チャネル 2 の MPF200
N201 M30	; チャネル 1 のプログラム終了

- チャネル 2:チャネル 1 で、ブロック N10 と N20 を使用してチャネル 2 のためにプログラム MPF200_MPF が選択され、開始されます。

プログラムコード	コメント
;$PATH=/_N_MPF_DIR	; プログラム MPF200
...	
N70 WAITM(1,1,2)	チャネル 1 と 2 で WAIT マーク 1 を待機
N71 ...	; チャネル 1、N81 とチャネル 2、N71 は ; 同時に開始されます
...	
N270 WAITM(2,1,2)	チャネル 1 と 2 で WAIT マーク 2 を待機
N271 ...	; チャネル 1、N181 とチャネル 2、N271 は ; 同時に開始されます
...	
N400 M30	チャネル 2 のプログラム終了



必要条件

待機マーク後の後続ブロック実行の非同期開始

待機マークを使用したチャネル協調の場合は、次のブロックの実行時に非同期開始が発生する可能性があります。この動作は、共通の待機マークに到達する直前に、チャネルのいずれかで操作がトリガされた場合に発生します。その結果、この残移動距離削除で自動的に再位置決め(REPOSA)が行われます。

前提条件:チャネル 1 と 2 での現在の軸割り当て

- チャネル 1:軸 X1 と軸 U
- チャネル 2:軸 X2

表 2-2　　チャネル 1 と 2 の処理の流れ

チャネル 1	チャネル 2	説明
...	...	チャネル 1 と 2 で任意の処理
N100 WAITM(20,1,2)		チャネル 1:WAIT マークに到達し、チャネル 2 との同期のために待機

チャネル 1	チャネル 2	説明
GETD(U)処理の開始: • *軸入れ替え* • *残移動距離削除* • *REPOSA* *終了*	N200 GETD(U)	チャネル 2:チャネル 1 に軸 U を要求 チャネル 1:バックグラウンドで GET(U) の処理
	N210 WAITM(20,1,2)	チャネル 2:WAIT マークに到達 ⇒ これによってチャネル 1 と 2 の同期が完了
	N220 G0 X2=100	チャネル 2:N220 の処理を開始
N110 G0 X1=100		チャネル 1:N110 の処理をずらして開始

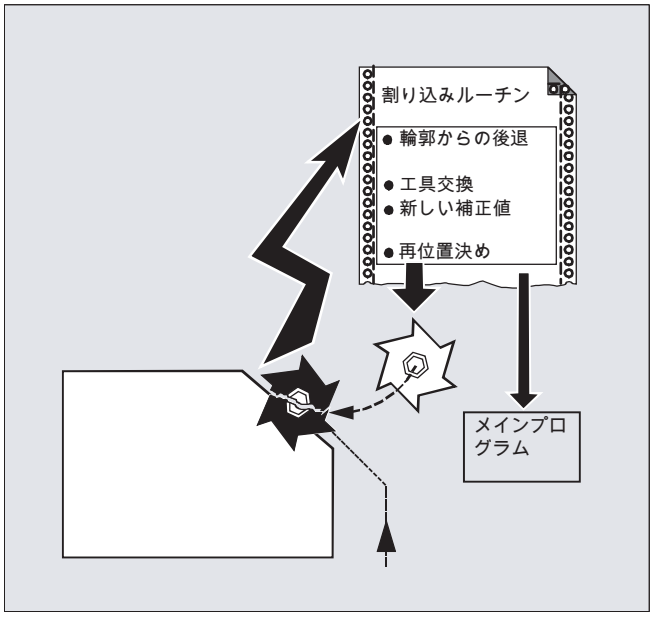
2.14 割り込みルーチン(ASUB)

2.14.1 割り込みルーチンの機能

注記

用語「非同期サブプログラム(ASUB)」と「割り込みルーチン」は、以下の説明で同義的に使用され、同じ機能を表わします。

次の一般的な例は、割り込みルーチンの機能をわかりやすく説明しています。



工具が加工中に破損します。これにより、現在の加工処理を停止する信号が出力され、同時に、割り込みルーチンと呼ばれるサブプログラムが起動されます。割り込みルーチンには、この場合に実行するすべての命令が含まれます。

割り込みルーチンの実行が終了し、機械が動作を続行できるようになると、制御がメインプログラムに戻り、割り込み発生点から加工を続行します。ただし、これは、REPOS 命令の場合は異なります(「 輪郭での再位置決め (ページ 640) 」を参照してください)。

⚠ 注意

干渉の可能性

REPOS 命令をサブプログラムにプログラム指令していない場合は、制御は、割り込まれたブロックの次のブロックの終点に移動します。

参照先

機能マニュアル 基本機能、モードグループ、チャネル、プログラム運転、リセット応答 (K1)、「非同期サブプログラム(ASUBs)、割り込みルーチン」の章

2.14.2 割り込みルーチンの作成

サブプログラムとしての割り込みルーチンの作成

割り込みルーチンは定義ではサブプログラムとして認識されます。

例:

プログラムコード	コメント
PROC LIFT_Z	; プログラム名称「ABHEB_Z」
N10 ...	; この後に NC ブロックが続きます。
...	
N50 M17	; 最後に、プログラムを終了し、メインプログラムに戻ります。

モーダル G 命令の保存(SAVE)

割り込みルーチンは、SAVE で定義して指定できます。

SAVE 属性は、動作中のモーダル G 命令が割り込みルーチンの前に保存され、割り込みルーチンの後に再度有効にされることを意味します(「SAVE メカニズムを持つサブプログラム(SAVE) (ページ 196)」を参照してください)。

つまり、中断点の処理は、割り込みルーチンが完了後に再開できます。

例:

プログラムコード
PROC LIFT_Z SAVE
N10 ...
...
N50 M17

追加の割り込みルーチンの割り当て(SETINT)

SETINT 命令を割り込みルーチン内にプログラム指令すると(「割り込みルーチンの割り当てと起動(SETINT)」 (ページ 149)を参照してください)、追加の割り込みルーチンを起動できます。これらの割り込みルーチンは入力で起動されます。

参照先

サブプログラムの作成方法について詳しくは、「サブプログラム、マクロ」の章を参照してください。

2.14.3 割り込みルーチンの割り当てと起動(SETINT、PRIO、BLSYNC)

制御装置には複数の高速入力(入力 1 ... 8)があり、これが割り込み(1 ... 8)を開始します。SETINT 命令を使用して、各割り込みに優先度と割り込みルーチンを割り当てることができます。高速入力を設定して割り込みを開始する場合、チャネルでの処理が中断され、割り込みルーチンが開始されます。

割り込みの優先度

パートプログラムで複数の入力が割り込みに割り当てられている場合、割り込みにそれぞれ異なる優先度を割り当ててください。

割り込みには、優先度の値 1 ... 128 を割り当てることができます。優先度の値 1 は最高優先度で、128 は最低優先度です。

構文

```
SETINT(<n>) <名称>
SETINT(<n>) PRIO=<値> <名称>
SETINT(<n>) PRIO=<値> <名称> BLSYNC
SETINT(<n>) PRIO=<値> <名称> LIFTFAST
```

意味

SETINT(<n>):	入力**<n>**は、割り込みルーチン**<**名称**>**に割り当てられます。割り当てられた割り込みルーチンは、入力**<n> == 1** が検出されると直ちに起動されます。 **注:** すでにプログラム指令された入力**<n>**が別の割り込みルーチンに割り当てられた場合、以前の割り当ては無効になります。	
<n>:	入力番号	
	タイプ:	INT
	値の範囲:	1 ... 8
PRIO= :	割り込みの優先度 (任意選択)	

<値>:	優先度の値 (任意選択)	
	タイプ:	INT
	値の範囲:	1 ... 128 (1 ⇒ 最高優先度)
<名称>:	割り込みルーチン(サブプログラム)の名称	
BLSYNC:	BLSYNC は、割り込みの開始後、まず現在のブロックが完了するまでシステムが待機するようにします。これが完了するまで割り込みルーチンは実行されません。 (任意選択)	
LIFTFAST:	LIFTFAST は、割り込みの開始後、まず高速リトラクトをおこなうようにします(「輪郭からの高速リトラクト(SETINT LIFTFAST、ALF) (ページ 153)」の章を参照してください)。これが完了するまで割り込みルーチンは実行されません。 (任意選択)	

必要条件

割り込み規則

1. すぐに実行できない、または現在すでに処理されているすべての割り込みに対して、追加の割り込み要求が保存されます。この割り込みに対するその他の割り込み要求はすべて失われます。
2. 割り込みが現在処理されていて、これよりも優先度の高い追加の割り込みが開始された場合、優先度の低い割り込みが中断されます。優先度の低い割り込みは、優先度の高い割り込みの完了後に続行されます。優先度の高い割り込みが処理されている間、優先度の低い割り込みに対する追加の要求を受信した場合、その要求は保存されます。その他すべての要求は失われます。
3. 割り込みが現在処理されていて、優先度の高い追加の割り込みが開始された場合、優先度の低い割り込みは中断されます。優先度の高い割り込みが処理されます。優先度の高い割り込みが開始された場合、現在の割り込みは中断され、優先度の高い割り込みが処理されます。6 つまでの有効な割り込みレベルを使用できます。1 つの割り込みレベルが現在処理され、5 つの割り込みレベルが待機します。有効な割り込みレベルそれぞれに対して、最大 1 つの追加の割り込み要求が保存されます。その他の割り込み要求はすべて失われます。その他の割り込みレベル(割り込みレベル ≥ 7)に対して割り込みが要求された場合も、割り込み要求は失われます。

例

例 1:割り込みルーチンの割り当てと優先度の定義

プログラムコード	コメント
...	

プログラムコード	コメント
N20 SETINT(3) PRIO=1 ABHEB_Z	; 入力 3 == 1 の場合、割り込みルーチン「ABHEB_Z」を起動します。
N30 SETINT(2) PRIO=2 ABHEB_X	; 入力 2 == 1 の場合、割り込みルーチン「ABHEB_X」を起動します。
...	

入力が同時に使用可能になった(同時に通電された)場合は、割り込みルーチンが優先度の値の順に実行されます。つまり、最初に「ABHEB_Z」、次に「ABHEB_X」が実行されます。

例 2:割り込みルーチンの新規割り当て

プログラムコード	コメント
...	
N20 SETINT(3) PRIO=2 ABHEB_Z	; 入力 3 == 1 の場合、割り込みルーチン「**ABHEB_Z**」を起動します。
...	
N80 SETINT(3) PRIO=1 ABHEB_X	; 入力 3 == 1 の場合、割り込みルーチン「**ABHEB_X**」を起動します。
...	

2.14.4 割り込みルーチンの割り当ての解除/再起動(DISABLE、ENABLE)

SETINT 命令は、DISABLE で無効に、ENABLE で、入力(割り込みルーチンの割り当て)をそのまま保持して、再度有効にできます。

構文

```
DISABLE(<n>)
ENABLE(<n>)
```

意味

DISABLE(<n>):	命令:割り込みルーチンの入力**<n>**の割り当ての**解除**	
ENABLE(<n>):	命令:割り込みルーチンの入力**<n>**の割り当ての**再設定**	
<n>:	パラメータ:割り込み信号番号	
	タイプ:	INT
	値の範囲:	1 ... 32

例

プログラムコード	コメント
N20 SETINT(3) PRIO=1 ABHEB_Z	; 入力 3 が切り替わると、割り込み
	; ルーチン「ABHEB_Z」が起動します。
...	
N90 DISABLE(3)	; N20 からの SETINT 命令を無効にします。
...	
N130 ENABLE(3)	; N20 からの SETINT 命令を再度有効にします。
...	

2.14.5 割り込みルーチンの割り当ての解除(CLRINT)

NC プログラム(ASUP)に対して SETINT で定義された割り込み信号の割り当てを CLRINT で解除することができます。

構文

```
CLRINT(<n>)
```

意味

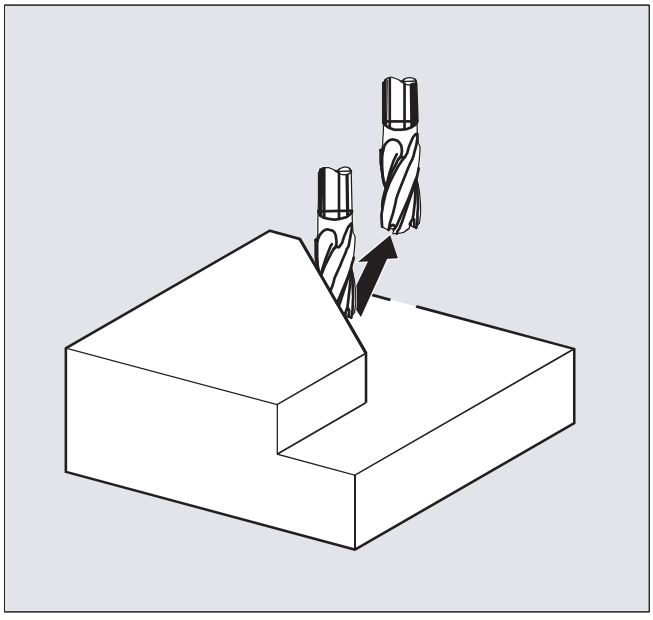
CLRINT(<n>):	命令:SETINT **<n>**で定義した NC プログラム(ASUP)への割り込み信号**<n>**の割り当てを解除します。	
<n>:	パラメータ:割り込み信号番号	
	タイプ:	INT
	値の範囲:	1 ... 32

例

プログラムコード	コメント
N20 SETINT(3) PRIO=2 ABHEB_Z	
...	
N50 CLRINT(3)	; 入力「3」と割り込みルーチン「ABHEB_Z」の割り当てを解除します。

2.14.6 輪郭からの高速リトラクト(SETINT LIFTFAST、ALF)

SETINT 命令が LIFTFAST と一緒に指令された場合は、入力が変化すると、工具がワーク輪郭から高速リトラクト機能を使って離れます。



それ以降の順序は、SETINT 命令が LIFTFAST の他に割り込みルーチンを含むかどうかにより変わります。

割り込みルーチンを含む 高速リトラクトの**後**に割り込みルーチンが実行されます。
場合:

割り込みルーチンを含ま 高速リトラクトの後に加工が停止し、アラームが発生します。
ない場合:

構文

```
SETINT(<n>) PRIO=1 LIFTFAST
SETINT(<n>) PRIO=1 <NAME> LIFTFAST
```

意味

SETINT(<n>):	命令:入力**<n>**を割り込みルーチンに割り当てます。割り当てた割り込みルーチンは、入力**<n>**が変化すると起動します。	
<n>:	パラメータ:入力番号	
	タイプ:	INT
	値の範囲:	1 ... 8
PRIO= :	優先度の定義	
<値>:	優先度の値	
	値の範囲:	1 ... 128
	優先度 1 が最高の優先度です。	
<NAME>:	実行するサブプログラム(割り込みルーチン)の名称です。	
LIFTFAST:	命令:輪郭からの高速リトラクト	
ALF=… :	命令:プログラム指令可能な(移動ブロックの)移動方向 ALF によるプログラミング方法については、「輪郭からの高速リトラクトの移動方向 (ページ 155)」の章を参照してください。	

必要条件

ミラーリングを使用した動作中のフレームの動作

後退方向を特定するときは、ミラーリングを使用したフレームが有効かどうかの確認をおこないます。この場合の後退方向は、接線方向を基準にして左右が入れ替わります。工具方向の方向成分はミラーリングされません。この動作は、次のマシンデータの設定により有効になります。

MD21202 $MC_LIFTFAST_WITH_MIRROR = TRUE

例

破損した工具は、自動的に予備工具へ交換されます。その後、新しい工具で加工が続行されます。

メインプログラム

メインプログラム	コメント
N10 SETINT(1) PRIO=1 W_WECHS LIFTFAST	; 入力 1 が変化すると、工具は直ちに輪郭から高速リトラクトで後退します (工具径補正 G41 のコード番号 7 の場合です)。そして、割り込みルーチン「W_WECHS」が実行されます。

メインプログラム	コメント
N20 G0 Z100 G17 T1 ALF=7 D1	
N30 G0 X-5 Y-22 Z2 M3 S300	
N40 Z-7	
N50 G41 G1 X16 Y16 F200	
N60 Y35	
N70 X53 Y65	
N90 X71.5 Y16	
N100 X16	
N110 G40 G0 Z100 M30	

サブプログラム:

サブプログラム	コメント
PROC W_CHANGE SAVE	; 実際の動作状態が保存されるサブプログラム
N10 G0 Z100 M5	; 工具交換位置、主軸停止
N20 T11 M6 D1 G41	;工具交換
N30 REPOSL RMBBL M3	; 輪郭に再位置決め、その後メインプログラムへ復帰ジャンプします (これを 1 ブロックにプログラム指令します)

2.14.7 輪郭からの高速リトラクトの移動方向

後退移動

次の G 命令は、後退移動平面を定義します。

- LFTXT
  後退移動平面は、軌跡タンジェントと工具方向(初期設定)で定義されます。
- LFWP
  後退移動平面は、G 命令 G17、G18 または G19 により選択された有効な作業平面です。後退移動の方向は、軌跡タンジェントには影響されません。このため、高速リトラクトは、軸と平行にプログラム指令できます。
- LFPOS
  POLFMASK/POLFMLIN を使用して宣言された軸が、POLF でプログラム指令されたアブソリュート軸位置へ後退します。
  ALF は、直線系の複数軸だけでなく、複数軸の後退方向にも影響を与えません。
  参照先:
  プログラミングマニュアル 基本編、「ねじ切り時の高速リトラクト」の章

プログラム指令可能な移動方向(ALF=...)

方向は、後退移動の平面で ALF により、45°の分割単位でプログラム指令します。

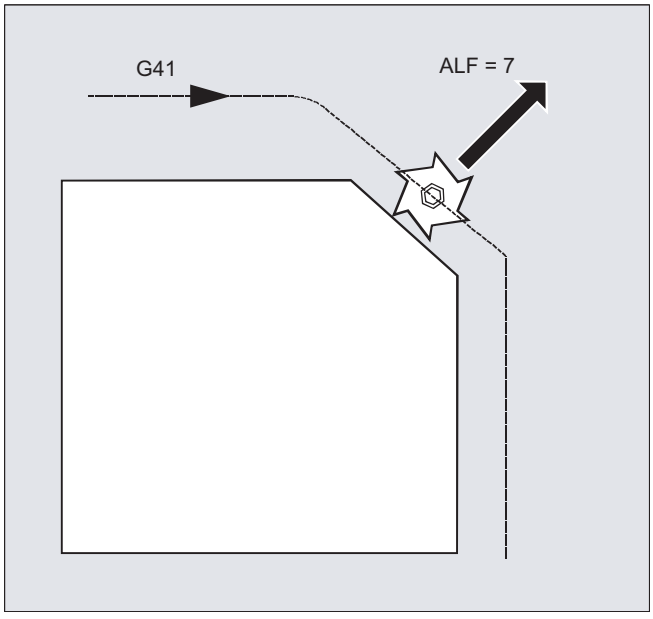
可能な移動方向は、コントローラに特殊コード番号で格納され、これらの番号を使用して呼び出すことができます。

例:

プログラムコード

```
N10 SETINT(2) PRIO=1 ABHEB_Z LIFTFAST
ALF=7
```

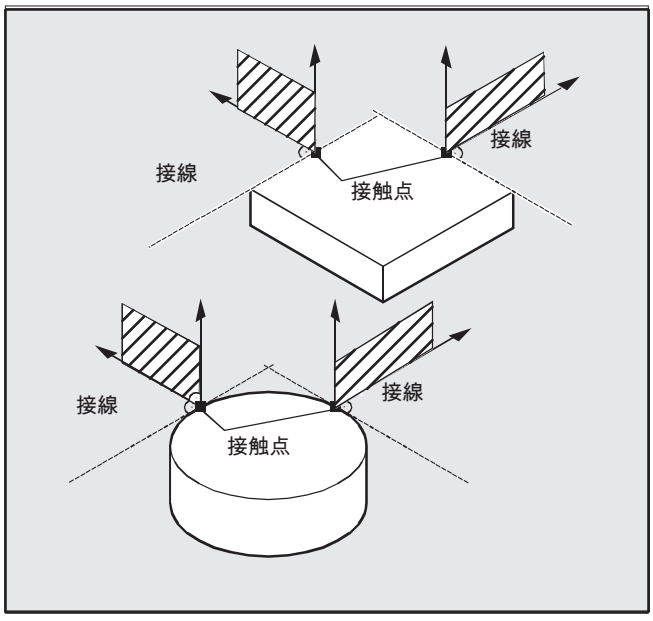
G41 を有効(加工方向は輪郭の左側)にした場合は、工具が輪郭から垂直方向に離れます。



LFTXT の移動方向を定義するための基準平面

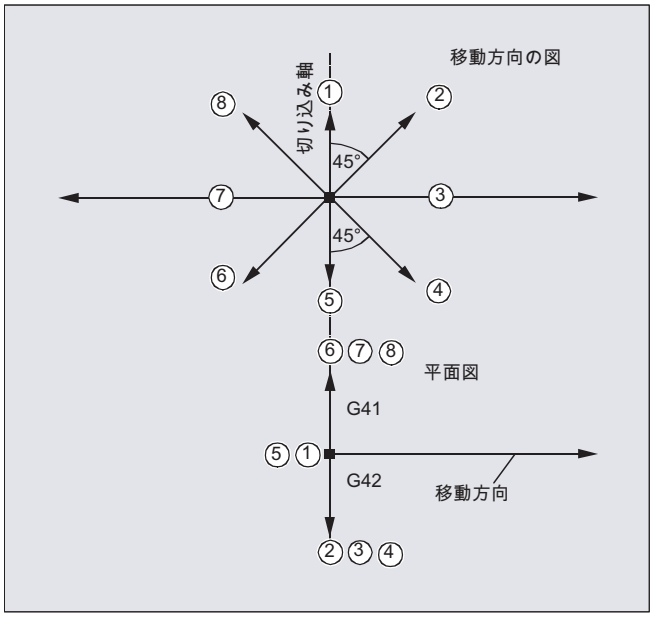
プログラム指令輪郭へ工具を適用するときに、工具は、対応するコード番号によって、戻し移動を指定するための基準として使用される平面で固定されます。

工具を適用時には、基準平面は、長手工具軸(切り込み方向)、この軸に垂直に位置するベクトル、およびこの接線に垂直に位置するベクトルから求められます。



LFTXT の移動方向を表すコード番号

基準平面から開始するために、次の図に移動方向を表すコード番号があります。



工具方向の後退は ALF=1 で定義されます。

「高速リトラクト」機能は ALF=0 で無効にします。

⚠ 注意

干渉の可能性

工具径補正を有効にした場合、次の規則を遵守してください。

- G41 の場合はコード 2、3、4
- G42 の場合はコード 6、7、8

工具が輪郭へ移動してワークに干渉するような可能性がある場合は、上記のコード番号は**使用しないで**ください。

LFWP の移動方向を表すコード番号

LFWP を使用すると、作業平面内の方向が、次の割り当てから導出されます。

- G17: X/Y 平面
  ALF=1:X 方向の後退
  ALF=3:Y 方向の後退
- G18: Z/X 平面
  ALF=1:Z 方向の後退
  ALF=3:X 方向の後退
- G19: Y/Z 平面
  ALF=1:Y 方向の後退
  ALF=3:Z 方向の後退

2.14.8 割り込みルーチンの動作順序

LIFTFAST を使用しない割り込みルーチン

軸移動は軌跡に沿って、停止状態まで減速します(ゼロ速度)。 その後、割り込みルーチンが開始されます。

停止状態の位置は、中断位置として保存され、RMIBL を使用した REPOS に対して、割り込みルーチンの終点でアプローチされます。

LIFTFAST を使用する割り込みルーチン

軸移動は、軌跡に沿って減速します。 LIFTFAST 移動は、重畳移動で同時に実行されます。 軌跡移動と LIFTFAST 移動が停止状態(ゼロ速度)となった場合に、割り込みルーチンが開始されます。

輪郭上の位置は中断位置として保存され、ここから LIFTFAST 移動が開始されて、軌跡から離れます。

LIFTFAST と ALF=0 を使用する割り込みルーチンは、LIFTFAST を使用しない割り込みルーチンと全く同じ方法で動作します。

注記

輪郭からの高速後退時にジオメトリ軸が移動するアブソリュート値は、マシンデータを使用して設定できます。

2.15 軸入れ替え、主軸入れ替え(RELEASE、GET、GETD)

複数の軸または主軸は、1 つのチャネルでのみ補間できます。1 つの軸が 2 つの異なるチャネル間で切り替わる(パレット交換など)場合は、最初に軸を現在のチャネルで有効にして、その後に別のチャネルに移行してください。軸入れ替えは、複数のチャネル間で有効です。

軸入れ替えの拡張機能

軸/主軸は、先読みとメインランの間の同期制御で、先読み停止あり、または先読み停止なしでも入れ替えできます。軸入れ替えは、次の方法によって有効にすることもできます。

- GET/GETD を自動的に使用して、軸コンテナ回転 AXCTSWE または AXCTWED で有効にできます。
- 軸入れ替え処理で、軸を他の軸と結合する場合は、回転フレームで有効にできます。
- シンクロナイズドアクションで有効にできます。これについては、「シンクロナイズドアクション」の「軸入れ替え RELEASE、GET」を参照してください。

工作機械メーカ

工作機械メーカの取扱説明書を参照してください。軸入れ替えの場合は、1 つの軸を、設定可能なマシンデータで、すべてのチャネルで一義的であるように定義してください。また、軸入れ替えの特性もマシンデータを使用して設定できます。

構文

RELEASE (軸名称、軸名称、....) または RELEASE (S1)

GET (軸名称、軸名称、....) または GET (S2)

GETD (軸名称、軸名称、....) または GETD (S3)

GETD (直接の GET 命令)の場合は、軸が直接に、別のチャネルから取得されます。このため、別チャネルでは、この GETD 命令用の RELEASE 命令をプグラム指令してはいけません。また、他のチャネルとの通信も確立してください(待機マークなど)。

意味

RELEASE (軸名称, 軸名称, ...):	軸を解放します。
GET (軸名称, 軸名称, ...):	軸を受け取ります。
GETD (軸名称, 軸名称, ...):	軸を直接受け取ります。

軸名称:	システムでの軸割り当て:AX1、AX2、...または機械軸名称を指定します。
RELEASE(S1)：	主軸 S1、S2、...を解放します。
GET(S2)：	主軸 S1、S2、...を受け取ります。
GETD(S3)：	主軸 S1、S2、...を直接受け取ります。

先読み停止のない GET 要求

先読み停止の**ない** GET 要求の後に、軸が RELEASE(軸) または WAITP(軸) で再度有効になった場合は、その後の GET が、先読み停止**あり**の GET になります。

⚠ 注意

軸割り当ての変更

GET で受け取られた軸または主軸は、キー RESET またはプログラム RESET の後でも、そのままこのチャネルに割り当てられます。

プログラムの再起動時に、元のチャネルで軸が必要となる場合は、入れ替えた軸または主軸をプログラムのなかで、再度割り当ててください。

電源投入時の軸の割り当ては、マシンデータで定義されたチャネルに割り当てられます。

例

例 1:2 つのチャネル間の軸入れ替え

6 つの軸のうち、チャネル 1 で加工に使用するのは、1 番目、2 番目、3 番目、および 4 番目の軸です。
チャネル 2 の 5 番目と 6 番目の軸は、ワークの入れ替えに使用します。

軸 2 は、2 つのチャネル間で入れ替える必要があり、電源投入後は、チャネル 1 に割り当てることができます。

チャネル 1 のプログラム「MAIN」:

プログラムコード	コメント
INIT (2,"TRANSFER2")	; チャネル 2 でプログラム TRANSFER2 を選択します。
N… START (2)	; チャネル 2 でプログラムを起動します。
N… GET (AX2)	;軸 AX2 を受け取ります。
...	
N… RELEASE (AX2)	; 軸 AX2 を解放します。
N… WAITM (1,1,2)	; チャネル 1 と 2 で WAIT マークを待機し、両チャネルで同期します。

プログラムコード	コメント
...	; 軸入れ替えの後のプログラムの残り部分です。
N… M30	

チャネル 2 のプログラム「TRANSFER2」:

プログラミング	コメント
N… RELEASE (AX2)	
N160 WAITM(1,1,2)	; チャネル 1 と 2 で WAIT マークを待機し、両チャネルで同期します。
N150 GET(AX2)	;軸 AX2 を受け取ります。
...	; 軸入れ替えの後のプログラムの残り部分です。
N… M30	

例 2:同期制御のない軸入れ替え

軸を同期する必要がない場合は、GET によって先読み停止はおこなわれません。

プログラミング	コメント
N01 G0 X0	
N02 RELEASE(AX5)	
N03 G64 X10	
N04 X20	
N05 GET(AX5)	; 同期制御が不要な場合は、このブロックの実行はできません。
N06 G01 F5000	; このブロックは実行できません。
N07 X20	; N04 の X 軸の位置決めと同じであるため、このブロックは実行できません。
N08 X30	; N05 の後に実行可能な最初のブロックです。
...	

例 3:先読み停止のない軸入れ替えの起動

必要条件:先読み停止のない軸入れ替えは、マシンデータで設定してください。

プログラミング	コメント
N010 M4 S100	
N011 G4 F2	
N020 M5	
N021 SPOS=0	
N022 POS[B]=1	
N023 WAITP(B)	; 軸 B は中立軸になります。
N030 X1 F10	
N031 X100 F500	

プログラミング	コメント
N032 X200	
N040 M3 S500	; 軸は先読み停止/REORG をおこないません。
N041 G4 F2	
N050 M5	
N099 M30	

たとえば主軸または軸 B が 180°の角度だけ移動した後に、**PLC 軸**として N023 ブロックの直後に 1°の角度だけ戻った場合、この軸は中立状態に戻ります、このため、N40 ブロックで先読み停止をおこないません。

詳細情報

軸入れ替えの必要条件

- 軸を使用するすべてのチャネルで、マシンデータを使用して軸を定義してください。
- 電源投入後に軸をどのチャネルに割り当てるかを、**軸**マシンデータで定義してください。

説明

軸の解放:RELEASE

軸を有効にするときの注意事項:

1. 軸は座標変換に使用しないでください。
2. 軸リンク(法線方向制御)に使用されるすべての軸を有効にしてください。
3. 同時位置決め軸は、この事例では入れ替えることができません。
4. ガントリマスタ軸のスレーブ軸はすべて、マスタ軸とともに移行します。
5. 連結軸の場合(連結移動、軸間連動機能、電子ギヤ)は、そのグループのマスタ軸のみを有効にすることができます。

軸の受け取り:GET;GET

実際の軸の入れ替えはこの命令でおこなわれます。プログラム指令した命令の対象であるチャネルが、軸を受け取ります。

GET の動作:

同期制御による軸入れ替え:

軸は、入れ替えのときに別のチャネルまたは PLC に割り当てたが、「WAITP」また G74 を実行して再原点同期をおこなっていない、または GET の前に残移動距離を削除した場合は、常に原点同期をおこなってください。

- その後に(STOPRE として)先読み停止が実行されます。
- 実行は、入れ替えが完了するまで中断されます。

自動の「GET」

軸が、原則としてチャネルで使用でき、現在「チャネル軸」として定義されている場合は、GET が自動的に実行されます。軸がすでに同期している場合は、先読み停止はおこなわれません。

軸入れ替え動作の応用

軸の移行タイミングは、マシンデータを使用して、次のように設定できます。

- WAITP で軸が中立状態になると、2 つのチャネル間で自動的に軸入れ替えもおこなわれます(以前と同じ動作です)。
- 軸コンテナ回転を要求すると、実行チャネルへの割り当てが可能な軸コンテナのすべての軸が、GET または GETD を自動的に使用して、このチャネルに入ります。軸コンテナ回転が完了すると、その後の軸入れ替えが、再び可能になります。
- 中間ブロックがメインランに挿入されると、確認がおこなわれ、再解析が必要かどうかが特定されます。再解析は、このブロックの軸ステータスが現在の軸ステータスに**一致しない**場合にのみ必要です。
- 先読みとメインランの間の、先読み停止と同期制御のある GET ブロックの代わりに、軸を先読み停止なしで入れ替えることができます。この場合は、単に中間ブロックが GET 要求により生成されます。メインランでこのブロックが実行されると、システムにより、このブロックの各軸のステータスが現在の軸ステータスに一致するかどうかを確認します。

軸または主軸の入れ替えの機能について詳しくは、
機能マニュアル上級機能、モードグループ、チャネル、チャネル間の待ち合わせと軸入れ替え(K5)を参照してください。

2.16 別のチャネルへの軸の移行(AXTOCHAN)

AXTOCHAN 言語命令を使用すると、別のチャネルに移動する軸を要求できます。軸は、NC パートプログラムとシンクロナイズドアクションの両方から、対応するチャネルに移動できます。

構文

```
AXTOCHAN(軸名称,チャネル番号[,軸名称,チャネル番号[,...]])
```

意味

要素	説明
AXTOCHAN:	特定のチャネル用の軸を要求します
軸名称:	システムでの軸割り当て:X、Y、…または、関連する機械軸名称の入力です。実行チャネルは、同じチャネルとは限りません。また、軸に対して現在の補間権限のあるチャネルとも限りません。
チャネル番号:	軸の割り当て先のチャネルの名称です。

注記

競合位置決め軸と PLC のみの制御軸

PLC 軸は、競合位置決め軸のようにチャネルを入れ替えることはできません。PLC のみが制御する軸を、NC プログラムに割り当てることはできません。

参照先:
機能マニュアル 上級機能; 位置決め軸(P2)

例

NC プログラムの AXTOCHAN

軸 X と Y は、1 番目と 2 番目のチャネルで宣言されています。現在は、チャネル 1 に補間権限があり、次のプログラムはこのチャネルで起動されます。

プログラムコード	コメント
N110 AXTOCHAN(Y,2)	; Y 軸を 2 番目のチャネルに移動します。
N111 M0	
N120 AXTOCHAN(Y,1)	;Y 軸を元に戻します(中立)。
N121 M0	
N130 AXTOCHAN(Y,2,X,2)	;Y 軸と X 軸を 2 番目のチャネルに移動します(両軸は中立です)。
N131 M0	

プログラムコード	コメント
N140 AXTOCHAN(Y,2)	;Y 軸を 2 番目のチャネルに移動します(NC プログラム)。
N141 M0	

詳細情報

NC プログラムの AXTOCHAN

GET は、同じチャネルの NC プログラムに対してその軸が要求された場合にのみ、実行されます(つまり、システムは状態が実際に変わるまで待機します)。その軸が別のチャネルに要求されるか、同じチャネルで中立軸になる場合は、それに従って要求が送られます。

シンクロナイズドアクションからの AXTOCHAN

同じチャネルに対して軸が要求される場合は、シンクロナイズドアクションの AXTOCHAN がシンクロナイズドアクションの GET に割り当てられます。この場合、軸は、同じチャネルに対する 1 回目の要求時に中立軸になります。2 回目の要求時には、NC プログラムの GET 要求と同様に、軸が NC プログラムに割り当てられます。シンクロナイズドアクションからの GET 要求について詳しくは、「シンクロナイズドアクション」を参照してください。

2.17 マシンデータの起動(NEWCONF)

NEWCONF 命令はすべてのマシンデータを有効化します。この機能は、HMI 操作画面で「マシンデータ有効化」ソフトキーを押して有効にすることもできます。

「NEWCONF」機能を実行すると、自動的に先読み停止が実行されます。つまり、軌跡移動が中断されます。

構文

```
NEWCONF
```

意味

NEWCONF:	「NEW_CONFIG」有効レベルのすべてのマシンデータを有効にする命令

パートプログラムからの、複数チャネルにわたる NEWCONF の実行

パートプログラムから軸のマシンデータが変更され、NEWCONF で起動された場合は、パートプログラムチャネルに影響する変更を含むマシンデータのみが NEWCONF で有効になります。

注記

すべての変更が確実に適用されるよう、NEWCONF 命令を、マシンデータの変更が影響する軸、または機能が計算される、すべてのチャネルで実行してください。

NEWCONF は軸マシンデータには影響しません。

PLC 制御軸には軸 RESET をおこなってください。

例

フライス加工:さまざまなテクノロジによるドリル位置決め加工

プログラムコード	コメント
N10 $MA_CONTOUR_TOL[AX]=1.0	;マシンデータを変更します。
N20 NEWCONF	;マシンデータを有効にします。
...	

2.18 ファイルの書き込み(WRITE)

WRITE 命令は、ブロック/データを NC プログラムからパッシブファイルシステム内のファイル(ログファイル)の末尾に、または外部のプログラムメモリに書き込みます。これは、現在実行中のプログラムでもかまいません。

注記

プログラムメモリに該当ファイルが存在しない場合は、1 つのファイルが作成され、WRITE 命令による書き込みが可能となります。

必要条件

現在設定されている保護レベルを、ファイルの WRITE 権のレベル以上にしてください。そうでない場合は、アクセスが拒否されてエラーメッセージが表示されます(エラー変数の戻り値= 13)。

構文

```
DEF INT <エラー>
...
WRITE(<エラー>,"<ファイル名称>"/"<ExtG>","<ブロック/データ>")
```

意味

WRITE:	指定したファイルの末尾にブロックまたはデータを付加する命令です。		
<エラー>:	**パラメータ 1:**エラー値を返すための変数		
	タイプ:	INT	
	値:	0	エラーなし
		1	パスは許容されません
		2	パスが見つかりません
		3	ファイルが見つかりません
		4	ファイルタイプが不正です
		10	ファイルがいっぱいです
		11	ファイルが使用中です
		12	使用可能なリソースがありません
		13	アクセス権がありません
		14	出力機器の EXTOPEN が欠落しているか正常に終了しませんでした
		15	外部機器への書き込み時にエラーが発生しました
		16	無効な外部パスがプログラム指令されています。
<ファイル名称>:	**パラメータ 2:**指定したブロックまたはデータを追加するファイルの名称です。		
	タイプ:	STRING	
	実際のファイル名の前に絶対パスを指定できます。パスを指定しない場合は、現在のディレクトリ(選択したプログラムのディレクトリ)でファイルを検索します。 パスデータに関する規則については、「プログラムメモリファイルのアドレス指定 (ページ 255)」を参照してください。		

<ExtG>:	「Process DataShare」機能を使用してデータを外部機器/ファイルに出力する場合は、ファイル名称ではなく、開こうとしている外部機器/ファイルのシンボルの識別子を指定してください。	
	タイプ:	STRING
	詳細については、「Process DataShare - 外部機器/ファイルへの出力 (EXTOPEN、WRITE、EXTCLOSE): (ページ 793)」を参照してください。 **注:** 識別子は EXTOPEN 命令で指定した識別子と同一のものにしてください。	
<ブロック/データ>:	**パラメータ 3:**指定したファイルに追加されるブロックまたはデータです。	
	タイプ:	STRING

注記

パッシブファイルシステムまたは外部プログラムメモリに書き込むときに、WRITE 命令により自動的に「LF」文字(LINE FEED = 新しい行)が出力文字列の末尾に挿入されます。この動作は、「Process DataShare」機能を使用した外部機器/ファイルへの出力には適用されません。「LF」も出力する場合は、出力文字列でこれを明示的に指定してください。
→ 例 3 も参照してください:自動的/明示的「LF」!

必要条件

- **最大ファイルサイズ(→ 工作機械メーカ)**
  パッシブファイルシステム内のログファイルの最大可能ファイルサイズは、次のマシンデータで設定されます。
  MD11420 $MN_LEN_PROTOCOL_FILE
  最大ファイル長は、WRITE 命令を使用して作成したパッシブファイルシステム内のすべてのファイルに適用されます。この値を超えた場合は、エラーメッセージが出力され、ブロックまたはデータは保存されません。十分な空きメモリがある場合は、新しいファイルを作成できます。

例

例 1:絶対パスデータを使用しないパッシブファイルシステムへの WRITE 命令

プログラムコード	コメント
N10 DEF INT ERROR	; エラー変数を定義します。
N20 WRITE(ERROR,"PROT","LOG FROM 7.2.97")	;「LOG FROM 7.2.97」からファイル _N_PROT_MPF にテキストを書き込みます。
N30 IF ERROR	; エラーを評価します。
N40 MSG ("Error with WRITE command:" <<ERROR)	
N50 M0	
N60 ENDIF	
...	

例 2:絶対パスデータを使用したパッシブファイルシステムへの WRITE 命令

プログラムコード
...
WRITE(ERROR,"/_N_WKS_DIR/_N_PROT_WPD/_N_PROT_MPF","LOG FROM 7.2.97")
...

例 3:自動的/明示的「LF」

a) 明示的に生成された「LF」ありのパッシブファイルシステムへの書き込み

プログラムコード
...
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

出力結果:

MY_STRING

MY_STRING

b) 自動的に生成された「LF」なしの外部ファイルへの書き込み

プログラムコード
...
N200 DEF STRING[30] DEV_1

プログラムコード

```
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
```

出力結果:

MY_STRINGMY_STRING

c) 明示的に生成された「LF」ありの外部ファイルへの書き込み

a の場合と同じ結果を達成するには、以下をプログラム指令してください。

プログラムコード

```
...
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
```

出力結果:

MY_STRING

MY_STRING

2.19 ファイルの削除(DELETE)

DELETE 命令は、WRITE 命令で作成したかどうかにかかわらず、すべてのファイルを削除します。より高いアクセス権限で作成したファイルも、DELETE で削除できます。

構文

```
DEF INT <エラー>
DELETE(<エラー>,"<ファイル名称>")
```

意味

DELETE:	指定したファイルを削除する命令です。		
<エラー>:	エラー値を返す変数です。		
	タイプ:	INT	
	値:	0	エラーなし
		1	パスは使用できません
		2	パスが見つかりません
		3	ファイルが見つかりません
		4	ファイルタイプが不正です
		11	ファイルが使用中です
		12	使用可能なリソースがありません
		20	その他のエラー
<ファイル名称>:	削除するファイルの名称		
	タイプ:	STRING	
	実際のファイル名の前に絶対パスを指定できます。パスを指定しない場合は、現在のディレクトリ(選択したプログラムのディレクトリ)でファイルを検索します。 パスデータに関する規則については、「プログラムメモリファイルのアドレス指定 (ページ 255)」を参照してください。		

例

プログラムコード	コメント
N10 DEF INT ERROR	; エラー変数を定義します。
N15 STOPRE	; 先読み停止

プログラムコード	コメント
N20 DELETE(ERROR,"/_N_SPF_DIR/_N_TEST1_SPF")	;サブプログラムのディレクトリのファイル TEST1 を削除します。
N30 IF ERROR	; エラーを評価します。
N40 MSG("error for DELETE command:" <<ERROR)	
N50 M0	
N60 ENDIF	

2.20 ファイルの行の読み取り(READ)

READ 命令は、指定したファイルの 1 行以上の行を読み取り、読み取った情報を STRING タイプの配列で保存します。この配列では、読み取った各行が 1 つの配列要素を使用します。

必要条件

現在設定されている保護レベルを、ファイルの READ 権レベル以上にしてください。そうでない場合は、アクセスが拒否されてエラーメッセージが表示されます(エラー変数の戻り値= 13)。

構文

```
DEF INT <エラー>
DEF STRING[<文字列長>] <結果>[<n>,<m>]
READ(<エラー>,"<ファイル名>",<開始行>,<行数>,<結果>)
```

意味

READ:	指定したファイルから行を読み取り、それらを配列変数に保存する命令です。		
<エラー>:	エラー値を返す変数です(参照渡しパラメータ)。		
	タイプ:	INT	
	値:	0	エラーなし
		1	パスは使用できません
		2	パスが見つかりません
		3	ファイルが見つかりません
		4	ファイルタイプが不正です
		11	ファイルが使用中です
		13	アクセス権が不十分です
		21	行が存在しません(<開始行>または<行数>パラメータが指定したファイルの行数を超えています)。
		22	結果変数(結果>)のフィールド長が短すぎます。
		23	行範囲が大きすぎます(選択した<行数>パラメータが大きすぎるため、ファイルの最後を超えて読み取られます)。

<ファイル名称>:	読み取るファイルの名称です(値渡しパラメータ)。		
	タイプ:	STRING	
	実際のファイル名の前に絶対パスを指定できます。パスを指定しない場合は、現在のディレクトリ(選択したプログラムのディレクトリ)でファイルを検索します。 パスデータに関する規則については、「プログラムメモリファイルのアドレス指定 (ページ 255)」を参照してください。		
<開始行>:	読み取るファイル区間の開始行(値渡しパラメータ)		
	タイプ:	INT	
	値:	0	<行数>パラメータで指定した行数をファイルの最後に到達する前まで読み取ります。
		1 ～ n	読み取る最初の行の番号です。
<行数>:	読み取る行数です(値渡しパラメータ)。		
	タイプ:	INT	
<結果>:	結果変数です(参照渡しパラメータ)。 読み取ったテキストの保存先の配列変数です。		
	タイプ:	STRING (最大長:255)	
	<行数>パラメータで指定した行数が、結果変数の配列サイズ [<n>,<m>] より少ない場合、残りの配列要素は変更されません。 制御文字「LF」(改行)または「CR LF」(復帰改行)による行の終了は、結果変数に**保存されません**。 読み取った行が、指定した文字列長より長い場合は、切り捨てられます。エラーメッセージは出力されません。		

注記

バイナリファイルは読み込むことができません。「不正なデータタイプ」エラーが出力されます(エラー変数の戻り値= 4)。次のタイプのファイルは読み取ることができません: _BIN、_EXE、_OBJ、_LIB、_BOT、_TRC、_ACC、_CYC、_NCK。

例

プログラムコード	コメント
N10 DEF INT ERROR	; エラー変数を定義します。
N20 DEF STRING[255] RESULT[5]	; 結果変数を定義します。

プログラムコード	コメント
N30 READ(ERROR,"/_N_CST_DIR/_N_TESTFILE_MPF", 1,5,RESULT)	; ドメインとファイル識別子、およびパス名称の付いた ファイル名称
N40 IF ERROR <>0	; エラーを評価します。
N50 MSG("ERROR"<<ERROR<<"ON READ COMMAND")	
N60 M0	
N70 ENDIF	
...	

2.21 ファイルの存在の確認(ISFILE)

ISFILE 命令は、ファイルがプログラムメモリに存在するかどうかをチェックします。

構文

<結果>=ISFILE("<ファイル名>")

意味

ISFILE:	ファイルの使用可能性をチェックするコマンド		
<ファイル名>:	使用可能性をチェックするファイルの名称。		
	タイプ:	STRING	
	実際のファイル名の前に絶対パスを指定できます。パスを指定しない場合は、現在のディレクトリ(選択したプログラムのディレクトリ)でファイルを検索します。 パスデータに関する規則については、「プログラムメモリファイルのアドレス指定 (ページ 255)」を参照してください。		
<結果>:	確認結果の割り当て先となる結果変数です。		
	タイプ:	BOOL	
	値:	TRUE	ファイルが存在します
		FALSE	ファイルが存在しません

例

例 1

プログラムコード	コメント
N10 DEF BOOL RESULT	; 結果変数を定義します。
N20 RESULT=ISFILE("TESTFILE")	
N30 IF(RESULT==FALSE)	
N40 MSG("ファイルが存在しません")	
N50 M0	
N60 ENDIF	
...	

例 2

プログラムコード	コメント
N10 DEF BOOL RESULT	; 結果変数を定義します。

プログラムコード	コメント
N20 RESULT=ISFILE("TESTFILE")	
N30 IF(NOT ISFILE("TESTFILE"))	
N40 MSG("ファイルが存在しません")	
N50 M0	
N60 ENDIF	
...	

2.22 ファイル情報の読み出し(FILEDATE、FILETIME、FILESIZE、FILESTAT、FILEINFO)

FILEDATE、FILETIME、FILESIZE、FILESTAT、および FILEINFO 命令は、最後の書き込みアクセスの日付/時刻、現在のファイルサイズ、ファイル状態、またはこれらの情報などすべて、特定のファイル情報を読み出します。

必要条件

現在設定されている保護レベルを、上位のディレクトリの表示権レベル以上にしてください。そうでない場合は、アクセスが拒否されてエラーメッセージが表示されます(エラー変数の戻り値= 13)。

構文

FILE....(<エラー>,"<ファイル名>",<結果>)

意味

FILEDATE:	ファイルへの最後の書き込みアクセスの**日付**を返します。
FILETIME:	ファイルへの最後の書き込みアクセスの**時刻**を返します。
FILESIZE:	ファイルの**現在のサイズ**を返します。
FILESTAT:	次の**権限**の中からファイルの**状態**を返します。 • 読み取り(r: read) • 書き込み(w: write) • 実行(x: execute) • 表示(s: show) • 削除(d: delete) **注:** これらの保護レベルはパッシブファイルシステムに固有の特性です。そのため、外部のプログラムメモリにアクセスする場合、FILESTAT は初期設定のアクセス権(77777)を返します。
FILEINFO:	ファイルに対して、FILEDATE、FILETIME、FILESIZE、FILESTAT から読み出せる**情報の合計**を返します。

<Error>:	エラー値を返す変数です(参照渡しパラメータ)。			
	タイプ:	VAR INT		
	値:	0	エラーなし	
		1	パスは使用できません	
		2	パスが見つかりません	
		3	ファイルが見つかりません	
		4	ファイルタイプが不正です	
		13	アクセス権が不十分です	
		22	結果変数の文字列長(<結果>)が短すぎます。	
<ファイル名>:	ファイル情報の読み出し元のファイルの名称です。			
	タイプ:	CHAR[160]		
	実際のファイル名の前に絶対パスを指定できます。パスを指定しない場合は、現在のディレクトリ(選択したプログラムのディレクトリ)でファイルを検索します。 パスデータに関する規則については、「プログラムメモリファイルのアドレス指定 (ページ 255)」を参照してください。			
<結果>:	結果変数(Call-By-Reference パラメータ) 要求したファイル情報の保存先となる変数です。			
	タイプ:	VAR CHAR[8]	情報	FILEDATE 形式:「dd.mm.yy」
		VAR CHAR[8]	情報	FILETIME 形式:「hh.mm.ss」
		VAR INT	情報	FILESIZE ファイルサイズはバイト単位で出力されます。
		VAR CHAR[5]	情報	FILESTAT 形式:「rwxsd」 (r: **r**ead, w: **w**rite, x: e**x**ecute, s: **s**how, d: **d**elete)
		VAR CHAR[32]	情報	FILEINFO 形式:「rwxsd nnnnnnnn dd.mm.yy hh:mm:ss」

例

プログラムコード	コメント
N10 DEF INT ERROR	; エラー変数を定義します。
N20 STRING[32] RESULT	; 結果変数を定義します。
N30 FILEINFO(ERROR,"/_N_MPF_DIR/ _N_TESTFILE_MPF",RESULT)	; ドメイン、ファイル識別子、およびパス指定を含むファイル名称です。
N40 IF ERROR <> 0	; エラー分析
N50 MSG("ERROR"<<ERROR<<"FOR FILE INFORMATION COMMAND")	
N60 M0	
N70 ENDIF	
...	

結果変数 RESULT では、例によって以下の結果が得られる場合があります。

"77777 12345678 26.05.00 13:51:30"

2.23 切り上げ(ROUNDUP)

入力値、REAL タイプ(小数点有りの有理数)は、「ROUNDUP」機能を使用して、次に大きい整数に切り上げることができます。

構文

```
ROUNDUP(<値>)
```

意味

ROUNDUP:	入力値を切り上げる命令
<値>:	入力値、REAL タイプ

注記

入力値が INTEGER タイプ(整数)の場合は、変更されずに返されます。

例

例 1:さまざまな入力値とその切り上げ結果

例	切り上げ結果
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

例 2:NC プログラムの ROUNDUP

プログラムコード

```
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])
...
```

2.24 サブプログラム機能

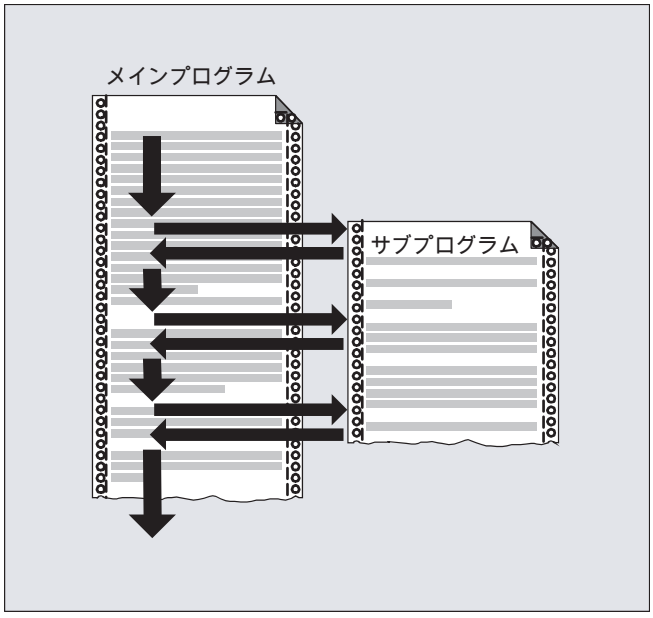
2.24.1 概要

2.24.1.1 サブプログラム

用語「サブプログラム」は、パートプログラムが、厳密にメインプログラムとサブプログラムに分類されたときから使用されるようになりました。メインプログラムとは、制御装置で処理をおこなうために選択され、起動されるパートプログラムでした。サブプログラムとは、メインプログラム内から呼び出されるパートプログラムでした。

今日の SINUMERIK NC 言語では、このように厳密には分類されていません。原則として、各パートプログラムは、メインプログラムとして選択して起動したり、サブプログラムとして別のパートプログラムから呼び出したりできます。

したがって、サブプログラムを使用すると、別のパートプログラム内から呼び出したパートプログラムを参照できます。



用途

すべての高レベルのプログラミング言語と同様に、NC 言語では、サブプログラムは、複数回使用されるプログラム区間を、独立した完全なプログラムに置き換えます。

サブプログラムには次の特長があります。

- プログラムをさらに使い易く、分かり易くする
- テスト済みのプログラム要素を再利用して品質を高める
- 特定の加工ライブラリの作成方法を提供する
- メモリ空間を節約する

2.24.1.2 サブプログラム名称

命名規則

サブプログラム名称は、以下の規則に従って自由に選択できます。

- 使用可能な文字:
  - 英字:A ... Z、a ... z
  - 数字:0 ... 9
  - アンダースコア: _
- 先頭の 2 文字は、英字 2 文字またはアンダースコアと英字 1 文字にしてください。

  注記

  この条件を満たす場合は、プログラム名称を指定するだけで、別のプログラムからサブプログラムとして NC プログラムを呼び出すことができます。ただし、プログラム名称の先頭が数字である場合、サブプログラム呼び出しは CALL 命令でのみ呼び出すことができます。

- 最大長さ:24 文字

注記

大文字/小文字

SINUMERIK NC 言語では、大文字と小文字は **区別されません**。

注記

使用できないプログラム名称

Windows アプリケーションで問題が生じることのないように、次のプログラム名の使用は **避けて** ください。

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

制御装置内部の拡張

サブプログラムの作成時に割り当てたプログラム名称には、制御装置内で接頭語と接尾語が追加され、拡張されます。

- 接頭語: _N_
- 接尾語: _SPF

プログラム名称の使用

サブプログラム呼び出し処理などでプログラム名称を使用するときは、接頭語、プログラム名称、および接尾語のすべての組み合わせを使用できます。

例:

プログラム名称が「SUB_PROG」のサブプログラムは、次の識別子で起動できます。

1. SUB_PROG
2. _N_SUB_PROG
3. SUB_PROG_SPF
4. _N_SUB_PROG_SPF

同じ名称のメインプログラムとサブプログラム

同じ名称のメインプログラム(.MPF)とサブプログラム(.SPF)が存在する場合は、NC プログラムでその名称を使用するときに、プログラムが一義的に識別されるように、適切なファイル拡張子を指定してください。そうしない場合、検索パスで最初に見つかった、指定した名称のプログラムが使用されます。

2.24.1.3 サブプログラムの入れ子

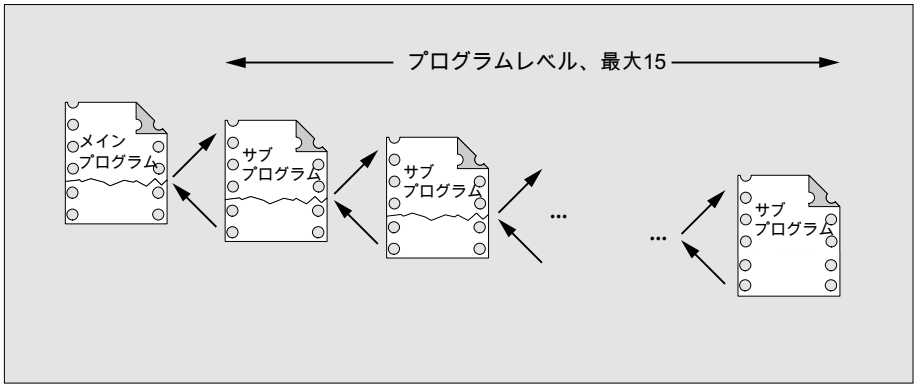
メインプログラムはサブプログラムを呼び出すことができ、そのサブプログラムはさらにサブプログラムを呼び出すことができます。 このようにして、プログラムの順序はそれぞれに入れ子となります。 各プログラムは、専用のプログラムレベルで実行されます。

ネストレベル

NC 言語では現在、16 個のプログラムレベルが使用できます。 メインプログラムは常に、最上位のプログラムレベル、0 で実行します。サブプログラムは常に、呼び出しレベルの次に高いプログラムレベルで実行します。 したがって、プログラムレベル 1 が最初のサブプログラムレベルです。

プログラムレベルの分類:

- プログラムレベル 0: メインプログラムのレベル
- プログラムレベル 1 ～ 15: サブプログラムレベル 1 ～ 15:



割り込みルーチン(ASUB)

サブプログラムを割り込みルーチン処理で呼び出した場合は、チャネル(n)で現在有効なプログラムレベルではなく、その次に低いプログラムレベル(n+1)でサブプログラムが実行されます。 現在のレベルが、最低のプログラムレベルでも可能となるよう、割り込みルーチンとと組み合わせて、2 個の追加プログラムレベル(16 と 17)を使用できます。

2 個を超えるプログラムレベルが必要な場合は、そのチャネルで実行するパートプログラムの構築時に、これを別途考慮してください。 つまり、割り込み処理に必要な、十分な数のプログラムレベルの最大数まで、使用する場合があります。

たとえば、割り込み処理に 4 個のプログラムレベルが必要な場合、13 までのプログラムレベルを使用するパートプログラムを構成してください。これで、割り込みが生じた場合に、必要な 4 個のプログラムレベル(14 ～ 17)を使用できます。

SIEMENS サイクル

SIEMENS サイクルには 3 個のプログラムレベルが必要です。 したがって、SIEMENS サイクルは、少なくとも次のプログラムレベルで呼び出してください。

- パートプログラムの処理: プログラムレベル 12
- 割り込みルーチン: プログラムレベル 14

2.24.1.4 検索パス

パスを指定せずにサブプログラムを呼び出すと、コントロールシステムは事前定義された検索順序を使用して使用可能なプログラムメモリを検索します(「サブプログラム呼び出しの検索パス (ページ 261)」を参照してください)。

2.24.1.5 仮パラメータと実パラメータ

仮パラメータと実パラメータは、定義と、パラメータ転送を含むサブプログラム呼び出しを組み合わせると発生します。

仮パラメータ

サブプログラムを定義すると、サブプログラムに転送されるパラメータ(仮パラメータと呼ばれます)を、タイプとパラメータ名称で定義する必要があります。

したがって、仮パラメータはサブプログラムのインタフェースです。

例:

プログラムコード	コメント
PROC CONTOUR (REAL X, REAL Y)	; 仮オペランド:X と Y、ともに REAL タイプです。
N20 X1=X Y1=Y	; 軸 X1 が位置 X へ、および軸 Y1 が位置 Y へ移動
...	
N100 RET	

実パラメータ

サブプログラムを呼び出すときは、それに絶対値または変数(実パラメータと呼ばれます)を転送する必要があります。

そして、実パラメータは、サブプログラム呼び出し時に、そのインタフェースに最新値を割り当てます。

例:

プログラムコード	コメント
N10 DEF REAL WIDTH	;変数の定義
N20 WIDTH=20.0	; 変数割り当て
N30 CONTOUR(5.5, WIDTH)	; 実パラメータによるサブプログラム呼び出し:5.5 と WIDTH
...	
N100 M30	

2.24.1.6 パラメータ転送

パラメータ転送を含むサブプログラムの定義

パラメータ転送を含むサブプログラムは、PROC キーワード、とサブプログラムが予期しているすべてのパラメータのリスト全体を使用して定義されます。

不完全なパラメータ転送

サブプログラムの呼び出し時に、サブプログラムインタフェースで定義されたすべてのパラメータを明示的に転送する必要があるわけではありません。パラメータを省略すると、初期値「0」が転送されます。

ただし、パラメータの順序が一義的に識別されるよう、パラメータ区切り文字として、コンマを含めてください。最後のパラメータは例外です。呼び出しで最後のパラメータを省略した場合は、最後のコンマも省略できます。

例:

サブプログラム:

プログラムコード	コメント
PROC SUB_PROG (REAL X, REAL Y, REAL Z)	; 仮オペランド:X、Y、および Z
...	
N100 RET	

メインプログラム

プログラムコード	コメント
PROC MAIN_PROG	
...	
N30 SUB_PROG(1.0,2.0,3.0)	; 完全なパラメータの転送によるサブプログラム呼び出し: X=1.0、Y=2.0、Z=3.0
…	
N100 M30	

N30 に、不完全なパラメータ転送によるサブプログラム呼び出しの例:

```
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
```

通知
参照渡しパラメータ転送 参照渡しで転送されたパラメータは、サブプログラム呼び出しでは省略しないでください。

通知
AXIS データタイプ AXIS データタイプのパラメータは、サブプログラム呼び出しでは省略しないでください。

転送パラメータのチェック

システム変数$P_SUBPAR [n] (ここで n = 1, 2, ...)を使用すると、パラメータが明示的に転送されたか、サブプログラムで省略されたかをチェックできます。インデックス n は、仮パラメータの順序を表わします。インデックス n = 1 は、1 番目の仮パラメータ、インデックス n = 2 は 2 番目の仮パラメータを表わし、以降も同様です。

次のプログラム引用部分は、1 番目の仮パラメータに基づいてチェックがおこなわれる例を示します。

プログラミング	コメント
`PROC SUB_PROG (REAL X, REAL Y, REAL Z)`	`;` 仮オペランド:X、Y、および Z
`N20 IF $P_SUBPAR[1]==TRUE`	`;` 1 番目の仮オペランド X のチェック。
`...`	`;` この部分の動作は、仮オペランド X が明示的に転送された場合に実行されます。
`N40 ELSE`	
`...`	`;` この部分の動作は、仮オペランド X が転送されなかった場合に実行されます。
`N60 ENDIF`	
`...`	`;` 一般動作
`N100 RET`	

2.24.2 サブプログラムの定義

2.24.2.1 パラメータ転送を含まないサブプログラム

パラメータ転送を含まないサブプログラムを定義するときは、プログラムの先頭の定義行を省略できます。

構文

```
[PROC <プログラム名称>]
...
```

意味

PROC:	プログラムの先頭の定義演算
<プログラム名称>:	プログラムの名前

例

例 1:PROC 命令のあるサブプログラム

プログラムコード	コメント
PROC SUB_PROG	; 定義行
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	; サブプログラム戻り

例 2 :PROC 命令のないサブプログラム

プログラムコード	コメント
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	; サブプログラム戻り

下記も参照

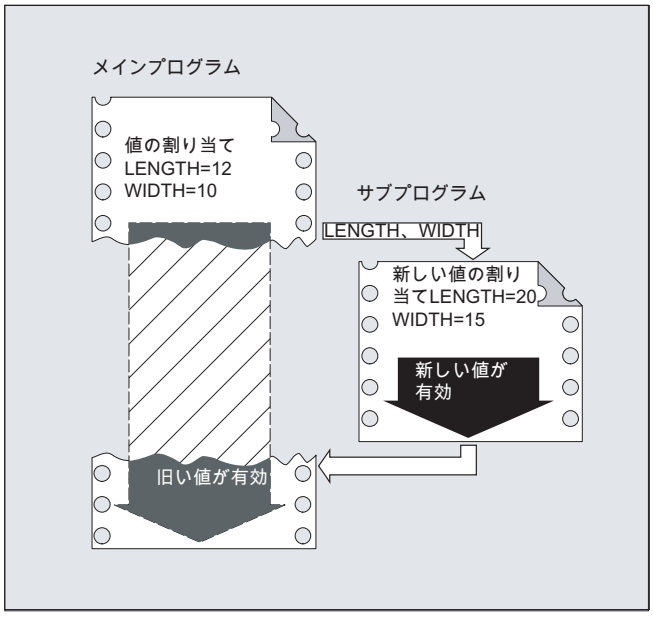
パラメータ転送を含まないサブプログラム呼び出し (ページ 221)

2.24.2.2 値渡しパラメータ転送を含むサブプログラム(PROC)

値渡しパラメータ転送を含むサブプログラムは、PROC キーワードと、その次にプログラム名称、その次にすべてのパラメータのタイプと名称の完全なリストを使用して定義します。定義演算は、最初のプログラム行に入れてください。

値渡し

呼び出しプログラムは、値渡しパラメータ転送時に変数の値をサブプログラムに転送するだけです。そのため、サブプログラムには変数の直接アクセスは与えられません。このように、パラメータ値が変更されるときに、サブプログラム内に表示される値だけが変更されます。呼び出しプログラムで定義された変数の値は変更されないままです。結果として、値渡しパラメータ転送は、呼び出しプログラムには影響しません。



構文

```
PROC <プログラム名称> (<パラメータタイプ> <パラメータ名称>=<初期値>, ...)
```

注記

127 個までのパラメータを転送できます。

意味

PROC:	プログラムの先頭の定義演算
<プログラム名称>:	プログラムの名称
<パラメータタイプ>:	パラメータのデータタイプ(REAL、INT、BOOL など)

<パラメータ名称>:	パラメータの名称
<初期値>:	パラメータの初期値の選択(**オプション**) サブプログラムを呼び出したときにパラメータが指定されていない場合は、パラメータに初期値が割り当てられます。

例

例 1

初期値を持つ REAL タイプの 3 つのパラメータによるサブプログラム SUB_PROG の定義

プログラムコード

```
PROC SUB_PROG(REAL LENGTH=10.0, REAL WIDTH=20.0, REAL HIGHT=30.0)
```

例 2

さまざまな呼び出しバージョン

プログラムコード

```
PROC MAIN_PROG
  REAL PAR_1 = 100
  REAL PAR_2 = 200
  REAL PAR_3 = 300
  ; 呼び出し変数
  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
```

下記も参照

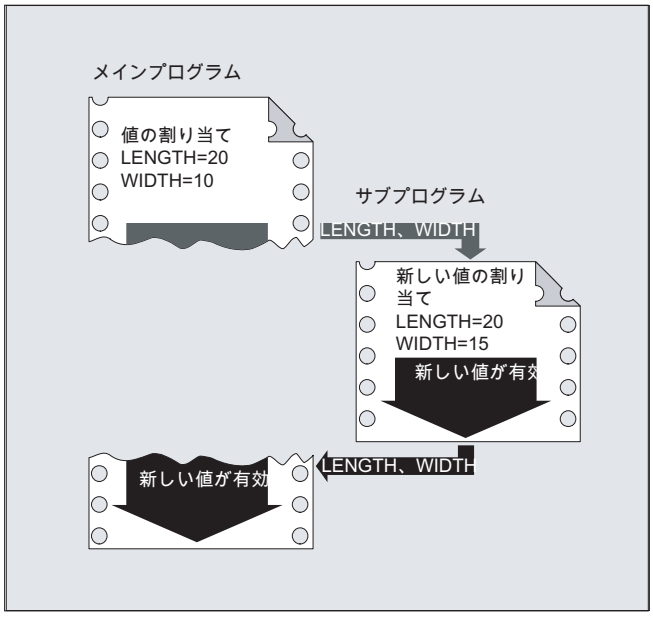
パラメータ転送を含むサブプログラム呼び出し(EXTERN) (ページ 224)

2.24.2.3 参照渡しパラメータ転送を含むサブプログラム(PROC、VAR)

参照渡しパラメータ転送を含むサブプログラムは、PROC キーワードと、その次にプログラム名称、その次にすべてのパラメータの VAR キーワード、タイプ、および名称のリスト全体を使用して定義します。定義演算は、最初のプログラム行に入れてください。パラメータとして、配列に対する参照先も転送できます。

参照渡し

呼び出しプログラムは、参照渡しパラメータ転送時に変数の値をサブプログラムに転送するだけでなく、変数への参照(ポインタ)も転送します。これにより、サブプログラムに変数への直接アクセスが与えられます。このように、パラメータ値が変更されるときに、サブプログラム内に表示される値が変更されるだけでなく、呼び出しプログラムで定義された変数の値も変更されます。したがって、参照渡しパラメータ転送は、サブプログラムの終了後も呼び出しプログラムに影響を与えます。



注記

参照渡しパラメータ転送が必要となるのは、転送した変数を呼び出しプログラムで(LUD)でローカルに定義している場合のみです。チャネルグローバル変数または NC グローバル変数は、サブプログラム内から直接アクセスできないため、これらの変数を転送しないでください。

構文

```
PROC <プログラム名称> (VAR <パラメータタイプ> <パラメータ名称>, ...)
PROC <プログラム名称> (VAR <配列タイプ><配列名称>, [<m>,<n>,<o>], ...)
```

注記

127 個までのパラメータを転送できます。

意味

PROC:	プログラムの先頭の定義演算	
VAR:	参照によるパラメータ転送のキーワード	
<プログラム名称>:	プログラムの名称	
<パラメータタイプ>:	パラメータのデータタイプ(REAL、INT、BOOL など)	
<パラメータ名称>:	パラメータの名称	
<配列タイプ>:	配列要素のデータタイプ(REAL、INT、BOOL など)	
<配列名称>:	配列の名称	
[<m>,<n>,<o>]:	配列サイズ 現在は、3 次元までの配列を使用できます。	
	<m>:	1 次元の配列サイズ
	<n>:	2 次元の配列サイズ
	<o>:	3 次元の配列サイズ

注記

- PROC キーワードの後に指定するプログラム名称は、操作画面で割り当てたプログラム名称と一致させてください。
- 配列長を定義していない配列を使用すると、サブプログラムは、可変長の配列を仮パラメータとして処理できます。たとえば、2 次元の配列を仮パラメータとして定義すると、1 次元の長さは指定されません。ただし、コンマは入れてください。
  例:PROC <プログラム名称> (VAR REAL ARRAY[,5])

例

2 個のパラメータを含むサブプログラムを、REAL タイプの参照として定義:

プログラムコード

```
; パラメータ 1:タイプの参照:REAL、名称:LENGTH
; パラメータ 2:タイプの参照:REAL、名称:WIDTH
PROC SUB_PROG(VAR REAL LENGTH, VAR REAL WIDTH)
```

下記も参照

パラメータ転送を含むサブプログラム呼び出し(EXTERN) (ページ 224)

2.24.2.4 モーダル G 機能の保存(SAVE)

SAVE 属性は、サブプログラム呼び出しの前に、動作中のモーダル G 命令が保存され、サブプログラムの終了後に再度有効になることを意味します。

通知
連続軌跡モードの中断 連続軌跡モードが動作中の場合に、サブプログラムが SAVE 属性で呼び出されると、サブプログラムの終了時に連続軌跡モードが中断します(復帰ジャンプ)。

構文

```
PROC <サブプログラム名称> SAVE
```

意味

SAVE:	サブプログラム呼び出しの前にモーダル G 命令を保存し、サブプログラムの終了後に復帰します。

例

CONTOUR サブプログラムには、モーダル G 命令 G91 が適用されます。モーダル G 命令 G90 は、メインプログラムで有効です(アブソリュート指令)。G90 は、SAVE によるサブプログラム定義により、サブプログラムの終了後に再度メインプログラムで有効になります。

サブプログラム定義:

プログラムコード	コメント
PROC CONTOUR (REAL VALUE1) SAVE	; SAVE パラメータによるサブプログラムの定義
N10 G91 ...	; モーダル G 命令 G91:インクレメンタル指令
N100 M17	; サブプログラム終了

メインプログラム

プログラムコード	コメント
N10 G0 X...Y...G90	; モーダル G 命令 G90:アブソリュート指令

プログラムコード	コメント
N20 ...	
...	
N50 CONTOUR (12.4)	; サブプログラム呼び出し
N60 X...Y...	; SAVE を使用してモーダル G 命令 G90 を再度有効にします

必要条件

フレーム

SAVE 属性を含むサブプログラムに関するフレームの動作は、フレームタイプによって異なっており、マシンデータで設定できます。

参照先

機能マニュアル 基本機能; 軸、座標系、フレーム(K2)、
「SAVE 属性をもつサブプログラムからの復帰」の章

2.24.2.5 シングルブロック実行のマスク(SBLOF、SBLON)

プログラム全体のシングルブロックマスク

SBLOF で指定したプログラムは、シングルブロック実行が有効な場合でも、1 つのブロックと同じように完全に実行されます。つまり、シングルブロック実行がプログラム全体でマスクされます。

SBLOF は PROC 行にあり、サブプログラムの終了まで、または中断されるまで有効です。リターン命令によって、サブプログラムの最後で停止するかどうかが決まります。

M17 による復帰ジャンプ:　サブプログラムの終了で停止します
RET による復帰ジャンプ:　サブプログラムの終了で停止しません

プログラム内のシングルブロックマスク

SBLOF だけが、そのままこのブロックに残ります。シングルブロックはこのブロックの後に、以下のタイミングまでは無効になります。

- 次の SBLON
  または
- 動作中のサブプログラムレベルの終了

構文

プログラム全体のシングルブロックマスク:

```
PROC ... SBLOF
```

プログラム内のシングルブロックマスク:

```
SBLOF
...
SBLON
```

意味

PROC:	プログラムの最初の命令
SBLOF:	シングルブロック実行を解除する命令 SBLOF を PROC ブロックに入れることも、このブロックを SBLOF 単独にすることもできます。
SBLON:	シングルブロック実行を起動する命令 SBLON は、個別のブロックに指令してください。

必要条件

- **シングルブロックマスクとブロック表示**
  DISPLOF を使用して、実行中のブロック表示をサイクル/サブプログラムでマスクできます。DISPLOF を SBLOF と一緒にプログラム指令した場合は、サイクル/サブプログラム内でのシングルブロック停止で、サイクル呼び出し/サブプログラム呼び出し側のプログラムの表示が続行されます。
- **システム ASUB、またはユーザー ASUB のシングルブロックマスク**
  システムまたはユーザー ASUB のシングルブロック停止を、マシンデータ MD10702 $MN_IGNORE_SINGLEBLOCK_MASK (bit0 = 1 または bit1 = 1)の設定内容を使用してマスクした場合は、ASUB で SBLON をプログラム指令することでシングルブロック停止を再度有効にできます。
  ユーザー ASUB でのシングルブロック停止を、マシンデータ MD20117 $MC_IGNORE_SINGLEBLOCK_ASUP での設定を使用してマスクした場合は、ASUB で SBLON をプログラム指令してシングルブロック停止を再度有効にすることは**できません**。
- **さまざまなシングルブロック実行タイプのシングルブロックマスクの特記事項**
  シングルブロック実行 SBL2 が有効な(各パートプログラムブロック後に停止する)ときに、MD10702 $MN_IGNORE_SINGLEBLOCK_MASK (シングルブロック停止をおこなわない)のビット 12 を「1」に設定している場合は、SBLON ブロックで実行は**停止されません**。
  シングルブロック実行 SBL3 が有効な(各パートプログラムブロック後、およびサイクルでも停止する)ときは、SBLOF 命令がマスクされます。

例

例 1:プログラム内のシングルブロックマスク

プログラムコード	コメント
N10 G1 X100 F1000	
N20 SBLOF	; シングルブロックの停止
N30 Y20	
N40 M100	
N50 R10=90	
N60 SBLON	; シングルブロックの再起動
N70 M110	
N80 ...	

N20 と N60 間の領域は、シングルブロックモードでは 1 ステップとして実行されます。

例 2 :ユーザー命令と同じように動作するサイクル

メインプログラム

プログラムコード
N10 G1 X10 G90 F200
N20 X-4 Y6
N30 CYCLE1
N40 G1 X0
N50 M30

サイクル CYCLE1:

プログラムコード	コメント
N100 PROC CYCLE1 DISPLOF SBLOF	; シングルブロックのマスク
N110 R10=3*SIN(R20)+5	
N120 IF (R11 <= 0)	
N130 SETAL(61000)	
N140 ENDIF	
N150 G1 G91 Z=R10 F=R11	
N160 M17	

CYCLE1 は、動作中のシングルブロックを実行する処理されます。つまり、1 回の起動キー押下で、CYCLE1 が処理されます。

例 3 :変更されたゼロオフセットと工具オフセットを有効にするために、PLC が起動した ASUB が、表示されずに実行されます。

プログラムコード
N100 PROC ZO SBLOF DISPLOF
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

プログラムコード
N200 END:D=$P_TOOL T=$P_TOOLNO
N210 RET

例 4 :MD10702 ビット 12 = 1 により停止しません

初期状態

- シングルブロック実行が有効です。
- MD10702 $MN_IGNORE_SINGLEBLOCK_MASK Bit12 = 1

メインプログラム

プログラムコード	コメント
N10 G0 X0	; このパートプログラム行で停止します。
N20 X10	; このパートプログラム行で停止します。
N30 CYCLE	; サイクルが生成した移動ブロックです。
N50 G90 X20	; このパートプログラム行で停止します。
M30	

サイクル CYCLE:

プログラムコード	コメント
PROC CYCLE SBLOF	; シングルブロック停止をマスクします。
N100 R0 = 1	
N110 SBLON	; 実際は MD10702 bit12=1 のため、このパートプログラム行で実行が停止しません。
N120 X1	; このパートプログラム行で実行が停止します。
N140 SBLOF	
N150 R0 = 2	
RET	

例 5 :入れ子のプログラムでのシングルブロックマスク

初期状態

シングルブロック実行が有効です。

入れ子のプログラム:

プログラムコード	コメント
N10 X0 F1000	; このブロックで実行が停止します。
N20 UP1(0)	
PROC UP1(INT _NR) SBLOF	; シングルブロック停止をマスクします。
N100 X10	
N110 UP2(0)	
PROC UP2(INT _NR)	
N200 X20	
N210 SBLON	; シングルブロック停止を適用します。
N220 X22	; このブロックで実行が停止します。
N230 UP3(0)	
PROC UP3(INT _NR)	
N300 SBLOF	; シングルブロック停止をマスクします。
N305 X30	
N310 SBLON	; シングルブロック停止を適用します。
N320 X32	; このブロックで実行が停止します。
N330 SBLOF	; シングルブロック停止をマスクします。
N340 X34	
N350 M17	; SBLOF が有効です。
N240 X24	; このブロックで実行が停止します。SBLON が有効です。
N250 M17	; このブロックで実行が停止します。SBLON が有効です。
N120 X12	
N130 M17	; この復帰ジャンプブロックで実行が停止します。PROC 命令の SBLOF が有効です。
N30 X0	; このブロックで実行が停止します。
N40 M30	; このブロックで実行が停止します。

詳細情報

非同期サブプログラムで無効なシングルブロック

ASUB を 1 ステップで実行するには、PROC 命令を SBLOF を含む ASUB でプログラム指令してください。これは、機能「編集可能なシステム ASUB」(MD11610 $MN_ASUP_EDITABLE)の場合にも該当します。

編集可能なシステム ASUB の例:

プログラムコード	コメント
N10 PROC ASUB1 SBLOF DISPLOF	
N20 IF $AC_ASUP=='H200'	
N30 RET	; モード変更では REPOS を実行しません。

プログラムコード	コメント
N40 ELSE	
N50 REPOSA	; その他の場合はすべて REPOS を実行します。
N60 ENDIF	

シングルブロックモードのプログラム制御

シングルブロック実行機能を使用すると、ユーザーはパートプログラムをブロック毎に実行できます。次の設定タイプがあります。

- SBL1:IPO シングルブロックは、各運転機能ブロックの後に停止します。
- SBL2:シングルブロックが、各ブロックの後に停止します。
- SBL3:サイクルで停止します(SBL3 の選択により SBLOF 命令がマスクされます)。

入れ子のプログラムでのシングルブロックマスク

SBLOF をサブプログラムの PROC 命令でプログラム指令した場合は、M17 のサブプログラム復帰ジャンプで実行が停止します。これにより、呼び出しプログラムの次のブロックが先に実行されないようにします。SBLOF をサブプログラムの PROC 命令と一緒にプログラム指令しても、しなくても、 シングルブロックのマスクが有効になり、呼び出しプログラムの次の運転機能ブロックの後にのみ、実行が停止します。この動作を望まない場合は、サブプログラムで戻りジャンプ(M17)の前に SBLON をプログラム指令してください。RET では、実行が停止して、より高レベルのプログラムへ復帰ジャンプすることはありません。

2.24.2.6 実行中のブロック表示のマスク(DISPLOF、DISPLON、ACTBLOCNO)

通常では、実行中のプログラムブロックがブロック表示に表示されます。DISPLOF 命令を使用すると、実行中のサイクルとサブプログラムのブロック表示をマスクできます。実行中のブロックではなく、サイクルまたはサブプログラムの呼び出しが表示されます。DISPLON 命令は、ブロック表示のマスクを無効にします。

DISPLOF と DISPLON は、PROC 命令でプログラム行にプログラム指令され、サブプログラム全体に対して有効です。また、このサブプログラムが呼び出した、DISPLON 命令も DISPLOF 命令も含まないすべてのサブプログラムにも自動的に有効になります。これは、すべての ASUB に当てはまります。

構文

```
PROC … DISPLOF
PROC … DISPLOF ACTBLOCNO
PROC … DISPLON
```

意味

DISPLOF:	実行中のブロック表示をマスクする命令です。	
	場所:	PROC 命令を含むプログラム行の末尾
	効果:	サブプログラムから、またはプログラムの終点からの復帰ジャンプまでです。
	注: サブプログラムから DISPLOF 命令を使用して、さらにサブプログラムを呼び出した場合は、実行中のブロック表示は、これらのサブプログラムでもマスクされます。ただし、これらのサブプログラムで DISPLON が明示的にプログラム指令されている場合はマスクされません。	
DISPLON:	実行中のブロック表示のマスクを無効にする命令です。	
	場所:	PROC 命令を含むプログラム行の末尾
	効果:	サブプログラムから、またはプログラムの終了からの復帰ジャンプまで。
	注: サブプログラムから DISPLON 命令で、さらにサブプログラムを呼び出した場合は、実行中のブロックが、これらのサブプログラムでも表示されます。ただし、これらのサブプログラムで DISPLOF が明示的にプログラム指令されている場合は表示されません。	
ACTBLOCNO:	DISPLOF を ACTBLOCNO 属性と併用すると、アラームが発生する場合に、アラームが発生した実際のブロック番号が出力されます。これは、より低いプログラムレベルで DISPLOF のみをプログラム指令した場合にも当てはまります。 また、DISPLOF を ACTBLOCNO と併用しない場合は、DISPLOF で指定されていない最後のプログラムレベルのサイクル呼び出しまたはサブプログラム呼び出しのブロック番号が表示されます。	

例

例 1:サイクルの実行中のブロック表示をマスク

プログラムコード	コメント
PROC CYCLE (AXIS TOMOV, REAL POSITION) SAVE DISPLOF	; 実行中のブロック表示をマスクし、代わりに、サイクル呼び出しを表示します。例:CYCLE(X,100.0)
DEF REAL DIFF	; サイクルの内容
G01 ...	

プログラムコード	コメント
...	
RET	; サブプログラム復帰ジャンプ。サイクル呼び出しの次のブロックがブロック表示に表示されます。

例 2 :アラーム発生ブロックの表示

サブプログラム SUBPROG1 (ACTBLOCNO を含む):

プログラムコード	コメント
PROC SUBPROG1 DISPLOF ACTBLOCNO	
N8000 R10 = R33 + R44	
...	
N9040 R10 = 66 X100	; アラーム 12080 のトリガ
...	
N10000 M17	

サブプログラム SUBPROG2 (ACTBLOCNO を含まない):

プログラムコード	コメント
PROC SUBPROG2 DISPLOF	
N5000 R10 = R33 + R44	
...	
N6040 R10 = 66 X100	; アラーム 12080 のトリガ
...	
N7000 M17	

メインプログラム

プログラムコード	コメント
N1000 G0 X0 Y0 Z0	
N1010 ...	
...	
N2050 SUBPROG1	; アラーム出力=「12080 チャネル K1 ブロック N9040 テキスト R10=の構文エラー」
N2060 ...	
N2350 SUBPROG2	; アラーム出力=「12080 チャネル K1 ブロック N2350 テキスト R10=の構文エラー」
...	
N3000 M30	

例 3 :実行中のブロック表示のマスクは無効です

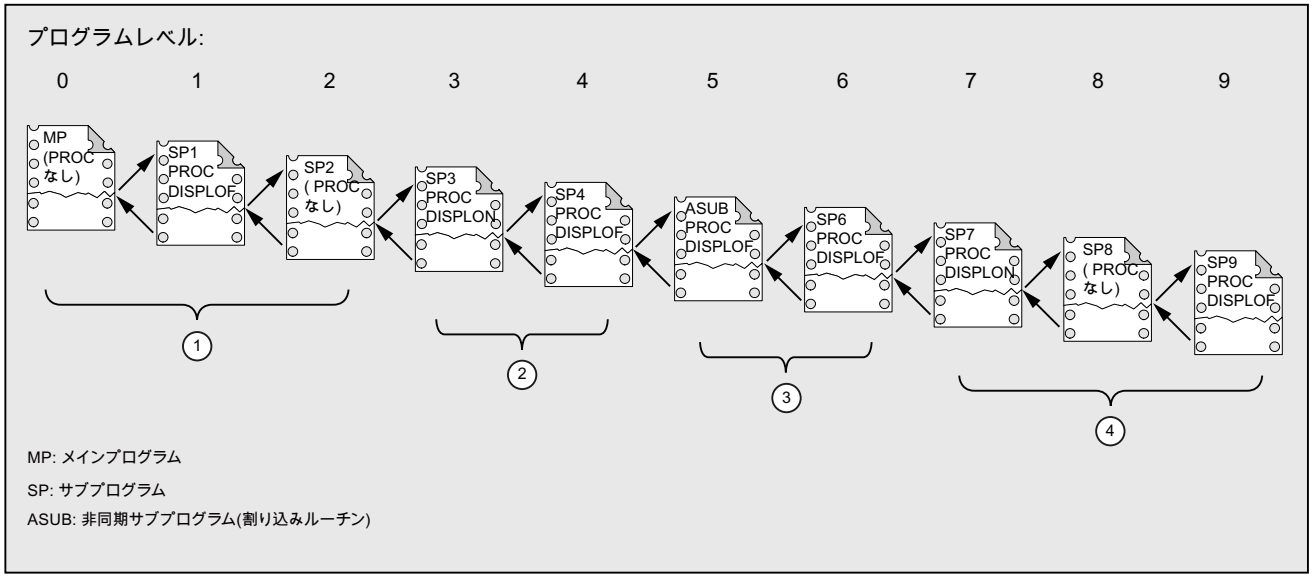
マスクを含むサブプログラム SUB1:

プログラムコード	コメント
PROC SUB1 DISPLOF	; サブプログラム SUB1 で実行中のブロック表示をマスクします。代わりに、ブロックが SUB1 呼び出しで表示されます。
...	
N300 SUB2	; サブプログラム SUB2 を呼び出します。
...	
N500 M17	

マスクを含まないサブプログラム SUB2:

プログラムコード	コメント
PROC SUB2 DISPLON	; サブプログラム SUB2 で実行中のブロック表示のマスクを無効にします。
...	
N200 M17	; サブプログラム SUB1 に戻ります。実行中のブロック表示のマスクを SUB1 で復帰します。

例 4 :DISPLON/DISPLOF の組み合わせが異なる場合の表示動作



① プログラムレベル 0 のパートプログラム行が、実行中のブロック表示に表示されます。

② プログラムレベル 3 のパートプログラム行が、実行中のブロック表示に表示されます。

③ プログラムレベル 3 のパートプログラム行が、実行中のブロック表示に表示されます。

④ プログラムレベル 7/8 のパートプログラム行が、実行中のブロック表示に表示されます。

2.24.2.7 解析によるサブプログラムの識別(PREPRO)

すべてのファイルは電源投入時に、PROC 命令行の末尾の PREPRO キーワードで識別されます。

注記

このプログラム解析のタイプは、当該のマシンデータの設定により異なります。工作機械メーカの説明書に従ってください。

参照先:
機能マニュアル 応用機能; 先読み処理(V2)

構文

```
PROC … PREPRO
```

意味

PREPRO:	電源投入時に解析される(サイクルディレクトリに格納された NC プログラムの)全ファイル識別用キーワードです。

解析されたサブプログラムの読み取り、およびサブプログラムの呼び出し

サイクルディレクトリは、パラメータにより、サブプログラムが電源投入時に先読みされる場合も、サブプログラム呼び出しのときに先読みされる場合も、同じ順序で処理されます。

1. _N_CUS_DIR ユーザーのサイクル
2. _N_CMA_DIR メーカサイクル
3. _N_CST_DIR 標準サイクル

同名称でも内容が異なる NC プログラムの場合は、最初に見つかった PROC 命令が有効になり、それ以外の PROC 命令はアラームメッセージなしで無視されます。

2.24.2.8 サブプログラム戻り M17

復帰命令 M17 (またはパートプログラム終了命令 M30)は、サブプログラムの末尾にあります。この命令は、サブプログラム呼び出しの次のパートプログラムブロックの位置で、呼び出しプログラムに戻るよう指示します。

注記

M17 と M30 は、NC 言語では同等に扱われます。

構文

```
PROC <プログラム名称>
...
M17/M30
```

必要条件

連続軌跡モードのサブプログラム戻りの動作

M17 (または M30)が単独でパートプログラムブロックに含まれる場合は、チャネルで動作中の連続軌跡モードが中断されます。

連続軌跡モードの中断を回避するには、M17 (または M30)を最後の移動ブロックに含めてください。さらに、次のマシンデータを「0」に設定してください。

MD20800 $MC_SPF_END_TO_VDI = 0 (NC/PLC インタフェースへの M30/M17 出力なし)

例

1.個別のブロックの M17 のサブプログラム

プログラムコード	コメント
N10 G64 F2000 G91 X10 Y10	
N20 X10 Z10	
N30 M17	; 連続軌跡モードを中断して復帰ジャンプします。

2.最後の移動ブロックに M17 を含むサブプログラム

プログラムコード	コメント
N10 G64 F2000 G91 X10 Y10	

プログラムコード	コメント
N20 X10 Z10 M17	; 連続軌跡モードを中断しない復帰ジャンプです。

2.24.2.9 RET サブプログラム戻り

RET 命令も、M17 復帰ジャンプ命令の代わりとしてサブプログラムで使用できます。RET は、個別のパートプログラムブロックでプログラム指令してください。M17 と同様に、RET は、サブプログラム呼び出しの次のパートプログラムブロックで呼び出しプログラムに戻るよう指示します。

注記

RET の復帰ジャンプ動作は、パラメータをプログラム指令して変更することができます(「設定可能なサブプログラム復帰ジャンプ(RET ...) (ページ 210)」を参照してください)。

用途

G64 連続軌跡モード(G641 ～ G645)が復帰ジャンプで中断されないようにする場合は、RET 命令を使用してください。

必要条件

RET 命令は、SAVE 属性で定義されていないサブプログラムでのみ使用できます。

構文

```
PROC <プログラム名称>
...
RET
```

例

メインプログラム

プログラムコード	コメント
PROC MAIN_PROGRAM	; プログラムの開始
...	
N50 SUB_PROG	; サブプログラム呼び出し:SUB_PROG
N60 ...	
...	

プログラムコード	コメント
N100 M30	; プログラム終了

サブプログラム:

プログラムコード	コメント
PROC SUB_PROG	
...	
N100 RET	; メインプログラムの N60 ブロックへ復帰ジャンプ

2.24.2.10 設定可能なサブプログラム復帰ジャンプ(RET ...)

通常、復帰ジャンプは、RET 命令を使用してサブプログラムから呼び出しプログラムに戻ります。そして、サブプログラム呼び出しの次のプロゴラム行から処理が続行されます。ただし、次の例のように、プログラムの処理を別の位置から続行することもできます。

- (輪郭の記述後に)ISO 系言語モードの切削サイクルの呼び出し後に、プログラムの実行を再開します。
- エラー処理のために、(ASUB 後でも)任意のサブプログラムレベルからメインプログラムに戻ります。
- コンパイルサイクルと ISO 系言語モードで、特別な用途のために複数のプログラムレベルをジャンプして復帰します。

これを達成するには、RET 命令を追加のパラメータを付けてプログラム指令します。

検索方向

パラメータ<ターゲットブロック>を指定した場合、最初に、呼び出しブロックの後のブロックに復帰ジャンプします。次に、復帰ジャンプ先のプログラムの**末尾**方向にターゲットの検索がおこなわれます。見つからなかった場合は、プログラムの先頭方向に検索がおこなわれます。

構文

```
RET("<ターゲットブロック>")
RET("<ターゲットブロック>",<ターゲットブロックの後のブロック>)
RET("<ターゲットブロック>",<ターゲットブロックの後のブロック> <復帰ジャンプレベルの数>)
RET("<ターゲットブロック>", ,<復帰ジャンプレベルの数>)
RET("<ターゲットブロック>",<ターゲットブロックの後のブロック>,<復帰ジャンプレベルの数>,
<プログラムの先頭への復帰ジャンプ>)
RET( , ,復帰ジャンプレベルの数>,<プログラムの先頭への復帰ジャンプ>)
```

意味

RET:	サブプログラム終了	
<ターゲットブロック>:	ブロックを、プログラムの実行を再開するジャンプ先として宣言します。 パラメータ< 復帰ジャンプレベルの数>をプログラム指令していない場合は、ジャンプ先は実行中のサブプログラムを呼び出したプログラムになります。 指定できるデータには以下のものがあります。	
	<ブロック番号>	ターゲットブロックの番号です。 ブロック番号の検索は、復帰ジャンプ先プログラムで、最初にプログラムの末尾方向に実行されます。
	<ジャンプマーク>	復帰ジャンプ先プログラムに存在するジャンプマークにしてください。 ジャンプマークの検索は、復帰ジャンプ先プログラムで、最初にプログラムの末尾方向に実行されます。
	<文字列>	復帰ジャンプ先プログラムに存在する文字列にしてください(プログラム名称や変数名称など)。 文字列の検索は、復帰ジャンプ先プログラムで、最初にプログラムの末尾方向に実行されます。 文字列をプログラム指令するときは、以下の規則が適用されます。 ● **末尾にブランクを付けます** (末尾に「:」を付けるジャンプマークとは異なります)。 ● **文字列の前には、**1 つのブロック番号やジャンプマークのいずれか、または両方のみを設定できます。**プログラム命令は設定できません。**

<ターゲットブロックの後のブロック>:	プログラム処理を、パラメータ<ターゲットブロック>で指定したブロックから続行するか、その次のブロックから続行するかを指定します。		
	タイプ:	INT	
	値:	0	復帰ジャンプのジャンプ先は、パラメータ<ターゲットブロック>で指定したブロックです。
		> 0	復帰ジャンプのジャンプ先は、パラメータ<ターゲットブロック>で指定したブロックの次のブロックです。
<復帰ジャンプレベルの数>:	ターゲットブロックを検索してプログラムの処理を続行するために、ジャンプして戻るべき(復帰ジャンプ)プログラムレベルの数を指定します。		
	タイプ:	INT	
	値:	1	プログラムが「現在のプログラムレベル - 1」で再開されます(パラメータなしの RET と同様)。
		2	プログラムが「現在のプログラムレベル - 2」で再開されます。つまり、レベルを 1 つスキップします。
		3	プログラムが「現在のプログラムレベル - 3」で再開されます。つまり、レベルを 2 つスキップします。
		...	
	データ範囲:	1 ... 15	
<プログラムの先頭への復帰ジャンプ>:	メインプログラムへの復帰ジャンプの場合、有効な **ISO 系言語モード**でプログラムをプログラムの先頭から続行するかどうかを指定します。		
	タイプ:	BOOL	
	値:	1	復帰ジャンプのジャンプ先がメインプログラムの中にあり、そこで **ISO 系言語モード**が有効である場合は、プログラムはプログラムの先頭に分岐します。

注記

ターゲットブロック検索を指定する文字列を含むサブプログラム復帰ジャンプの場合は、最初に、呼び出しプログラムのジャンプマークの検索が必ず実行されます。

文字列を使用してジャンプ先を一義的に定義する場合は、ジャンプマーク名称と一致する文字列は使用できません。これは、一致する文字列を使用すると、サブプログラム復帰ジャンプのジャンプ先が常に文字列でなく、ジャンプマークとなるためです(例 2 を参照してください)。

必要条件

複数のプログラムレベルにわたる復帰ジャンプをおこなうときは、個々のプログラムレベルの SAVE 命令が使用されます。

複数のプログラムレベルの復帰ジャンプのときに、モーダルサブプログラムが有効な場合、およびスキップするプログラムのいずれかで選択解除命令 MCALL をモーダルサブプログラムに対してプログラム指令している場合は、モーダルサブプログラムがそのまま有効になります。

通知
プログラム指令エラー 複数のプログラムレベルでの復帰ジャンプの場合、処理が必要なモーダル設定で続行されることを必ず確認してください。これは、適切なメインブロックをプログラム指令するなどの方法でおこなうことができます。

例

例 1:ASUB 実行後のメインプログラムの再開

```
プログラミング                          コメント
N10010 CALL "UP1"                      プログラムレベル 0 (メインプログラム)
  N11000 PROC UP1                      ; プログラムレベル 1
  N11010 CALL "UP2"
  N12000 PROC UP2                      ; プログラムレベル 2
  ...
  N19000 PROC ASUP                     ; プログラムレベル 3 (ASUB の実行)
  ...
  N19100 RET("N10900", ,$P_STACK)      ; メインプログラムへサブプログラム復帰ジャンプ
                                       ; $P_STACK:実際のプログラムレベル
N10900                                 ; メインプログラムのターゲットブロック
```

プログラミング	コメント
N10910 MCALL	; モーダルサブプログラム呼び出しを解除します
N10920 G0 G60 G40 M5	; 追加モーダル設定を初期化します

例 2 :ターゲットブロック検索を指定する文字列(string>)

メインプログラム

プログラムコード	コメント
PROC MAIN_PROGRAM	
N1000 DEF INT iVar1=1, iVar2=4	
N1010 ...	
N1200 subProg1	; サブプログラム「subProg1」を呼び出します。
N1210 M2 S1000 X10 F1000	
N1220	
N1400 subProg2	; サブプログラム「subProg2」を呼び出します。
N1410 M3 S500 Y20	
N1420 ..	
N1500 lab1:iVar1=R10*44	
N1510 F500 X5	
N1520 ...	
N1550 subprog1:G1 X30	; ここで、「subProg1」がジャンプマークとして定義されます。
N1560 ...	
N1600 subProg3	; サブプログラム「subProg3」を呼び出します。
N1610 ...	
N1900 M30	

サブプログラム subProg1:

プログラムコード	コメント
PROC subProg1	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg2")	; メインプログラムの N1400 ブロックへ復帰ジャンプ

サブプログラム subProg2:

プログラムコード	コメント
PROC subProg2	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("iVar1")	; メインプログラムの N1500 ブロックへ復帰ジャンプ

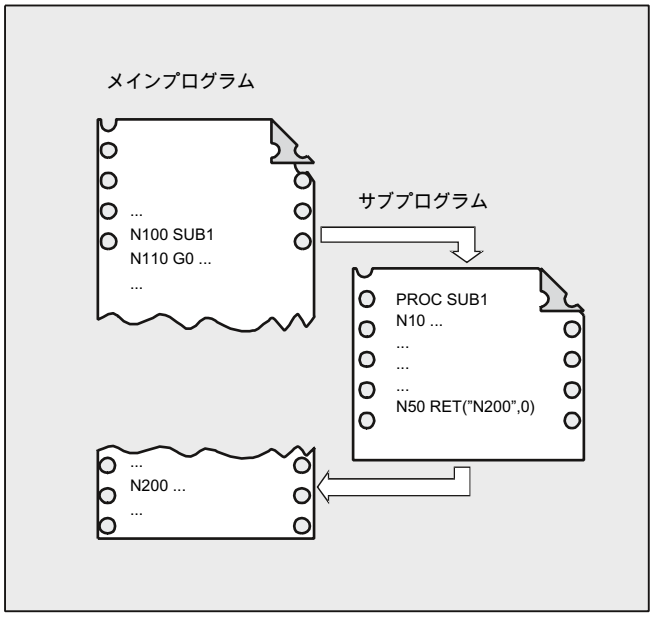
サブプログラム subProg3:

プログラムコード	コメント
PROC subProg3	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg1")	; メインプログラムの N1550 ブロックへ復帰ジャンプ

関連情報

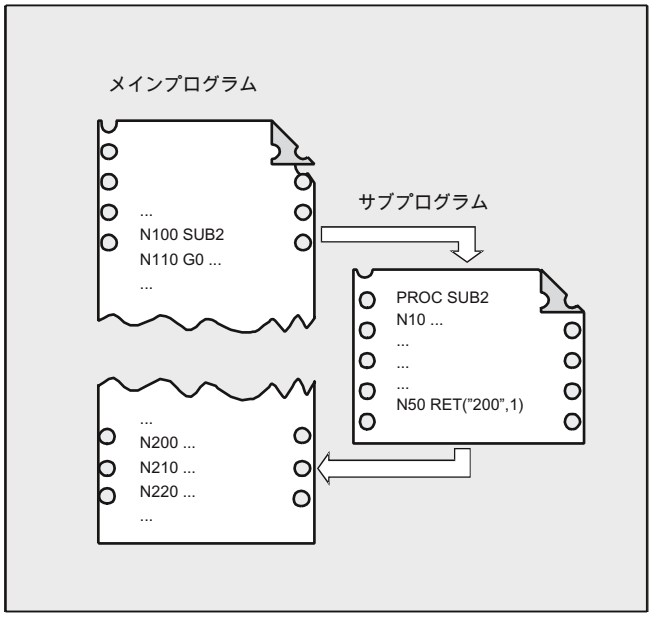
次の図は、復帰ジャンプパラメータのさまざまな働きを示しています。

1. <ターゲットブロック> = "N200"、<ターゲットブロックの後のブロック> = 0



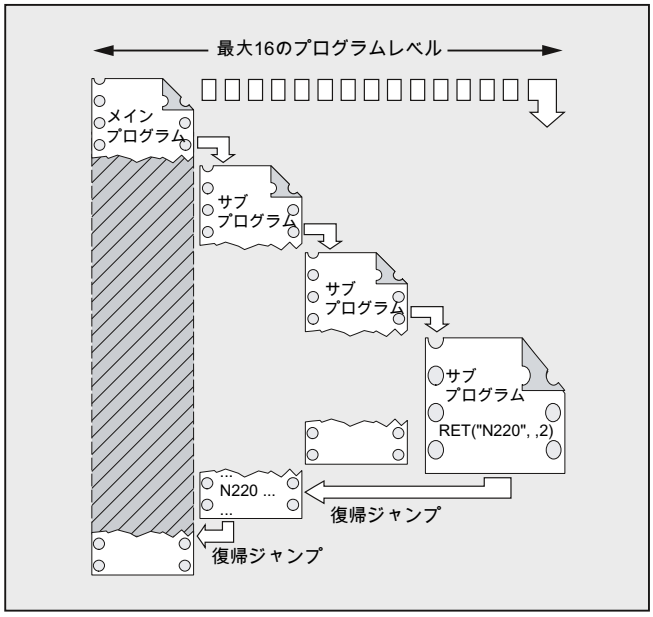
RET 命令の後に、プログラムの実行をメインプログラムの N200 ブロックで続行します。

2. <ターゲットブロック> = "N200"、<ターゲットブロックの後のブロック> = 1



RET 命令の後に、プログラムの実行を、メインプログラムの N200 ブロックの次の(N210)ブロックで続行します。

3. <ターゲットブロック> = "N220"、<復帰ジャンプレベルの数> = 2



RET 命令の後に、2 つのプログラムレベルをジャンプし、プログラムの実行を N220 ブロックで続行します。

2.24.2.11 パラメータ設定可能なサブプログラム復帰ジャンプ(RETB ...)

通常、復帰ジャンプは、RETB 命令を使用してサブプログラムから呼び出しプログラムに戻ります。そして、サブプログラム呼び出しの次のプロゴラム行から処理が続行されます。ただし、次の例のように、プログラムの処理を別の位置から続行することもできます。

- (輪郭の記述後に)ISO 系言語モードの旋削サイクルの呼び出し後に、プログラムの実行を再開します。
- エラー処理のために、(ASUB 後でも)任意のサブプログラムレベルからメインプログラムに戻ります。
- コンパイルサイクルと ISO 系言語モードで、特別な用途のために複数のプログラムレベルをジャンプして復帰します。

これを達成するには、RETB 命令を追加のパラメータを付けてプログラム指令します。

検索方向

パラメータ<ターゲットブロック>を指定した場合、最初に、呼び出しブロックの後のブロックに復帰ジャンプします。次に、復帰ジャンプ先のプログラムの**先頭**方向にターゲットの検索がおこなわれます。見つからなかった場合は、プログラムの末尾方向に検索がおこなわれます。

構文

```
RETB("<ターゲットブロック>")
RETB("<ターゲットブロック>",<ターゲットブロックの後のブロック>)
RETB("<ターゲットブロック>",<ターゲットブロックの後のブロック> <復帰ジャンプレベルの数>)
RETB("<ターゲットブロック>", ,<復帰ジャンプレベルの数>)
RETB("<ターゲットブロック>",<ターゲットブロックの後のブロック>,<復帰ジャンプレベルの数>,
<プログラムの先頭への復帰ジャンプ>)
RETB( , ,<復帰ジャンプレベルの数>,<プログラムの先頭への復帰ジャンプ>)
```

意味

RETB:	サブプログラム終了	
<ターゲットブロック>:	ブロックを、プログラムの実行を再開するジャンプ先として宣言します。 パラメータ< 復帰ジャンプレベルの数>をプログラム指令していない場合は、ジャンプ先は実行中のサブプログラムを呼び出したプログラムになります。 指定できるデータには以下のものがあります。	
	<ブロック番号>	ターゲットブロックの番号です。 ブロック番号の検索は、復帰ジャンプ先プログラムで、最初にプログラムの先頭方向に実行されます。
	<ジャンプマーク>	復帰ジャンプ先プログラムに存在するジャンプマークにしてください。 ジャンプマークの検索は、復帰ジャンプ先プログラムで、最初にプログラムの先頭方向に実行されます。
	<文字列>	復帰ジャンプ先プログラムに存在する文字列にしてください(プログラム名称や変数名称など)。 文字列の検索は、復帰ジャンプ先プログラムで、最初にプログラムの先頭方向に実行されます。 文字列をプログラム指令するときは、以下の規則が適用されます。 ● **末尾にブランクを付けます** (末尾に「:」を付けるジャンプマークとは異なります)。 ● **文字列の前には、**1 つのブロック番号やジャンプマークのいずれか、または両方のみを設定できます。**プログラム命令は設定できません**。

<ターゲットブロックの後のブロック>:	プログラム処理を、パラメータ<ターゲットブロック>で指定したブロックから続行するか、その次のブロックから続行するかを指定します。		
	タイプ:	INT	
	値:	0	復帰ジャンプのジャンプ先は、パラメータ<ターゲットブロック>で指定したブロックです。
		> 0	復帰ジャンプのジャンプ先は、パラメータ<ターゲットブロック>で指定したブロックの次のブロックです。
<復帰ジャンプレベルの数>:	ターゲットブロックを検索してプログラムの処理を続行するために、ジャンプして戻るべき(復帰ジャンプ)プログラムレベルの数を指定します。		
	タイプ:	INT	
	値:	1	プログラムが「現在のプログラムレベル- 1」で再開されます(パラメータなしの RET と同様)。
		2	プログラムが「現在のプログラムレベル- 2」で再開されます。つまり、レベルを 1 つスキップします。
		3	プログラムが「現在のプログラムレベル- 3」で再開されます。つまり、レベルを 2 つスキップします。
		...	
	データ範囲:	1 ... 15	
<プログラムへの先頭への復帰ジャンプ>:	メインプログラムへの復帰ジャンプの場合、有効な **ISO 系言語モード**でプログラムをプログラムの先頭から続行するかどうかを指定します。		
	タイプ:	BOOL	
	値:	1	復帰ジャンプのジャンプ先がメインプログラムの中にあり、そこで **ISO 系言語モード**が有効である場合は、プログラムはプログラムの先頭に分岐します。

注記

ターゲットブロック検索を指定する文字列を含むサブプログラム復帰ジャンプの場合は、最初に、呼び出しプログラムのジャンプマークの検索が必ず実行されます。

文字列を使用してジャンプ先を一義的に定義する場合は、ジャンプマーク名称と一致する文字列は使用できません。これは、一致する文字列を使用すると、サブプログラム復帰ジャンプのジャンプ先が常に文字列でなく、ジャンプマークとなるためです(例 2 を参照してください)。

必要条件

複数のプログラムレベルにわたる復帰ジャンプをおこなうときは、個々のプログラムレベルの SAVE 命令が使用されます。

複数のプログラムレベルの復帰ジャンプのときに、モーダルサブプログラムが有効な場合、およびスキップするプログラムのいずれかで選択解除命令 MCALL をモーダルサブプログラムに対してプログラム指令している場合は、モーダルサブプログラムがそのまま有効になります。

通知

プログラム指令エラー

複数のプログラムレベルでの復帰ジャンプの場合、処理が必要なモーダル設定で続行されることを必ず確認してください。これは、適切なメインブロックをプログラム指令するなどの方法でおこなうことができます。

例

プログラムコード	コメント

```
EXAMPLE.MPF
     …
   N3000 START_CYC(パラメータ 1, パラメータ 2,
…)
   N3010 TECH_CYC1(パラメータ, パラメータ 2,
…)
   N3020 TECH_CYC2(パラメータ, パラメータ 2,
…)
   N3030 TECH_CYC3(パラメータ, パラメータ 2,
…)
   N3040 END_CYC(パラメータ, パラメータ 2, …)
   N3040 END_CYC(パラメータ, パラメータ 2, …)
   N3050 …
```

プログラムコード	コメント
N4500 START_CYC(パラメータ 11, パラメータ 12, …)	
N4510 …	
N4590 END_CYC(パラメータ 11, パラメータ 12, ..)	
N5000 …	
…	
N6000 M30	

プログラムコード	コメント
PROC END_CYC(…)	; メインプログラム、N3040 行で呼び出し
N10000 …	
N15000 if status == 1	
N15010 RETB("START_CYC")	; 呼び出しプログラム EXAMPLE.MPF へ復帰ジャンプ
	; 文字列「START_CYC」を検索
	; 検索方向:プログラムの先頭方向へ
	; プログラム処理を N3000 行から続行します
N15020 ENDIF	
N15030 if status == 0	
N15040 RET	; 呼び出しプログラム EXAMPLE.MPF へ復帰ジャンプ
	; プログラム処理を N3050 行から続行します
N15050 ENDIF	
N16000 RET("START_CYC")	; 呼び出しプログラム EXAMPLE.MPF へ復帰ジャンプ
	; 文字列「START_CYC」を検索
	; 検索方向:プログラムの末尾方向へ
	; プログラム処理を N4500 行から続行します
N17060 RETB	; 呼び出しプログラム EXAMPLE.MPF へ復帰ジャンプ
	; プログラム処理を N3050 行から続行します
	; パラメータなしの RETB は RET と同じです

2.24.3 サブプログラムの呼び出し

2.24.3.1 パラメータ転送を含まないサブプログラム呼び出し

サブプログラムは、アドレス L とサブプログラム番号で、またはプログラム名称を指定して呼び出します。

メインプログラムもサブプログラムとして呼び出すことができます。この場合は、メインプログラムに設定したプログラムの終点 M2 または M30 を M17(呼び出しプログラムへの戻りがあるプログラムの終了)として使用します。

注記

したがって、サブプログラムをメインプログラムとして起動することもできます。

制御装置での検索方法:

*_MPF があるかどうか

*_SPF があるかどうか

つまり、呼び出すサブプログラムの名称がメインプログラムの名称と同じである場合は、呼び出しを発行したメインプログラムが再度呼び出されます。これは、一般的には不必要な動作のため、サブプログラムとメインプログラムに固有の名称を割り当てて回避してください。

注記

パラメータ転送が不要なサブプログラムは、初期化ファイルから呼び出すこともできます。

構文

L<番号>/<プログラム名称>

注記

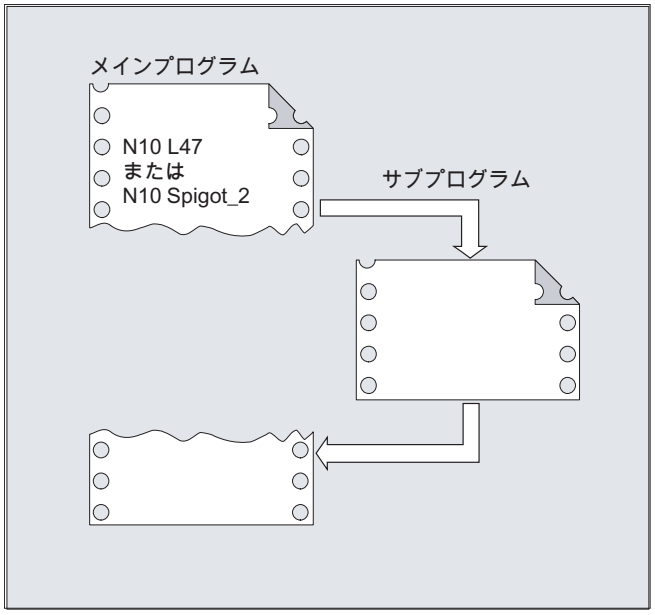
サブプログラム呼び出しは常に、個別の NC ブロックでプログラム指令してください。

意味

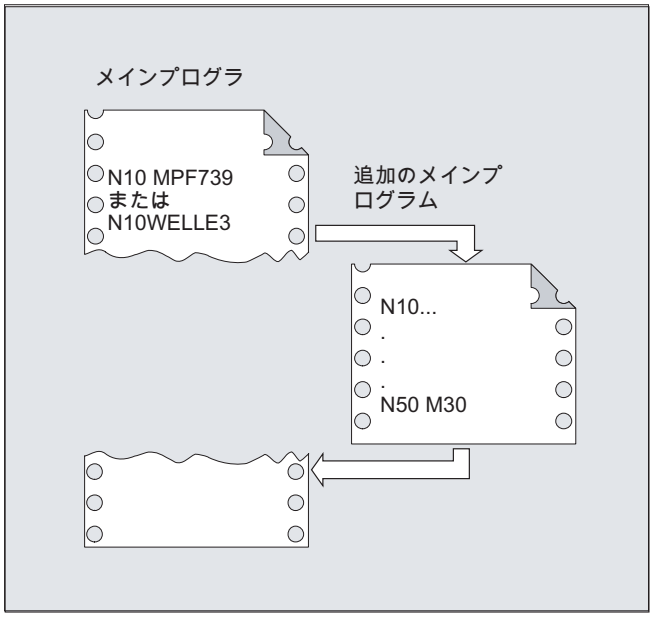
L:	サブプログラム呼び出しのアドレス	
<番号>:	サブプログラムの名称	
	タイプ:	INT
	値:	7 桁までの数値 **重要:** 名称は、先頭のゼロに意味があります(⇒ L123、L0123、および L00123 は 3 個の異なるサブプログラムです)。
<プログラム名称>:	サブプログラム(またはメインプログラム)の名称	

例

例 1:パラメータ転送を含まないサブプログラム呼び出し



例 2 :メインプログラムをサブプログラムとして呼び出し



下記も参照

パラメータ転送を含まないサブプログラム (ページ 190)

2.24.3.2 パラメータ転送を含むサブプログラム呼び出し(EXTERN)

パラメータ転送を含むサブプログラム呼び出しの場合は、変数または値を直接転送できます(ただし、VAR パラメータを除きます)。

パラメータ転送を含むサブプログラムは、メインプログラムのなかでで呼び出す前に、メインプログラム(プログラムの先頭など) に EXTERNAL で宣言してください。したがって、サブプログラムの名称と変数タイプは、転送される順に指定します。

通知
混乱の可能性 変数タイプとその転送順序は、サブプログラムの PROC で宣言した定義に一致させてください。メインプログラムとサブプログラムで、異なるパラメータ名称を使用できます。

構文

```
EXTERNAL <プログラム名称>(<タイプ_パラメータ 1>,<タイプ_パラメータ 2>,<タイプ_パラメータ 3>)
...
<プログラム名称>(<値_パラメータ 1>,<値_パラメータ 2>,<値_パラメータ 3>)
```

注記

サブプログラム呼び出しは常に、個別の NC ブロックでプログラム指令してください。

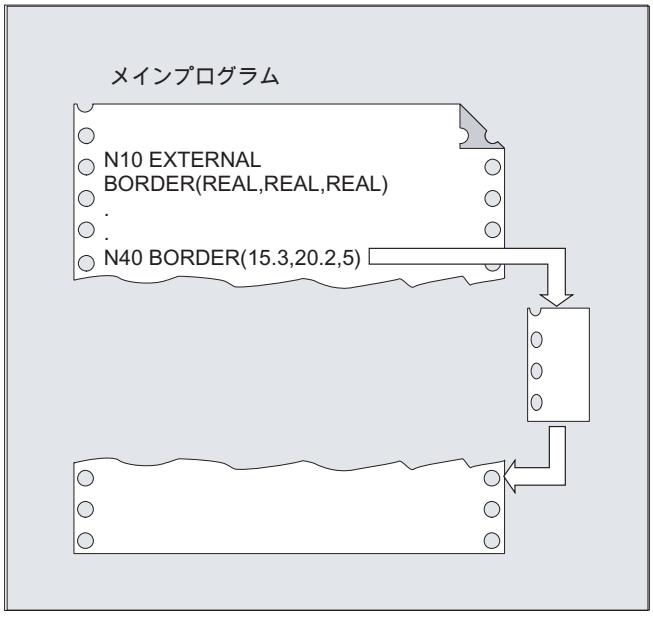
意味

<プログラム名称>:	サブプログラムの名称です。
EXTERNAL:	パラメータ転送を含むサブプログラムを宣言するキーワードです。 **注:** EXTERNAL は、サブプログラムがワーク、またはグローバルサブプログラムディレクトリにある場合にのみ指定してください。 サイクルは、EXTERNAL として宣言する必要はありません。
<タイプ_パラメータ 1>,<タイプ_パラメータ 2>,<タイプ_パラメータ 3>:	転送順に転送されるパラメータの変数タイプです。
<値_パラメータ 1>,<値_パラメータ 2>,<値_パラメータ 3>):	転送されるパラメータの変数値です。

例

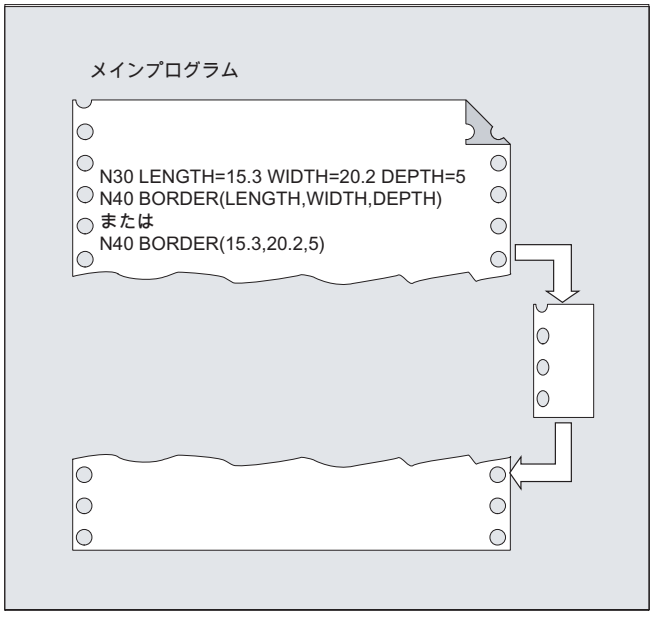
例 1:宣言があるサブプログラム呼び出し

プログラムコード	コメント
N10 EXTERNAL BORDERS(REAL,REAL,REAL)	; サブプログラムを指定します。
...	
N40 BORDER(15.3,20.2,5)	; パラメータ転送を含むサブプログラムを呼び出します。



例 2 :宣言のないサブプログラム呼び出し

プログラムコード	コメント
N10 DEF REAL LENGTH, WIDTH, DEPTH	
N20…	
N30 LENGTH=15.3 WIDTH=20.2 DEPTH=5	
N40 BORDER(LENGTH,WIDTH,DEPTH)	; または:N40 BORDER(15.3,20.2,5)



下記も参照

値渡しパラメータ転送を含むサブプログラム(PROC) (ページ 191)

参照渡しパラメータ転送を含むサブプログラム(PROC、VAR) (ページ 194)

2.24.3.3 プログラム繰り返し回数(P)

サブプログラムを複数回連続して実行する場合は、サブプログラム呼び出しと一緒にプログラム繰り返し回数を、このブロックのアドレス P へ入力できます。

⚠ 注意

プログラム繰り返しとパラメータ転送を含むサブプログラム呼び出し

パラメータは、プログラムが呼び出されたとき、つまり、1 回目の実行時にのみ転送されます。その後のプログラム繰り返しでは、同じパラメータがそのまま有効です。プログラム繰り返し中にパラメータを変更する場合は、サブプログラムのなかに、変更の準備をしてください。

構文

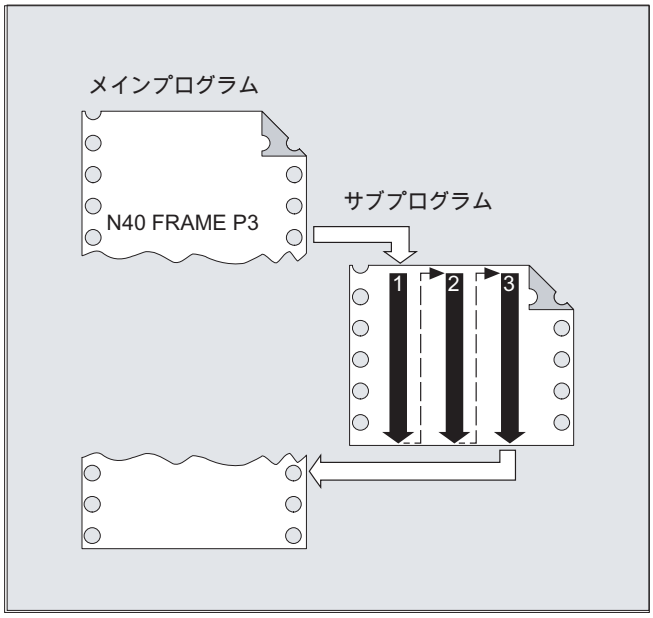
<プログラム名称> P<値>

意味

<プログラム名称>:	サブプログラム呼び出し	
P:	プログラム繰り返しを設定するアドレスです。	
<値>:	プログラム繰り返し回数です。	
	タイプ:	INT
	値の範囲:	1 … 9999 (符号なし)

例

プログラムコード	コメント
...	
N40 FRAME P3	; FRAME のサブプログラムが 3 回、連続して実行されます。
...	



2.24.3.4 モーダルサブプログラム呼び出し(MCALL)

モーダルサブプログラム呼び出し MCALL (<プログラム名称>) では、指定したサブプログラムはすぐには呼び出されません。そうではなく、軌跡移動を含む各移動ブロック後にパートプログラムで呼び出しが実行されます。プログラムレベル全体で実行することもできます。

注記

プログラムが実行される場合、最後のモーダルサブプログラム呼び出し MCALL (<プログラム名称>) のみが有効になります(常にこれが適用されます)。最新のモーダルサブプログラム呼び出しが、それまでに有効であった呼び出しと置き換わります。

パラメータをサブプログラムに転送する場合、パラメータは呼び出し MCALL (<プログラム名称> (Par1, Par2, ...)) でのみ転送されます。

通知

軌跡移動なしでのモーダルサブプログラム呼び出し

以下の事例では、モーダルサブプログラムも、軌跡移動のプログラム指令なしで呼び出されます。

- G0 または G1 が有効な場合に、アドレス S または F をプログラム指令した場合。
- G0 または G1 を、このブロックに単独で、または別の G 命令と一緒にプログラムした場合。

構文

```
MCALL <プログラム名称>
...
MCALL
```

意味

MCALL <プログラム名称>:	「モーダルサブプログラム呼び出し」機能を有効にします。
<プログラム名称>:	サブプログラムの名称です。
MCALL:	「モーダルサブプログラム呼び出し」機能を解除するには、プログラム名称の指定を含まない MCALL を使用します。

必要条件

ASUB

パートプログラムの処理が ASUB によって中断される場合(「割り込みルーチン(ASUB) (ページ 147)」の章を参照してください)、 この ASUB でモーダルサブプログラム呼び出しは実行されません。

ただし、ASUB が「リセット」チャネル状態で開始された場合は、モーダルサブプログラム呼び出しでの動作は、通常のパートプログラムと同じになります。

工具交換サイクル

「モーダルサブプログラム呼び出し」機能が工具交換サイクル中に選択解除された場合、検索 ASUB により、ブロック検索後であっても自動的に、またはオーバストアにより手動で、工具交換サイクルが呼び出されることに注意してください。この場合、「モーダルサブプログラム呼び出し」機能は選択解除しないでください。選択解除すると、正しい検索結果が得られません。そのため、工具交換サイクルでの「モーダルサブプログラム呼び出し」機能の選択解除は、以下のようにプログラム指令することをお勧めします。

プログラムコード	コメント
...	
IF $AC_ASUP == 0	; 呼び出しは検索 ASUB またはオーバストアにより実行されません。
MCALL	; 「モーダルサブプログラム呼び出し」機能を解除します。
ENDIF	
...	

例

例 1

プログラムコード	コメント
N10 G0 X0 Y0	
N20 MCALL L70	; L70 のモーダルサブプログラム呼び出しを有効にします。
N30 X10 Y10	; X10 Y10 へのアプローチ後に、L70 が呼び出されます。
N40 X20 Y20	; X20 Y20 へのアプローチ後に、L70 が呼び出されます。
...	
N100 MCALL	; 「モーダルサブプログラム呼び出し」機能を解除します。
N110 X0 Y0	; X0 Y0 にアプローチし、L70 は呼び出され**ません**。

例 2

プログラムコード
N10 G0 X0 Y0
N20 MCALL L70
N30 L80

この例では、プログラム指令された軌跡軸がある次の NC ブロックがサブプログラム L80 にあります。L70 は L80 で呼び出されます。

2.24.3.5 間接サブプログラム呼び出し(CALL)

プログラムの特定の位置での主な条件によっては、異なる複数のサブプログラムの呼び出しができます。サブプログラムの名称は、STRING タイプの変数に保存されます。サブプログラム呼び出しは、CALL と変数名称を使用して実行されます。

注記

間接サブプログラム呼び出しは、パラメータ転送を含まないサブプログラムに対してのみ使用できます。直接サブプログラム呼び出しをおこなうためには、STRING 定数に名称を保存してください。

構文

```
CALL <プログラム名称>
```

意味

CALL:	間接サブプログラム呼び出しの命令です。	
<プログラム名称>:	サブプログラムの名称(変数または定数)です。	
	タイプ:	STRING

例

STRING 定数による直接呼び出し:

プログラムコード	コメント
…	
CALL "/_N_WKS_DIR/_N_SUBPROG_WPD/_N_PART1_SPF"	; CALL によりサブプログラム PART1 を直接呼び出します。
…	

変数による間接呼び出し:

プログラムコード	コメント
…	
DEF STRING[100] PROGNAME	:変数を定義します。
PROGNAME="/_N_WKS_DIR/_N_SUBPROG_WPD/_N_PART1_SPF"	; サブプログラム PART1 を PROGNAME 変数に割り当てます。
CALL PROGNAME	; CALL と PROGNAME 変数によりサブプログラム PART1 の間接呼び出しをします。
…	

2.24.3.6 呼び出しプログラム要素を指定した間接サブプログラム呼び出し(CALL BLOCK ... TO ...)

CALL とキーワードの組み合わせ BLOCK ... TO を使用して、サブプログラムを間接的に呼び出し、開始ラベルと終了ラベルで指定されたプログラム区間を実行します。

構文

```
CALL <プログラム名称> BLOCK <開始ラベル> TO <終了ラベル>
CALL BLOCK <開始ラベル> TO <終了ラベル>
```

意味

CALL:	間接サブプログラム呼び出しの命令です。	
<プログラム名称>:	実行するプログラム区間を含むサブプログラムの名称(変数または定数) (**指定** は**オプション**)。	
	タイプ:	STRING
	注: <プログラム名称>をプログラム指令していない場合は、<開始ラベル>と<終了ラベル>が示すプログラム区間は、実行中のプログラムで検索され、実行されます。	
BLOCK ... TO ... :	プログラム区間を間接的に実行するためのキーワードの組み合わせ	
<開始ラベル>:	実行するプログラム区間の先頭を参照する変数	
	タイプ:	STRING
<終了ラベル>:	実行するプログラム区間の末尾を参照する変数	
	タイプ:	STRING

例

メインプログラム

プログラムコード	コメント
...	
DEF STRING[20] STARTLABEL, ENDLABEL	; 開始ラベルと終了ラベルの変数を定義します。
STARTLABEL="LABEL_1"	
ENDLABEL="LABEL_2"	
...	
CALL "CONTUR_1" BLOCK STARTLABEL TO ENDLABEL	; 間接サブプログラム呼び出し、および呼び出しプログラム区間に関連する識別子です。
...	

サブプログラム:

プログラムコード	コメント
PROC CONTUR_1 ...	
LABEL_1	; 開始ラベル:実行するプログラム区間の開始
N1000 G1 ...	

プログラムコード	コメント
...	
LABEL_2	; 終了ラベル:実行するプログラム区間の終了
...	

2.24.3.7 ISO 言語で作成したプログラムの間接呼び出し(ISOCALL)

ISO 言語で作成したプログラムは、間接プログラム呼び出し ISOCALL を使用して呼び出すことができます。この場合、マシンデータで設定した ISO モードが有効です。オリジナルの実行モードは、プログラムの終了時に再び有効になります。ISO モードをマシンデータで設定していない場合は、サブプログラムを SIEMENS モードで呼び出します。

ISO について詳しくは、
参照先
の「機能説明書 G コード言語」を参照してください。

構文

```
ISOCALL <プログラム名称>
```

意味

ISOCALL:	マシンデータで設定した ISO モードを有効にする、間接サブプログラム呼び出しのキーワードです。
<プログラム名称>:	ISO 言語で作成したプログラムの名称(変数または定数、STRING タイプ)です。

例:サイクルプログラミングによる ISO モードからの輪郭の呼び出し

プログラムコード	コメント
0122_SPF	; ISO モードの輪郭記述
N1010 G1 X10 Z20	
N1020 X30 R5	
N1030 Z50 C10	
N1040 X50	
N1050 M99	
N0010 DEF STRING[5] PROGNAME = "0122"	; SIEMENS パートプログラム (サイクル)
...	
N2000 R11 = $AA_IW[X]	
N2010 ISOCALL PROGNAME	

プログラムコード	コメント
N2020 R10 = R10+1 ... N2400 M30	; ISO モードでプログラム 0122.spf を実行します

2.24.3.8 パス指定とパラメータによるサブプログラム呼び出し(PCALL)

PCALL を使用すると、絶対パスとパラメータ転送によりサブプログラムを呼び出すことができます。

構文

PCALL <パス/プログラム名称>(<パラメータ 1>,…,<パラメータ n>)

意味

PCALL:	絶対パス名称によるサブプログラム呼び出しのキーワードです。
<パス/プログラム名称>:	サブプログラム名称を含む絶対パスデータ パスデータに関する規則については、「プログラムメモリファイルのアドレス指定 (ページ 255)」を参照してください。 絶対パス名称を指定しない場合は、PCALL が、プログラム識別子による標準のサブプログラム呼び出しと同様に動作します。 プログラム名称は、接頭語を付けずにファイル識別子なしで指定します。接頭語やファイル識別子を使用してプログラム名称をプログラム指令する場合は、EXTERN 命令を使用して、接頭語とファイル識別子を明示的に宣言してください。
<パラメータ 1>, ...:	サブプログラムの PROC 命令に従った実パラメータです。

例

プログラムコード
PCALL/_N_WKS_DIR/_N_SHAFT_WPD/SHAFT(パラメータ 1,パラメータ 2,…)

2.24.3.9 サブプログラム呼び出しの拡張検索パス(CALLPATH)

サブプログラム呼び出しの検索パスを、CALLPATH 命令を使用して拡張できます。これは、サブプログラムの完全な絶対パス名称を指定しなくても、選択していないワークディレクトリから直接、サブプログラムも呼び出すことができるということです。

グローバルなサブプログラムを保存するために外部プログラムメモリ上の別のディレクトリを使用する場合、EES モードでは別のアプリケーションオプション「EES without GDIR」(GDIR を含まない EES)を使用できます。この場合、CALLPATH を使用して、検索パスをこのサブプログラムディレクトリによって拡張できます。

拡張検索パスは、ユーザーサイクル(_N_CUS_DIR)の項目の前に作成されます。

拡張検索パスは、以下の事象が発生すると、再び選択解除されます。

- ブランクを含む CALLPATH
- パラメータのない CALLPATH
- パートプログラムの終了
- リセット

構文

```
CALLPATH("<パス名称>")
```

意味

CALLPATH:	プログラム指令可能な拡張検索パスのキーワードです。 個別のパートプログラム行にプログラム指令します。
<パス名称>:	定数または変数、STRING タイプ。 検索パスの拡張に使用するディレクトリの絶対パス名称を含みます。 パスデータに関する規則については、「プログラムメモリファイルのアドレス指定 (ページ 255)」を参照してください。

例

検索パスを特定のワークディレクトリによって拡張します。

プログラムコード

```
...
CALLPATH ("/_N_WKS_DIR/_N_MYWPD_WPD")
...
```

これは、次の検索パスが設定されることを意味します(項番 5 が新しい設定です)。

1. 現在のディレクトリ/*名称*
2. 現在のディレクトリ/*名称_SPF*
3. 現在のディレクトリ/*名称_MPF*
4. //NC:/_N_SPF_DIR / *名称_SPF*
5. **/_N_WKS_DIR/_N_MYWPD_WPD/名称_SPF**
6. /N_CUS_DIR/*名称_SPF*
7. /_N_CMA_DIR/*名称_SPF*
8. /_N_CST_DIR/*名称_SPF*

必要条件

- CALLPATH は、プログラム指令パス名称が実在するかどうかをチェックします。エラーが発生した場合は、パートプログラムの実行がブロック修正アラーム 14009 により中断されます。
- CALLPATH は、INI ファイルにプログラム指令することもできます。この場合は、INI ファイルの処理時のみ有効となります(WPD-INI ファイル、または 1 番目のチャネル _N_CH1_UFR_INI のフレームなど、NC の有効データの初期化プログラム)。検索パスは再度リセットされます。

2.24.3.10 外部サブプログラムの実行(840D sl) (EXTCALL)

パートプログラムを外部メモリからロードして EXTCALL 命令で実行できます。

外部メモリとしては、以下が使用できます。

- ローカルドライブ
- ネットワークドライブ
- USB ドライブ

注記

USB ドライブ上の外部プログラムの処理には、操作パネルまたは TCU の USB インタフェースのみインタフェースとして使用できます。

通知

USB フラッシュメモリ使用時の工具/ワークの損傷

外部サブプログラムの実行には、USB フラッシュメモリを使用しないことをお勧めします。サブプログラムの実行中に、フラッシュメモリに誤ってぶつかったり間違って抜いたために接触不良が発生すると、USB フラッシュメモリの通信が中断され、加工は即座に停止します。工具かワークまたは両方が損傷する恐れがあります。

外部プログラムのパスの初期設定

外部プログラムディレクトリのパスは、次のセッティングデータを使用して設定しておくことができます。

SD42700 $SC_EXT_PROG_PATH

これは、EXTCALL 呼び出しで指定したプログラムのパスと識別子で、呼び出すプログラムの完全なパスが形成されます。

注記

パラメータ

外部プログラムを呼び出すときに、パラメータはいずれも外部プログラムに転送されません。

構文

```
EXTCALL("<パス/><プログラム名称>")
```

意味

EXTCALL:	外部サブプログラムを呼び出す命令です。	
"<パス/><プログラム名称>":	STRING タイプの定数/変数です。	
	<パス>:	絶対パスまたは相対パスのデータ(**任意選択**)
	<プログラム名称>:	プログラム名称は、接頭語「_N_」を付けずに指定します。 文字「_」または「.」を使用して、プログラム名称にファイル拡張子(「MPF」、「SPF」)を付けることができます(**任意選択**)。 例: "SHAFT" "SHAFT_SPF" "SHAFT.SPF"

パスの指定:短縮名称

次のような短縮名称でパスを指定できます。

- ローカルドライブ:**"LOCAL_DRIVE:"**
- CF カード:**"CF_CARD:"**
- USB ドライブ(操作パネル):**"USB:"**

または、短縮名称「CF_CARD:」と「LOCAL_DRIVE:」を使用可能です。

例

ローカルドライブからの実行

「_N_MAIN_MPF」メインプログラムが NC メモリに格納され、実行するために選択されます。

サブプログラム「SP_1」

外部サブプログラム「SP_1.SPF」または「SP_1.MPF」は、ローカルドライブのディレクトリ「/user/sinumerik/data/prog/WKS.DIR/WST1.WPD」にあります。

外部プログラムディレクトリのパスは、以下を使用して設定されます。

SD42700 $SC_EXT_PROG_PATH = LOCAL_DRIVE:WKS.DIR/WST1.WPD

注記

外部サブプログラム呼び出し用のパスの指定

- 初期設定なし:「LOCAL_DRIVE:WKS.DIR/WST1.WPD/SP_1」
- 初期設定あり:「SP_1」

サブプログラム「SP_2」

外部サブプログラム「SP_2.SPF」または「SP_2.MPF」が、USB ドライブの WKS.DIR/WST1.WPD ディレクトリにあります。外部プログラムディレクトリへの初期パス設定がサブプログラム「SP_1」のパスで使用され、メインプログラムでは再作成もされません。そのため、サブプログラム「SP_2」を呼び出すときには完全なパスを指定する必要があります。

メインプログラム「MAIN」

プログラムコード

```
N010 PROC MAIN
N020 ...
N030 EXTCALL("SP_1")
N030 EXTCALL("USB:WKS.DIR/WST1.WPD/SP_2")
N050 ...
N060 M30
```

関連情報

絶対パス名称による EXTCALL 呼び出し

指定したパスにサブプログラムがある場合は、EXTCALL 呼び出しで実行されます。指定したパスにサブプログラムがない場合は、EXTCALL 呼び出しによるプログラムの実行は中止されます。

相対パス名称による、または相対パス名称を使用しない EXTCALL 呼び出し

相対パス名称による、または相対パス名称を使用しない EXTCALL 呼び出しの場合は、使用可能なプログラムメモリを、次のように検索します。

1. SD42700 $SC_EXT_PROG_PATH にパス名称が設定されている場合は、EXTCALL 呼び出しで指定されたデータ(プログラム名称または相対パス名による)の検索が最初におこなわれ、このパスから開始されます。絶対パスは、次の文字を結合して得られます。
   - SD42700 $SC_EXT_PROG_PATH の初期パス指定
   - 区切り文字「/」
   - EXTCALL 命令でのパス指定とサブプログラム名称
2. サブプログラムが 1 で見つからなかった場合は、ユーザーメモリのディレクトリが検索されます。

検索は、サブプログラムが初めて見つかったときに終了します。サブプログラムが見つからない場合は、EXTCALL 呼び出しによるプログラムの実行は中止されます。

調整可能な再ロードメモリ(FIFO バッファ)

外部サブプログラムの実行には、再ロードメモリが必要です。再ロードメモリのサイズは 30KB に設定済みで、(MD18360 MM_EXT_PROG_BUFFER_SIZE を使用して)工作機械メーカだけが変更できます。

注記

ジャンプ命令を含むサブプログラム

ジャンプ命令(GOTOF、GOTOB、CASE、FOR、LOOP、WHILE、REPEAT、IF、ELSE、ENDIF など.)を含む外部サブプログラムの場合、ジャンプ先はロードバッファメモリ内になければなりません。

注記

ShopMill/ShopTurn プログラム

ファイルの最後に輪郭の説明を追加した場合、ShopMill および ShopTurn プログラムをすべて読み取り専用メモリに保存する必要があります。

外部サブプログラムを並列実行する場合は、別の再メモリロードが必要です。

リセット/エンドオブプログラム/電源投入

リセットおよび電源投入をおこなうと、外部サブプログラム呼び出しが中断され、関連するロードメモリが消去されます。

「外部メモリからの実行」で選択されたプログラムは、リセット/エンドオブプログラムの後も、「外部メモリからの実行」で選択されたままです。外部プログラムメモリがまだ使用可能であれば、動作は内部で選択したプログラムと同じです。

参照先

「外部メモリからの実行」の詳細については、以下を参照してください。

機能マニュアル 基本機能; モードグループ、チャネル、プログラム運転、リセット応答 (K1)

2.24.3.11 外部サブプログラムの実行(828D) (EXTCALL)

パートプログラムを外部メモリからロードして EXTCALL 命令で実行できます。

外部メモリとしては、以下が使用できます。

- ユーザー CF カード
- ネットワークドライブ
- USB ドライブ

注記

USB ドライブに存在する外部プログラムの実行用インタフェースとしては、操作パネルの USB インタフェース(PPU)だけを使用できます。

通知

USB フラッシュメモリ使用時の工具/ワークの損傷

外部サブプログラムの実行には、USB フラッシュメモリを使用しないことをお勧めします。サブプログラムの実行中に、フラッシュメモリに誤ってぶつかったり間違って抜いたために接触不良が発生すると、USB フラッシュメモリの通信が中断され、加工は即座に停止します。工具かワークまたは両方が損傷する恐れがあります。

外部プログラムのパスの初期設定

外部プログラムディレクトリのパスは、次のセッティングデータを使用して設定しておくことができます。

SD42700 $SC_EXT_PROG_PATH

これのパスと、EXTCALL 呼び出しで指定したプログラムのパスと識別子で、呼び出すプログラムの完全なパスが形成されます。

注記

パラメータ

外部プログラムを呼び出すときに、パラメータはいずれも外部プログラムに転送されません。

構文

```
EXTCALL("<パス/><プログラム名称>")
```

意味

EXTCALL:	外部サブプログラムを呼び出す命令です。	
"<パス/><プログラム名称>":	STRING タイプの定数/変数です。	
	<パス>:	絶対パスまたは相対パスのデータ(任意選択)
	<プログラム名称>:	プログラム名称は、接頭語「_N_」を付けずに指定します。 文字「_」または「.」を使用して、プログラム名称にファイル拡張子(「MPF」、「SPF」)を付けることができます(任意選択)。 例: "SHAFT" "SHAFT_SPF" "SHAFT.SPF"

パスの指定:短縮名称

次のような短縮名称でパスを指定できます。

- ユーザー CF カード:"CF_CARD:"
- USB ドライブ(操作パネル):"USB:"

例

「_N_MAIN_MPF」メインプログラムが NC メモリに格納され、実行するために選択されます。

サブプログラム「SP_1」

外部サブプログラム「SP_1.SPF 」または「SP_1.MPF 」が、ユーザー CF カードの「/WKS.DIR /WST1.WPD」ディレクトリに存在します。

外部プログラムディレクトリのパスは、以下を使用して設定されます。

SD42700 $SC_EXT_PROG_PATH = CF_CARD:WKS.DIR/WST1.WPD

注記

外部サブプログラム呼び出し用のパスの指定

- 初期設定なし:「CF_CARD:WKS.DIR/WST1.WPD/SP_1」
- 初期設定あり:「SP_1」

サブプログラム「SP_2」

外部サブプログラム「SP_2.SPF」または「SP_2.MPF」は、USB ドライブの WKS.DIR / WST1.WPD ディレクトリにあります。外部プログラムディレクトリへの初期パス設定がサブプログラム「SP_1」のパスで使用され、メインプログラムでは再作成もされません。そのため、サブプログラム「SP_2」を呼び出すときには完全なパスを指定する必要があります。

メインプログラム「MAIN」

プログラムコード

```
N010 PROC MAIN
N020 ...
N030 EXTCALL("SP_1")
N030 EXTCALL("USB:WKS.DIR/WST1.WPD/SP_2")
N050 ...
N060 M30
```

関連情報

絶対パス名称による EXTCALL 呼び出し

指定したパスにサブプログラムがある場合は、EXTCALL 呼び出しで実行されます。指定したパスにサブプログラムがない場合は、EXTCALL 呼び出しによるプログラムの実行は中止されます。

相対パス名称による、または相対パス名称を使用しない EXTCALL 呼び出し

相対パス名称による、または相対パス名称を使用しない EXTCALL 呼び出しの場合は、使用可能なプログラムメモリを、次のように検索します。

1. SD42700 $SC_EXT_PROG_PATH にパス名称が設定されている場合は、EXTCALL 呼び出しで指定されたデータ(プログラム名称または相対パス名称による)の検索が最初におこなわれ、このパスから開始されます。絶対パスは、次の文字を結合して得られます。
   - SD42700 $SC_EXT_PROG_PATH の初期パス指定
   - 区切り文字「/」
   - EXTCALL 命令でのパス指定とサブプログラム名称
2. サブプログラムが 1 で見つからなかった場合は、ユーザーメモリのディレクトリが検索されます。

検索は、サブプログラムが初めて見つかったときに終了します。サブプログラムが見つからない場合は、EXTCALL 呼び出しによるプログラムの実行は中止されます。

調整可能な再ロードメモリ(FIFO バッファ)

外部サブプログラムの実行には、再ロードメモリが必要です。再ロードメモリのサイズを設定しておきます(MD18360 MM_EXT_PROG_BUFFER_SIZE を参照ください)。

注記

ジャンプ命令を含むサブプログラム

ジャンプ命令(GOTOF、GOTOB、CASE、FOR、LOOP、WHILE、REPEAT、IF、ELSE、ENDIF など.)を含む外部サブプログラムの場合、ジャンプ先はロードバッファメモリ内になければなりません。

注記

ShopMill/ShopTurn プログラム

ファイルの最後に輪郭の説明を追加した場合、ShopMill および ShopTurn プログラムをすべて読み取り専用メモリに保存する必要があります。

外部サブプログラムを並列して実行は、別の再メモリロードが必要です。

リセット/エンドオブプログラム/電源投入

リセットおよび電源投入をおこなうと、外部サブプログラム呼び出しが中断され、関連するロードメモリが消去されます。

「外部メモリからの実行」で選択されたプログラムは、リセット/エンドオブプログラムの後も、「外部メモリからの実行」で選択されたままです。外部プログラムメモリがまだ使用可能であれば、動作は内部で選択したプログラムと同じです。

参照先

「外部メモリからの実行」の詳細については、以下を参照してください。

機能マニュアル 基本機能; モードグループ、チャネル、プログラム運転、リセット応答 (K1)

2.25 マクロ機能(DEFINE ... AS)

通知
マクロはプログラムをより複雑にする マクロにより、制御装置のプログラミング言語が大幅に変わる場合があります。マクロ機能を使用する場合は、細心の注意が必要です。

マクロとは、マクロ名称と一緒に、個々の命令処理を割り当てたものです。プログラム運転中にマクロを呼び出すと、プログラム名称で指令された命令群が連続的に実行されます。

有効範囲(つまり、マクロ定義が有効な範囲)に応じて、以下のマクロカテゴリがあります。

- ローカルマクロ
  ローカルマクロは、実行時にメインプログラム以外の NC プログラムの先頭で定義した変数です。この変数は、NC プログラムが呼び出されると作成され、プログラムリセットの終了時またはコントロールシステムの起動時に削除されます。ローカルマクロには、この変数を定義した NC プログラム内でのみアクセスできます。
- プログラムグローバルマクロ
  プログラムグローバルマクロは、実行時にメインプログラム以外の NC プログラムの開始場所で定義した変数です。この変数は、NC プログラムが呼び出されると作成され、プログラムリセットの終了時またはコントロールシステムの起動時に削除されます。プログラムグローバルマクロは、メインプログラムと、すべてのサブプログラム内でアクセスできます。

  注記

  プログラムグローバルマクロの使用可能性

  次のマシンデータを設定した場合は、メインプログラムで定義したプログラムグローバルマクロはサブプログラムにのみ表示されます。

  MD11120 $MN_LUD_EXTENDED_SCOPE = 1

  MD11120 = 0 の場合、メインプログラムで定義したプログラムグローバルマクロはメインプログラムでのみ使用できます。

- グローバルマクロ
  グローバルマクロは、定義ファイル(マクロファイル)に定義され、プログラムリセット終了時、またはコントロールシステムの次回起動の後も保持される、NC グローバルマクロまたはチャネルグローバルマクロです。グローバルマクロは、任意のメインプログラムまたはサブプログラムで呼び出して実行できます。

  注記

  NC プログラムの**外部**マクロファイルのマクロを使用するには、マクロファイルを NC にダウンロードしてください。

マクロを使用する前に定義してください。この場合は、以下の規則を守ってください。

- あらゆる識別子、G、M、H の各機能、および L サブプログラムの名称は、マクロに定義できます。
- マクロは、プログラムの先頭、または専用の定義ファイル(マクロファイル)に定義できます。

- ローカルマクロとプログラムグローバルマクロは、プログラムの先頭に定義します。
- グローバルファイルは、_N_DEF_DIR/_N_UMAC_DEF などのマクロファイルに定義してください。
- G 命令マクロは、グローバルマクロとしてしか定義できません。
- H 機能と L 機能は 2 桁でプログラム指令できます。
- M 命令と G 命令は 3 桁でプログラム指令できます。

注記

キーワードと予約名称は、マクロを使用して上書きすることはできません。

構文

マクロ定義:
`DEFINE <マクロ名> AS <操作 1> <操作 2> ...`

NC プログラムでの呼び出し:
<マクロ名>

意味

`DEFINE ... AS`:	マクロを定義するキーワードの組み合わせです。
<マクロ名>:	マクロ名称です。 マクロ名称として使用できるのは識別子のみです。 マクロは、NC プログラムからマクロ名称を使用して呼び出します。
<操作 1>:	マクロ内の最初のプログラム命令
<操作 2>:	マクロ内の 2 番目のプログラム命令

必要条件

ネスト

マクロを入れ子にすることはできません。

例

例 1:プログラムの先頭のマクロ定義

プログラムコード	コメント
DEFINE LINE AS G1 G94 F300	; マクロ定義
...	
N70 LINE X10 Y20	; マクロの呼び出し
...	

例 2 :マクロファイルでのマクロ定義

プログラムコード	コメント
DEFINE M6 AS L6	; サブプログラムを工具交換時に呼び出し、必要なデータ送受信を処理します。実際の M 機能はサブプログラムのなかで出力されます (M106 など)。
DEFINE G81 AS DRILL(81)	; DIN G 命令のエミュレーション
DEFINE G33 AS M333 G333	; ねじ切りのときに PLC によって同期を要求します。ユーザーのプログラムが同一になるよう、オリジナルの G 命令の G33 の名称が、マシンデータによって G333 へ変更されました。

例 3 :外部マクロファイル

マクロファイルは、外部マクロファイルを制御装置に読み込んだ後に、NC にダウンロードしてください。その後にだけ、マクロを NC プログラムで使用できます。

プログラムコード	コメント
%_N_UMAC_DEF	
;$PATH=/_N_DEF_DIR	; ユーザー用マクロ
DEFINE PI AS 3.14	
DEFINE TC1 AS M3 S1000	
DEFINE M13 AS M3 M7	; 主軸は右回り、クーラントはオン
DEFINE M14 AS M4 M7	; 主軸は左回り、クーラントはオン
DEFINE M15 AS M5 M9	; 主軸は停止、クーラントはオフ
DEFINE M6 AS L6	; 工具交換プログラムの呼び出し
DEFINE G80 AS MCALL	; 穴あけサイクルの選択解除
M30	

ファイルとプログラムの管理 3

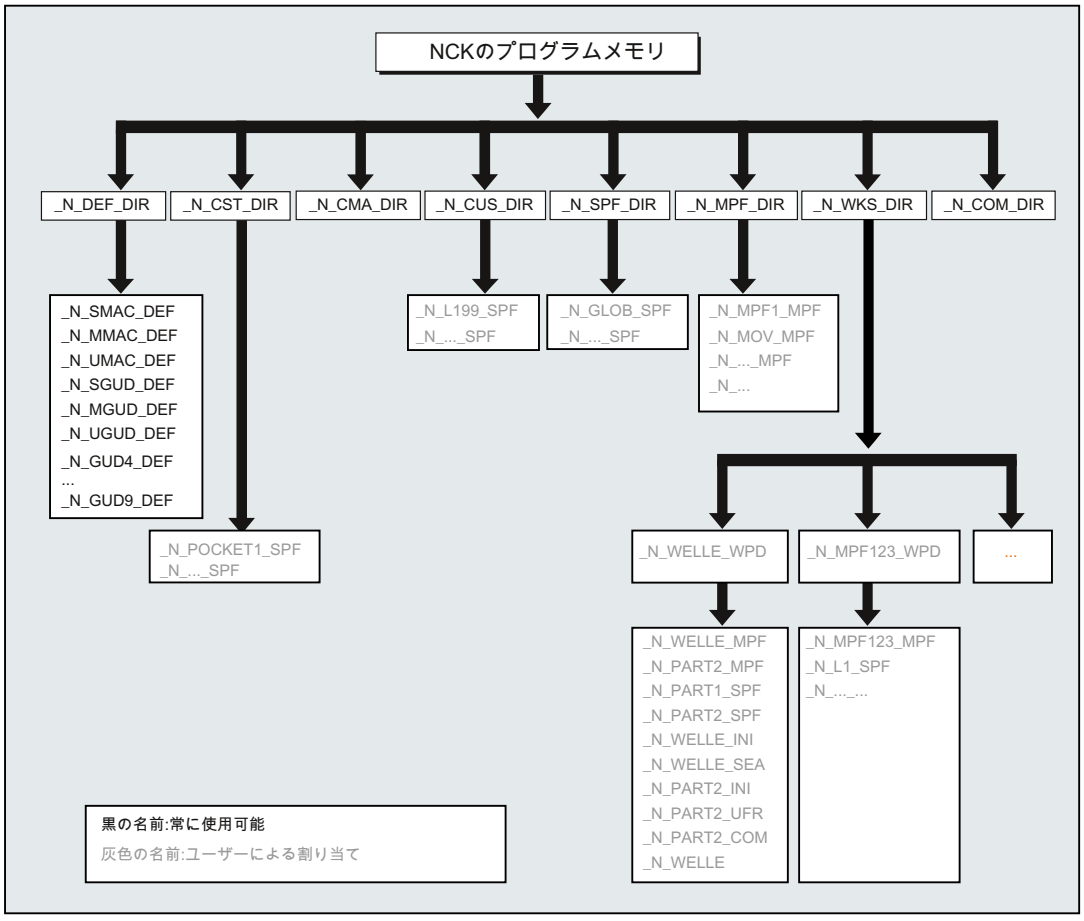
3.1 プログラムメモリ

3.1.1 NC プログラムメモリ

ファイルとプログラム(メインプログラムとサブプログラム、マクロ定義)は、不揮発性のプログラムメモリに保存されます(→パッシブファイルシステム)。

参照先:
機能マニュアル 上級機能; メモリ構成(S7)

多くのファイルタイプもここに一時的に格納されます。これらのファイルは必要に応じて(特定のワークの加工時の初期化のためなど)、作業メモリに転送されます。



標準ディレクトリ

以下の標準ディレクトリを使用できます。

ディレクトリ	内容
_N_DEF_DIR	データモジュールとマクロモジュール
_N_CST_DIR	標準サイクル
_N_CMA_DIR	メーカサイクル
_N_CUS_DIR	ユーザーサイクル
_N_WKS_DIR	ワーク
_N_SPF_DIR	グローバルサブプログラム
_N_MPF_DIR	メインプログラム
_N_COM_DIR	コメント

ファイルタイプ

以下のファイルタイプをメインメモリへ格納できます。

ファイルタイプ	説明
<名称>_MPF	メインプログラム
<名称>_SPF	サブプログラム
<名称>_TEA	マシンデータ
<名称>_SEA	セッティングデータ
<名称>_TOA	工具オフセット
<名称>_UFR	ゼロオフセット/フレーム
<名称>_INI	初期化ファイル
<名称>_GUD	グローバルユーザーデータ
<名称>_RPA	R 変数
<名称>_COM	コメント
<名称>_DEF	グローバルユーザーデータとマクロの定義

ワークメインディレクトリ(_N_WKS_DIR)

ワークメインディレクトリは、プログラムメモリの標準セットアップのなかにあり、_N_WKS_DIR という名称です。ワークメインディレクトリには、プログラム指令したワークのすべてのワークディレクトリが含まれます。

ワークディレクトリ(..._WPD)

ワークディレクトリには、ワークの加工に必要なすべてのファイルが含まれます。これらのファイルは、メインプログラム、サブプログラム、あらゆる初期化プログラム、およびコメントファイルなどです。

パートプログラムの最初の起動時には、選択したプログラムに応じて初期化プログラムが 1 回だけ実行されます(マシンデータ MD11280 $MN_WPD_INI_MODE に従います)。

例:

SHAFT ワーク用に作成されたワークディレクトリ_N_SHAFT_WPD には、次のファイルが含まれます。

ファイル	説明
_N_SHAFT_MPF	メインプログラム
_N_PART2_MPF	メインプログラム
_N_PART1_SPF	サブプログラム
_N_PART2_SPF	サブプログラム
_N_SHAFT_INI	ワークのデータの一般初期化プログラム
_N_SHAFT_SEA	セッティングデータ初期化プログラム
_N_PART2_INI	Part 2 プログラムのデータの一般初期化プログラム
_N_PART2_UFR	Part 2 プログラムのフレームデータの初期化プログラム
_N_SHAFT_COM	コメントファイル

ワークディレクトリに、NC が加工に直接必要としないデータを保存することもできます。ASCII ファイルの他に、JPG 形式の画像や PDF 形式の記述など、バイナリファイルも使用できます。これらのファイルを NC がバイナリファイルとして解釈できるように、ファイル拡張子を NCK で既知にしておいてください(セットアップ時に MD17000 $MN_EXTENSIONS_OF_BIN_FILES で設定。初期設定で次のファイル拡張子が設定されています:JPG、GIF、PNG、BMP、PDF、ICO、HTM)。

加工ワークの選択

チャネルで実行するワークディレクトリを選択できます。**同じ名称**のメインプログラム、または 1 つのメインプログラム(_MPF)のみがこのディレクトリに格納されている場合は、それが自動的に選択され、実行されます。

参照先:
操作マニュアル

3.1.2 外部プログラムメモリ

NC のパッシブファイルシステムに加えて、(たとえば、ローカルドライブやネットワークドライブ上の)外部プログラムメモリも機械で使用できます。

「外部から実行」機能や「EES (外部記憶からの実行)」機能を使用して、外部プログラムメモリからパートプログラムを**直接**実行できます。

参照先:
機能マニュアル 基本機能; K1:モードグループ、チャネル、プログラム運転、リセット応答

グローバルパートプログラムメモリ(GDIR)

ドライブを宣言するときに、いずれかのドライブをグローバルパートプログラムメモリ(GDIR)に指定することができます。

参照先:
操作マニュアル:章:「プログラムの管理」>「ドライブのセットアップ」

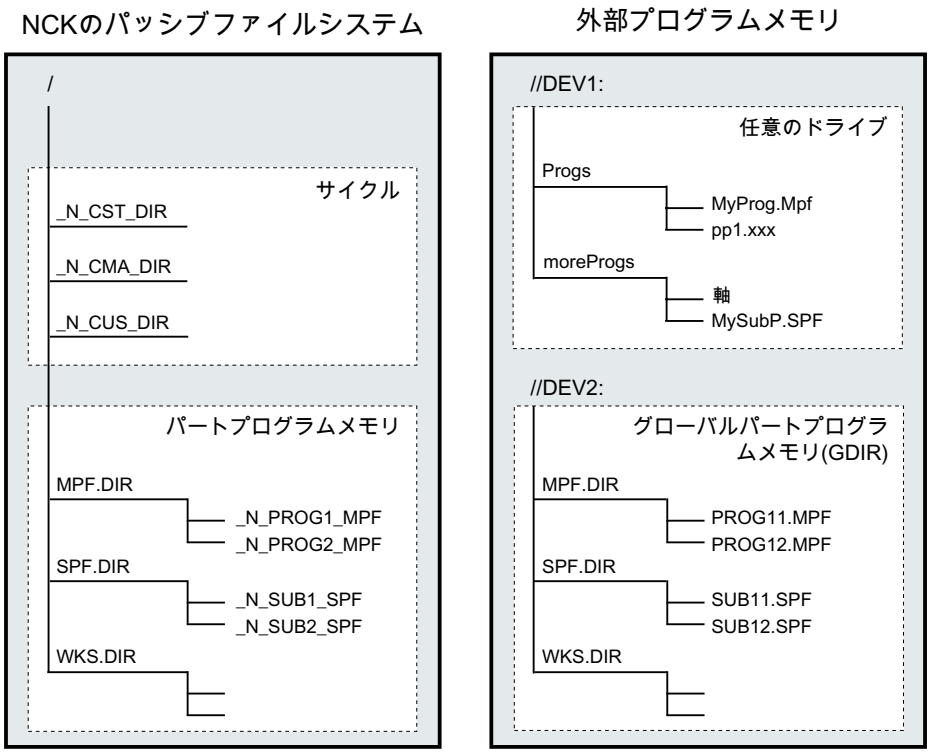
システムはドライブ上に MPF.DIR、SPF.DIR、WKS.DIR の各ディレクトリを自動的に作成します。これら 3 つのディレクトリで GDIR が形成されます。

GDIR は EES 機能のための役割を担うだけです。GDIR は、ドライブ設定に応じて、NC パートプログラムメモリを置き換えるか拡張します。ただし、GDIR の作成は、EES 操作に必須なわけではありません。

GDIR のディレクトリとファイルは、パートプログラムでパッシブファイルシステムと同じ方法でアドレス指定できます。このため、パスの詳細を含む NC プログラムを、パッシブファイルシステムから GDIR へ互換性を保って転送することができます。GDIR の SPF.DIR ディレクトリは、サブプログラムの検索パスに含められます。

プログラム構成

外部プログラムメモリでのプログラム構成を次の図に示します。



大文字小文字を区別しないファイルシステム

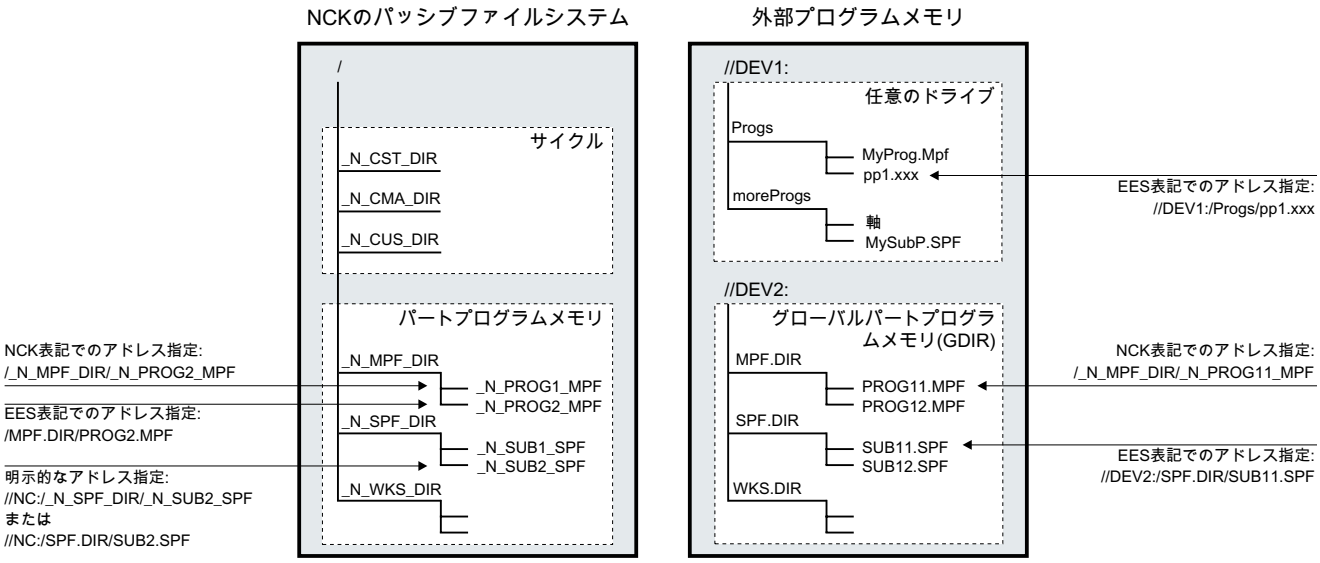
注記

ファイルのアドレス指定(「プログラムメモリファイルのアドレス指定 (ページ 255)」を参照してください)での大文字小文字の区別による問題を回避するために、外部プログラムメモリには**大文字小文字を区別しない**ファイルシステムを使用してください。

3.1.3 プログラムメモリファイルのアドレス指定

プログラムメモリ内のファイルは、ファイル処理命令(WRITE、DELETE、READ、ISFILE、FILEDATE、FILETIME、FILESIZE、FILESTAT、FILEINFO など)でアドレス指定され、絶対パスとファイル名、またはファイル名のみで参照されます。ファイル名のみで参照する場合、選択したプログラムのパスがファイルパスとして使用されます。

NC/EES 表記でのアドレス指定:



パッシブファイルシステムのファイルのアドレス指定

パッシブファイルシステムのファイルは通常、ドライブ名を指定せずに **NC 表記** (ディレクトリ名とファイル名の先頭がドメイン識別子「_N_」で、ファイル識別子の区切り文字が「_」)でアドレス指定されます。ただし、 **EES 表記** (ドメイン識別子「_N_」なし、ディレクトリ/ファイル拡張子の区切り文字が「.」)でのアドレス指定も使用できます。

例:

- NC 表記: "/_N_SPF_DIR/_N_SUB1_SPF"
- EES 表記: "/SPF.DIR/SUB1.SPF"

注記

EES 表記でのパッシブファイルシステムのファイルのアドレス指定は、以下の規則に従って内部で NC 表記に変換されます。

- ディレクトリ名とファイル名に、ドメイン識別子「_N_」を追加します。
- ディレクトリ名またはファイル名の末尾から 4 番目の文字がピリオド(「.」)の場合、これをアンダースコア(「_」)に変換します。

事前定義されたドライブ名「//NC:」を使用して、パッシブファイルシステムを明確にアドレス指定することもできます。

例:

- NC 表記: "//NC:/_N_SPF_DIR/_N_SUB1_SPF"
- EES 表記: "//NC:/SPF.DIR/SUB1.SPF"

外部プログラムメモリのファイルのアドレス指定

GDIR として記録されていない外部プログラムメモリのファイルは、EES 表記でアドレス指定してください。アドレス指定パスの先頭で、ドライブ名(例: 「//DEV1:」)を指定してください。/user/sinumerik/hmi/cfg/logdrive.ini で設定されているすべてのシンボリックデバイス名を使用できます。

例:

- EES 表記: "//DEV1:/MyProgDir/pp1.xxx"
- NC 表記不可

グローバルパートプログラムメモリ(GDIR)のファイルのアドレス指定

GDIR のファイルをアドレス指定する場合、EES 表記でのパス指定に加えて、NC 表記でのパス指定も使用できます。

例:

- EES 表記: "//DEV2:/MPF.DIR/PROG11.MPF"
- NC 表記: "/_N_MPF_DIR/_N_PROG11_MPF"

注記

NC 表記での GDIR のファイルのアドレス指定は、以下の規則に従って内部で EES 表記に変換されます。

- ディレクトリ名およびファイル名のドメイン識別子「_N_」が削除されます。
- ディレクトリ名またはファイル名の最後から 4 番目の文字がアンダースコア(「_」)の場合、これがピリオド(「.」)に変換されます。

パス指定の規則

完全なパス指定は、ドライブ名、ディレクトリパス、およびファイル名で構成されます。

ドライブ名

ドライブ名の指定には次の規則が適用されます。

- /user/sinumerik/hmi/cfg/logdrive.ini で設定されているすべてのシンボリックデバイス名を使用できます。
- 先頭は記号「//」で、その後に 1 つ以上の文字または数字が続きます。
- 後続の文字は、英文字、数字、「_」および空白文字の任意の組み合わせにすることができます。

- 名前の最後は文字または数字で、その後に「:」を付けます。
- その他の特殊文字は使用できません。

注記

パッシブファイルシステムにはデバイス名「//NC:」が事前定義されています。

例:

- 外部プログラムメモリ:
  - //Drive1:
  - //Drive_1:
  - //Drive 1:
  - //A B:
  - //1 B C 2:

ディレクトリパス

ディレクトリパスの指定には次の規則が適用されます。

- 「/」はディレクトリパスの最初と最後に付けられ、また個々のパス区分の区切り文字として使用されます。

  注記

  二重スラッシュ(「//」)は、ディレクトリパス内では使用 **できません**。

- ディレクトリ名:
  - ディレクトリ名の先頭は英文字または数字にしてください。NC 表記でのアドレス指定の場合のみ、ディレクトリ名の先頭にドメイン識別子「_N_」を付けます。
  - 後続の文字は英文字、数字、および「_」の任意の組み合わせにすることができます。

    注記

    外部プログラムメモリの場合は、ディレクトリ名に空白文字も使用できます。ただし、外部プログラムメモリをグローバルパートプログラムメモリ(GDIR)として作成する場合は例外です。

  - その他の特殊文字は使用できません。
- ディレクトリの拡張子:
  - ディレクトリの拡張子は英文字/数字 3 文字にしてください。
  - 拡張子とディレクトリ名を「_」(NC 表記)または「.」(EES 表記)で区切ります。

  注記

  パッシブファイルシステムのディレクトリ拡張子は_DIR と_WPD のみです。

例:

- パッシブファイルシステムまたは GDIR:
  - NC 表記: _N_WKS_DIR/_N_MYNCPROGS_WPD/...
  - EES 表記:WKS.DIR/MYPROGS.WPD/...
- 外部プログラムメモリ:
  - /abc
  - /ab_c.def
  - /ab c1.def
  - /a b c .d11
  - /abc.def/ghi.klm

ファイル名

ファイル名には次の規則が適用されます。

- NC 表記でのアドレス指定の場合のみ、先頭にドメイン識別子「_N_」を付けます。
- 次の 2 文字は、英字 2 文字またはアンダースコアと英字 1 文字にしてください。

  注記

  この条件を満たす場合は、プログラム名称を指定するだけで、別のプログラムからサブプログラムとして NC プログラムを呼び出すことができます。ただし、プログラム名称の先頭が数字である場合、サブプログラム呼び出しは CALL 命令でのみ呼び出すことができます。

- 後続の文字は英文字、数字、および「_」の任意の組み合わせにすることができます。
- ファイル拡張子:
  - ファイル拡張子は英文字/数字 3 文字にしてください。

    注記

    パッシブファイルシステムで使用できるファイル拡張子については、「NC プログラムメモリ (ページ 251)」を参照してください。

  - 拡張子とファイル名を「_」(NC 表記)または「.」(EES 表記)で区切ります。

例:

- パッシブファイルシステムまたは GDIR:
  - NC 表記: _N_SUB1_SPF
  - EES 表記:SUB1.SPF
- 外部プログラムメモリ:
  - Part 1
  - _Part1
  - Part_1.spf
  - Part1.mpf

DIN サブプログラム名称

DIN サブプログラム名称には以下の規則が適用されます。

- 先頭文字は英字「L」にしてください。
- 後続の文字は数字です(1 つ以上)。
- ファイル拡張子:
  - ファイル拡張子は英字 3 文字にしてください。
  - 拡張子とファイル名を「_」(NC 表記)または「.」(EES 表記)で区切ります。

例:

- L123
- L1_SPF (NC 表記)または L1.SPF (EES 表記)

最大軌跡長

ドライブ名およびディレクトリパスの指定には最大 128 バイトを使用できます。ファイル名の最大長は 31 バイトです。完全パスの最大長は 159 バイトです。

3.1.4 サブプログラム呼び出しの検索パス

パスデータを使用しないサブプログラム呼び出しの場合、固定の検索パスを処理することで絶対パスが決定されます。

その後、次の順番でプログラムメモリが検索されます。

		ディレクトリ	説明
1		現在のディレクトリ/*名称*	現在のディレクトリはプログラムを選択するディレクトリです。 これは以下のディレクトリにすることができます。 • NC パートプログラムメモリまたはグローバルパートプログラムメモリ内のワークディレクトリまたは標準ディレクトリ_N_MPF_DIR または • 外部プログラムメモリの任意のディレクトリ
2		現在のディレクトリ/*名称_SPF*	
3		現在のディレクトリ/*名称_MPF*	
4	A	//NC:/_N_SPF_DIR / *名称_SPF*	NC パートプログラムメモリ内のサブプログラムディレクトリ
	B	//DEV2:/_N_SPF_DIR/名称_SPF [1)]	グローバルパートプログラムメモリ内のサブプログラムディレクトリ **注:** この検索ステップは、グローバルパートプログラムメモリが作成されていない場合、または NC パートプログラムメモリでプログラムが選択されていない場合は実行されません。

	ディレクトリ	説明
5	CALLPATH でプログラム指令された拡張検索パス(「サブプログラム呼び出しの拡張検索パス(CALLPATH) (ページ 235)」を参照してください)。 **注:** この検索ステップは、CALLPATH がプログラム指令されていない場合は実行されません。	
6	/_N_CUS_DIR/*名称_SPF*	ユーザーサイクルディレクトリ
7	/_N_CMA_DIR/*名称_SPF*	メーカーサイクルディレクトリ
8	/_N_CST_DIR/*名称_SPF*	標準サイクルディレクトリ

1) //DEV2:"たとえば、グローバルパートプログラムメモリが作成されているドライブを表します。

検索には次の規則が適用されます。

- 個々のサブプログラム呼び出しごとに検索パスが使用されます通過します。つまり、上位レベルのプログラムが配置されている場所には無関係です。
- ディレクトリに応じて、さまざまなファイルタイプが考慮されます。
- 検索はディレクトリ内で実行され、下位の、つまりネストされたディレクトリ内では実行されません。

3.1.5 パスとファイル名の問い合わせ

以下のシステム変数は、パートプログラムで読み込むことができ、NC プログラムのパスとファイル名を問い合わせるために使用できます。

システム変数	タイプ	意味
$P_STACK	INT	現在の NC プログラムを実行するプログラムレベルを返します。
$P_PATH[<n>]	STRING	フィールドインデックス<n>を使用して選択したプログラムレベルで処理する NC プログラムのパスを返します。 例: $P_PATH[0]は、メインプログラムのパス、たとえば「/_N_WKS_DIR/_N_WELLE_WPD/」を返します。 $P_PATH[$P_STACK - 1]は、呼び出しプログラムのパスを返します。 パスが NC のパッシブファイルシステムまたはグローバルパートプログラムメモリ(GDIR)に保存されている NC プログラムを参照する場合、そのパスは NC 表記で返されます。 パスがグローバルパートプログラムメモリ以外の外部プログラムメモリで実行される NC プログラムを参照する場合、$P_PATH はパスを EES 表記で返します。
$P_PROG[<n>]	STRING	フィールドインデックス<n>を使用して選択したプログラムレベルで処理する NC プログラムの名称を返します。 NC プログラムが NC のパッシブファイルシステムまたはグローバルパートプログラムメモリに保存されている場合、プログラム名称は NC 表記で返されます。 NC プログラムがグローバルパートプログラムメモリ以外の外部ドライブで実行される場合、$P_PROG はプログラム名称を EES 表記で返します。
$P_PROGPATH	STRING	現在処理中の NC プログラムのパスを返します。 $P_PROGPATH の呼び出しは、$P_PATH[$P_STACK]と同じです。

システム変数	タイプ	意味	
$P_IS_EES_PATH[<n>]	BOOL	$P_PATH[<n>]で返されるパスまたは$P_PROG[<n>]で返されるプログラム名称が、NC 表記と EES 表記のどちらに対応しているかを問い合わせます。	
		= FALSE	$P_PATH[<n>]と$P_PROG[<n>]は、NC 表記を返します。これは、各識別子に接頭語「_N_」があることを意味します。ファイル識別子の区切り文字は「_」です。 例: • NC 表記のパス: "/_N_WKS_DIR/_N_MYWPD_WPD/" • NC 表記のプログラム名称: "_N_MYPROG_MPF" NC 表記のパスは、NC のパッシブファイルシステムだけでなく、グローバルパートプログラムメモリも参照できます。
		= TRUE	$P_PATH[<n>]と$P_PROG[<n>]は、EEC 表記を返します。これは、各識別子に接頭語「_N_」がないことを意味します。ファイル識別子の区切り文字は「.」です。 例: • EES 表記のパス: "//DEV1:/WKS.DIR/MYWPD.WPD/" • EES 表記のプログラム名称:"MYPROG.MPF"

<n>:インデックス**<n>**は、パス情報を読み込むプログラムレベルを定義します(数値の範囲: 0 ... 17)。

注記

EES モードでは、グローバルパートプログラムメモリ(GDIR)以外、システム変数 $P_PROG、$P_PATH、$P_PROGPATH のパス名は EES 表記です。EES モードでは、これらのパス名を評価および処理するユーザープログラムは、EES 表記でのパス名も処理できるように拡張してください。

3.2 作業メモリ(CHANDATA、COMPLETE、INITIAL)

機能

作業メモリには、現在のシステム、およびコントローラの運転に使用するユーザデータ(アクティブファイルシステム)が含まれます。データの例を以下に示します。

- 動作中のマシンデータ
- 工具オフセットデータ
- ゼロオフセット
- ...

初期化プログラム

これは、作業メモリデータの初期化に使用するプログラムです。これには、次のファイルタイプを使用できます。

ファイルタイプ	説明
name_TEA	マシンデータ
name_SEA	セッティングデータ
name_TOA	工具オフセット
name_UFR	ゼロオフセット/フレーム
name_INI	初期化ファイル
name_GUD	グローバルユーザーデータ
name_RPA	R 変数

データ領域

データは、その適用先のさまざまな領域で処理できます。たとえば、コントローラは複数のチャネルを持つことができます。または、通常の場合、複数の軸を自由に配置できます。

以下のものがあります。

識別子	データ領域
NC	NC 別データ
CH<n>	チャネル別データ(<n>でチャネル名称を指定します)

識別子	データ領域
AX<n>	軸別データ(<n>で機械軸の番号を指定します)
TO	工具データ
COMPLETE	全データ

外部 PC による初期化プログラムの作成

データ領域識別子とデータタイプ識別子を使用すると、データの保存時に一つの単位として扱われる領域を特定できます。

_N_AX5_TEA_INI	5 軸のマシンデータ
_N_CH2_UFR_INI	チャネル 2 のフレーム
_N_COMPLETE_TEA_INI	すべてのマシンデータ

コントローラの最初の起動時に、コントローラが正常に動作するよう、一組のデータのセットが自動的にロードされます。

マルチチャネル制御の手順(CHANDATA)

複数のチャネル用の CHANDATA(<チャネル番号) は、_N_INITIAL_INI ファイルのみで使用できます。これは、コントローラのすべてのデータの初期化に使用するセットアップファイルです。

プログラムコード	コメント
%_N_INITIAL_INI	
CHANDATA(1)	
	; 機械軸の割り当て、チャネル 1:
$MC_AXCONF_MACHAX_USED[0]=1	
$MC_AXCONF_MACHAX_USED[1]=2	
$MC_AXCONF_MACHAX_USED[2]=3	
CHANDATA(2)	
	; 機械軸の割り当て、チャネル 2:
$MC_AXCONF_MACHAX_USED[0]=4	
$MC_AXCONF_MACHAX_USED[1]=5	
CHANDATA(1)	
	; 軸マシンデータ:
	; 汎用イグザクトストップ範囲:
$MA_STOP_LIMIT_COARSE[AX1]=0.2	; 軸 1
$MA_STOP_LIMIT_COARSE[AX2]=0.2	; 軸 2
	; 精密イグザクトストップ範囲
$MA_STOP_LIMIT_FINE[AX1]=0.01	; 軸 1

プログラムコード	コメント
$MA_STOP_LIMIT_FINE[AX1]=0.01	; 軸 2

通知

CHANDATA 命令

パートプログラムでは、CHANDATA 命令は NC プログラムが実行されるチャネルに対してのみ設定できます。つまり、この命令は NC プログラムが間違ったチャネルで実行されないよう、NC プログラムを保護するのに使用されます。

プログラムの処理は、エラーが発生すると中止されます。

注記

CHANDATA 命令は、ジョブリストの INI ファイルは含まれません。

初期化プログラムの保存(COMPLETE、INITIAL)

作業メモリのファイルは、外部 PC に保存して、そこから再度読み込むことができます。

- ファイルは、COMPLETE で保存します。
- INITIAL を使用して、すべての領域にわたる INI ファイル(_N_INITIAL_INI)を作成します。

初期化プログラムの読み込み

通知

データの消失

ファイルを「INITIAL_INI」という名称で読み込んだ場合は、ファイルで提供されないデータがすべて、標準データを使用して初期化されます。例外はマシンデータのみです。つまり、**セッティングデータ、工具データ、ZO、GUD 値、...**が標準データ(通常は「ゼロ」)で提供されます。

たとえば、ファイル COMPLETE_TEA_INI は、個々のマシンデータの読み込みに対応しています。コントローラでは、このファイルにあるのはマシンデータのみであると想定されます。このため、この場合は、他のデータ領域は影響されずにそのままです。

初期化プログラムのロード

1 つのチャネルのデータのみを使用する場合は、INI プログラムを選択し、パートプログラムとして呼び出すこともできます。これは、プログラム制御データの初期化も可能であるという意味です。

保護領域 4

4.1 プロテクションゾーンの定義(CPROTDEF、NPROTDEF)

衝突から機械要素を保護するプロテクションゾーンは、ブロック内のパートプログラムで定義します。これには、次の要素が含まれます。

1. 加工平面の定義
   実プロテクションゾーン定義の前に、プロテクションゾーンの輪郭の記述で参照する加工平面を選択する必要があります。
2. 定義の開始
   特定の NC 命令に応じて、チャネル別または機械別のプロテクションゾーンを作成します。
3. プロテクションゾーンの輪郭の記述
   プロテクションゾーンの輪郭は、移動動作によって定義されます。これらは実行されず、以前または後続のジオメトリ軸指令との関連はありません。プロテクションゾーンが定義されるだけです。
4. 定義終了

構文

```
DEF INT <Var>
G17/G18/G19
CPROTDEF/NPROTDEF(<n>,<t>,<AppLim>,<AppPlus>,<AppMinus>)
G0/G1/...X/Y/Z...
...
EXECUTE(<Var>)
```

意味

DEF INT <Var>:	ローカルヘルプ変数定義、INTEGER データタイプ
<Var>:	ヘルプ変数の名称
G17/G18/G19:	加工平面 **注:** 定義を終了する前に加工平面を変更することはできません。定義の開始から終了までの間には、垂直軸をプログラムできません。

CPROTDEF():	**チャネル**別のプロテクションゾーンを定義する事前定義された手順		
NPROTDEF():	**機械**別のプロテクションゾーンを定義する事前定義された手順		
<n>:	プロテクションゾーン定義番号		
	データタイプ:	INT	
<t>:	プロテクションゾーンのタイプ		
	データタイプ:	BOOL	
	値:	TRUE	**工具**関連プロテクションゾーン
		FALSE	**ワーク**関連プロテクションゾーン
<applim>:	第 3 次元での制限のタイプ		
	データタイプ:	INT	
	値:	0	制限なし
		1	正方向の制限
		2	負方向の制限
		3	正および負方向の制限
<AppPlus>:	3 次元での正方向の制限値		
	データタイプ:	REAL	
<appminus>:	3 次元での負方向の制限値		
	データタイプ:	REAL	

G0/G1/...X/Y/Z... ...:	プロテクションゾーンの輪郭は、選択加工平面で、最大 11 回の移動動作を行って指定します。最初の移動は、輪郭への移動です。輪郭記述の最後の点は、常に輪郭記述の最初の点に一致させてください。 以下のように、有効なプロテクションゾーンは、輪郭の左側の領域です。 ● 内側プロテクションゾーン 内側プロテクションゾーンの輪郭は、反時計方向で記述してください。 ● 外側プロテクションゾーン(ワーク対応プロテクションゾーンでのみ許可) 外側プロテクションゾーンの輪郭は、時計方向で記述してください。 次の輪郭要素が使用可能です。 ● 直線の輪郭要素のための G0、G1 ● 右回り方向の円弧部分の場合、G2 ワーク対応プロテクションゾーンでのみ使用可能です。工具対応プロテクションゾーンは凸型形状が必要なため、使用できません。 ● 左回り方向の円弧部分の場合、G3 **注:** プロテクションゾーンでは一周円を記述できません。一周円を 2 つの半円に分割してください。 **注:** シーケンス G2 → G3 または G3 → G2 は使用できません!2 つの円弧ブロックの間には、短い G1 ブロックを挿入してください。
EXECUTE(<Var>):	定義の終了を特定する事前定義された手順 EXECUTE により、通常のプログラム処理に戻ります。

例

「プロテクションゾーンの起動/解除(CPROT、NPROT) (ページ 274)」の例を参照してください。。

他の情報

機械別保護領域

機械別プロテクションゾーンやその輪郭は、ジオメトリ軸を使用して、つまり、チャネルの基本座標系(BCS)を基準にして定義されます。機械別プロテクションゾーンが有効なすべてのチャネル内で正しいプロテクションゾーン監視が行われるようにするには、対応するすべてのチャネルの基本座標系(BCS)を同じにしてください。

- 機械原点を基準にした座標原点の位置
- 座標軸の向き

輪郭の記述の基準点

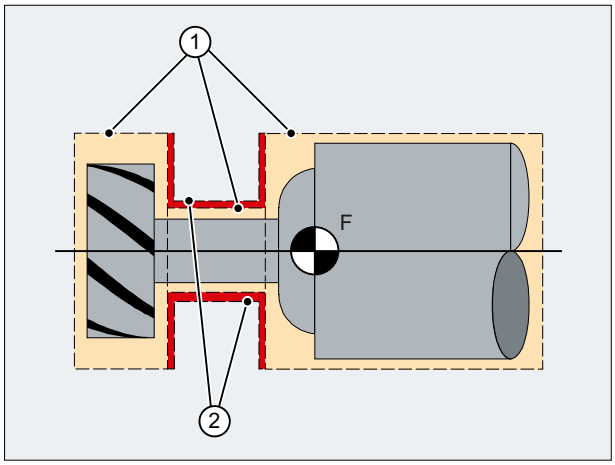
- 工具基準プロテクションゾーン
  工具基準プロテクションゾーンの座標値は、**工具ホルダレファレンス点 F** を基準とする絶対値として指定する必要があります。
- ワーク基準プロテクションゾーン
  ワーク基準プロテクションゾーンの座標値は、**基本座標系(BCS)**のゼロ点を基準とする絶対値として指定する必要があります。

回転対称なプロテクションゾーン

回転対称なプロテクションゾーン(主軸のチャックなど)の場合は、回転中心までの輪郭のみではなく、輪郭全体を記述してください。

工具基準プロテクションゾーン

工具関連プロテクションゾーンは常に凸型にしてください。凹型プロテクションゾーンが望ましい場合は、複数の凸型プロテクションゾーンに分割してください。



① 凸型プロテクションゾーン

② 凹型プロテクションゾーン(**使用できません**)

F 工具ホルダの基準点

一般条件

プロテクションゾーンの定義時には、次の機能を有効にしたり使用したりすることはできません。

- 工具径補正(フライス工具径補正、ノーズ R 補正)
- 座標変換
- レファレンス点復帰(G74)
- 固定点アプローチ(G75)
- ドウェル時間(G4)
- ブロック検索停止(STOPRE)
- プログラム終了(M17、M30)
- M 機能:M0、M1、M2

4.2 プロテクションゾーンの起動/解除(CPROT、NPROT)

以前にパートプログラムで定義したプロテクションゾーンはいつでも起動できます。また、PLC ユーザープログラムによる以降の起動に対して起動待ちにできます。有効なプロテクションゾーンはいつでも解除できます。

起動時または起動待ち時に、プロテクションゾーンのレファレンス点を相対的にシフトすることもできます。

注記

プロテクションゾーンは、起動されたチャネルのすべてのジオメトリ軸の原点確立後にのみ考慮されます。

注記

プロテクションゾーンの監視

動作中の工具関連プロテクションゾーンがない場合、工具軌跡がワーク関連プロテクションゾーンと照合されます。
ワーク方向のプロテクションゾーンが有効でない場合、プロテクションゾーンは監視されません。

構文

```
CPROT(<n>,<Status>,<XMov>,<YMov>,<ZMov>)
NPROT(<n>,<Status>,<XMov>,<YMov>,<ZMov>)
```

意味

CPROT:	**チャネル**別のプロテクションゾーンを有効にする事前定義された手順	
NPROT:	**機械**別のプロテクションゾーンを有効にする事前定義された手順	
<n>:	プロテクションゾーンの番号	
	データタイプ:	INT

<状態>:	チャネル別の起動状態は、このパラメータを使用して設定します		
	データタイプ:	INT	
	値:	0	プロテクションゾーンの解除
		1	プロテクションゾーンの起動待ち
		2	プロテクションゾーンの起動
		3	オプショナルストップありのプロテクションゾーンの起動待ち
<移動>,<y 移動>,<z 移動>:	X/Y/Z 方向の追加オフセット値 オフセットは 1、2、または 3 次元で実行できます。オフセット値は以下を示します。 ● **ワーク**関連プロテクションゾーンに対する機械原点 ● **工具**別プロテクションゾーンの工具ホルダのレファレンス点 F		
	データタイプ:	REAL	

例

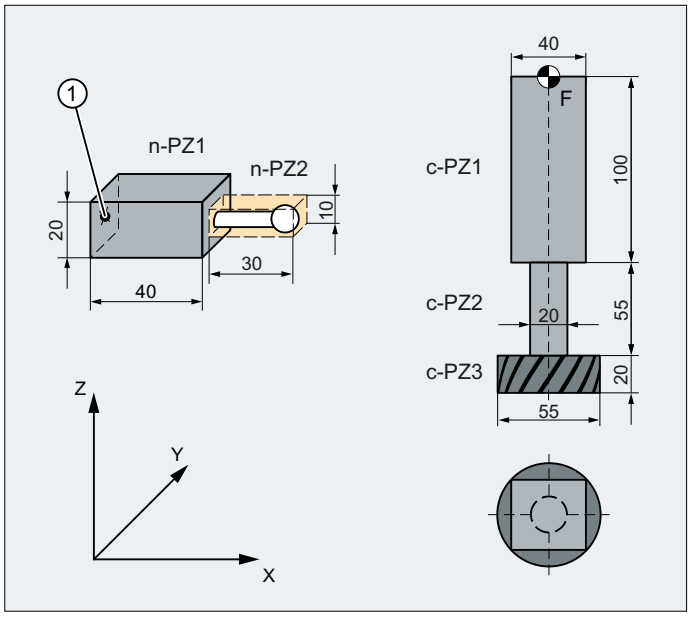
フライスカッターと計測プローブの衝突の発生の可能性を、フライス盤で監視します。機能が有効なときは、オフセットにより計測プローブの位置が定義されます。

このために、以下のプロテクションゾーンが定義されます。

- 計測プローブホルダ(n-PZ1)と計測プローブ(n-PZ2)両方の、機械別プロテクションゾーンとワーク関連プロテクションゾーン
- フライスカッターホルダ(c-PZ1)、カッターシャンク(c-PZ2)、およびフライスカッター(c-PZ3)のチャネル別プロテクションゾーンと工具関連プロテクションゾーン

すべてのプロテクションゾーンの向きは Z 方向です。

機能の起動時の計測プローブの基準点の位置は、X = -120、Y = 60、Z = 80 とします。



① プローブのプロテクションゾーンの名前

F 工具ホルダの基準点

プログラムコード	コメント
DEF INT PROTZONE	; Help 変数の定義
G17	; 加工平面 XY
; プロテクションゾーンの定義	
NPROTDEF(1,FALSE,3,10,-10)	; プロテクションゾーン n-PZ1
G01 X0 Y-10	
X40	
Y10	
X0	
Y-10	
EXECUTE(PROTZONE)	
NPROTDEF(2,FALSE,3,5,-5)	; プロテクションゾーン n-PZ2
G01 X40 Y-5	
X70	
Y5	
X40	
Y-5	
EXECUTE(PROTZONE)	
CPROTDEF(1,TRUE,3,0,-100)	; プロテクションゾーン c-PZ1
G01 X-20 Y-20	
X20	
Y20	
X-20	

プログラムコード	コメント
Y-20	
EXECUTE(PROTZONE)	
CPROTDEF(2,TRUE,3,-100,-150)	; プロテクションゾーン c-PZ2
G01 X0 Y-10	
G03 X0 Y10 J10	
X0 Y-10 J-10	
EXECUTE(PROTZONE)	
CPROTDEF(3,TRUE,3,-150,-170)	; プロテクションゾーン c-PZ3
G01 X0 Y-27.5	
G03 X0 Y27.5 J27.5	
X0 Y27.5 J-27.5	
EXECUTE(PROTZONE)	
; プロテクションゾーンの起動:	
NPROT(1,2,-120,60,80)	; オフセットを使用してプロテクションゾーン n-PZ1 を起動
NPROT(2.2,-120,60,80)	; オフセットを使用してプロテクションゾーン n-PZ2 を起動
CPROT(1,2,0,0,0)	; プロテクションゾーン c-PZ1 を起動
CPROT(2,2,0,0,0)	; プロテクションゾーン c-PZ2 を起動
CPROT(3,2,0,0,0)	; プロテクションゾーン c-PZ3 を起動

詳細情報

コントローラの電源投入後の状態の起動

コントロールシステムの電源を投入し、軸が原点確立した後に、プロテクションゾーンはすでに有効になっている場合があります。これは、プロテクションゾーンで、次のシステム変数が TRUE に設定されている場合に該当します。

- $SN_PA_ACTIV_IMMED[<n>] (機械別プロテクションゾーン)または
- $SC_PA_ACTIV_IMMED[<n>] (チャネル別プロテクションゾーン)
  インデックス"<n>"はプロテクションゾーンの数に対応します:0 = 1 プロテクションゾーン

プロテクションゾーンは状態=2、オフセットなしで起動します。

プロテクションゾーンの複数起動

機械別プロテクションゾーンは、同時に複数のチャネルで有効にできます(2 面が対向する心押し台のプロテクションゾーンなど)。プロテクションゾーンは、すべてのジオメトリ軸が原点確立済みの場合にのみ監視されます。

単一のチャネル内で 1 つのプロテクションゾーンを異なるオフセットで同時に有効にすることはできません。

工具径補正が動作中の場合のプロテクションゾーンの監視

工具径補正の動作中には、工具径補正の平面がプロテクションゾーン定義の平面と同じである場合にかぎり、プロテクションゾーンの監視機能を使用できます。

4.3 プロテクションゾーン違反、作業領域リミット、およびソフトウェアリミットスイッチのチェック(CALCPOSI)

機能

ワーク座標系(WCS)で CALCPOSI 機能は、有効なリミットに違反することなく指定した距離、開始位置から**ジオメトリ軸**を移動できるかどうかをチェックします。リミットが原因で距離を完全に移動できない場合、正の 10 進数コードの状態値と最大許容移動距離が返されます。

定義

```
INT CALCPOSI(VAR REAL[3] <Start>, VAR REAL[3] <Dist>, VAR REAL[5]
<Limit>, VAR REAL[3] <MaxDist>, BOOL <MeasSys>, INT <TestLim>)
```

構文

```
<Status> = CALCPOSI(VAR <Start>, VAR <Dist>, VAR <Limit>, VAR
<MaxDist>, <MeasSys>, <TestLim>)
```

意味

CALCPOSI(...):	ジオメトリ軸に関する限界値違反をテストするための事前定義された機能	
	先読み停止:	いいえ
	単独ブロック指令:	はい

<status>: (パート 1)	機能の戻り値。負の値は異常状態を示します。		
	データタイプ:		INT
	値の範囲:		-8 ≤ x ≤ 100000
	値:	0	距離をすべて移動できます。
		-1	少なくとも 1 つの成分が<リミット>に違反しています。
		-2	座標変換計算の異常。 例:移動距離が特異点を通過するため、軸位置を定義できません。
		-3	指定された移動距離<距離>と最大許容移動距離<最大距離>は、線形従属です。 注 <リミットのテスト>、ビット 4 == 1 と組み合わせた場合にのみ発生します。
		-4	リミット面までの<距離>に含まれている移動方向の突起がゼロベクトルであるか、移動方向が違反されたリミット面に垂直です。 注 <リミットのテスト>、ビット 5 == 1 と組み合わせた場合にのみ発生します。
		-5	<リミットのテスト>、ビット 4 == 1 およびビット 5 == 1 の場合
		-6	移動リミットのチェックのために考慮が必要な少なくとも 1 つの機械軸が、原点確立されていません。
		-7	干渉の回避機能:キネマティック結合またはプロテクションゾーンの定義が無効です。
		-8	干渉の回避機能:メモリが不足しているため、この命令を実行できません。

<status>: (パート 2)	1 の桁		
	注 複数のリミットに同時に違反している場合、指定された移動距離に最も大きな制限を課すリミットが通知されます。		
	値:	1	ソフトウェアリミットスイッチにより、移動距離が制限されています。
		2	作業領域リミットにより、移動距離が制限されています。
		3	プロテクションゾーンにより、移動距離が制限されています。
		4	干渉の回避機能:プロテクションゾーンにより、移動軌跡が制限されています。
	10 の桁		
	値:	1x	初期値が制限に違反しています
		2x	指定した直線が制限に違反しています。 終点はどの制限にも違反していないが、始点から終点までの軌跡により制限値の違反が発生する(極座標補間などの非線形座標変換に対して、WCS のソフトウェアリミットスイッチ曲線のプロテクションゾーンを通過するなど)場合でも、この値が返されます。
<status>: (パート 3)	100 の桁		
	値:	1xx	および 1 の桁== 1 または 2 正の制限値に違反しています。
			および 1 の桁== 3 [1)] NC 別のプロテクションゾーンに違反しています。
		2xx	および 1 の桁== 1 または 2 負の制限値に違反しています。
			および 1 の桁== 3 [1)] チャネル別プロテクションゾーンに違反しています。

<status>: (パート 4)	1000 の桁		
	値:	1xxx	および 1 の桁== 1 または 2 軸番号に乗算する係数がリミットに違反しています。軸のナンバリングは 1 から始まります。 基準 ● ソフトウェアリミットスイッチ:機械軸 ● 作業領域リミット:ジオメトリ軸 および 1 の桁== 3 1) 違反されたプロテクションゾーンの数に乗算する係数。
<status>: (パート 5)	100000 の桁		
	値:	0xxxxx	100000 の桁 == 0: <距離>はそのまま変更されません。
		1xxxxx	方向ベクトルが、リミット面での詳細な移動方向を定義する<距離>で返されます。 以下の補足条件でのみ発生します。 ● ソフトウェアリミットスイッチまたは作業領域リミットに違反した(始点以外で)。 ● 座標変換が有効で**ない**。 ● <TestID>、ビット 4 またはビット 5 == 1
<開始>:	開始位置のベクトルの参照 ● <開始> [0]:1 番目のジオメトリ軸 ● <開始> [1]:2 番目のジオメトリ軸 ● <開始> [2]:3 番目のジオメトリ軸		
	パラメータタイプ:	入力	
	データタイプ:	VAR REAL [3]	
	値の範囲:	-最大 REAL 値 ≤ x[<n>] ≤ +最大 REAL 値	

<距離>:	ベクトルの参照。	
	入力:インクレメンタル移動距離 ● <距離> [0]:1 番目のジオメトリ軸 ● <距離> [1]:2 番目のジオメトリ軸 ● <距離> [2]:3 番目のジオメトリ軸	
	出力(<状態>で 100000 桁を設定している場合のみ)	
	<距離>には、出力値として単位ベクトル **v** が含まれ、これが WCS 内での詳細な移動方向を定義します。	
	ケース 1:<TestID>、ビット 4 == 1 の場合のベクトル **v** の形成 入力ベクトル<距離>と<最大距離>は移動平面にかかります。この平面は、違反されたリミット面によって切断されます。2 つの平面の交線によって、ベクトル **v** の方向が定義されます。方向(符号)は、入力ベクトル<最大距離>と **v** の間の角度が 90°を超えないよう選択されます。	
	ケース 2:<TestID>、ビット 5 == 1 の場合のベクトル **v** の形成 ベクトル **v** は、リミット面上の<距離>に含まれる移動ベクトルの突起方向の単位ベクトルです。リミット面上の移動ベクトルの突起がゼロベクトルの場合、アラームが返されます。	
	パラメータタイプ:	入力/出力
	データタイプ:	VAR REAL [3]
	値の範囲:	-最大 REAL 値 ≤ x[<n>] ≤ +最大 REAL 値

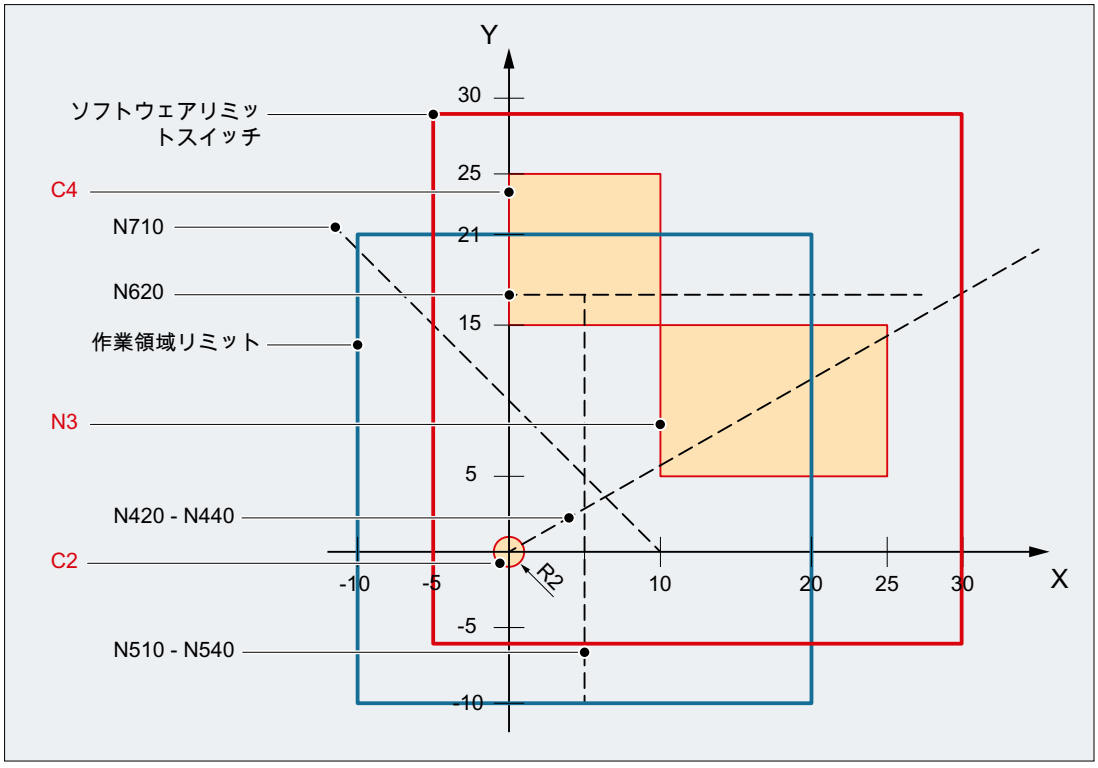
<リミット>:	長さ 5 の配列の参照 ● <リミット> [0 - 2]:リミットまでのジオメトリ軸の最小スペース – <リミット> [0]:1 番目のジオメトリ軸 – <リミット> [1]:2 番目のジオメトリ軸 – <リミット> [2]:3 番目のジオメトリ軸 最小スペースは、次のもので遵守されます。 – 作業領域リミット:制限なし – ソフトウェアリミットスイッチ:座標変換が有効でない場合や、5 軸座標変換など、直線機械軸へのジオメトリ軸の明確な割り当てが可能な座標変換が有効な場合。 ● <リミット> [3]:たとえば、非線形座標変換のためにジオメトリ軸を割り当てられない直線機械軸の最小スペースが含まれています。この値は、従来のプロテクションゾーンや干渉回避プロテクションゾーンの監視用の制限値にも使用されます。 ● <リミット> [4]:たとえば、非線形座標変換のためにジオメトリ軸を割り当てられない回転機械軸の最小スペースが含まれています。 **注** この値は、特殊な座標変換用のソフトウェアリミットスイッチの監視にだけ有効です。	
	パラメータタイプ:	入力
	データタイプ:	VAR REAL [5]
	値の範囲:	-最大 REAL 値 ≤ x[n] ≤ +最大 REAL 値

<最大距離>:	軸リミットの指定された最小スペースが、関連する機械軸のどれにも違反していないインクレメンタル移動距離のベクトルを参照。 ● **<距離>** [0]:1 番目のジオメトリ軸 ● **<距離>** [1]:2 番目のジオメトリ軸 ● **<距離>** [2]:3 番目のジオメトリ軸 移動距離に制限がない場合、この戻りパラメータの内容は**<距離>**の内容と同じです。		
	<テスト ID>、ビット 4 == 1 の場合: **<距離>**と**<最大距離>** **<最大距離>**と**<距離>**には、移動平面にかかるベクトルを入力値として含めてください。2 つのベクトルは、互いに線形従属しないものとします。**<最大距離>**の絶対値は任意です。移動方向の計算については、**<距離>**の説明を参照してください。		
	パラメータタイプ:		出力
	データタイプ:		VAR REAL [3]
	値の範囲:		-最大 REAL 値 ≤ x[<n>] ≤ +最大 REAL 値
<MeasSys>:	位置と距離の指定のための単位系(インチ/メトリック)(**オプション**)		
	データタイプ:		BOOL
	値:	FALSE (デフォルト)	G グループ 13(G70、G71、G700、G710)の現在有効な G 命令に対応する単位系です。 **注** G70 が有効で基本単位系がメトリックの場合(または、G71 が有効で基本単位系がインチの場合)、システム変数$AA_IW と$AA_MW は基本単位系で提供され、使用する場合、CALCPOSI 用に変換してください。
		TRUE	基本単位系に準拠する単位系: MD52806 $MN_ISO_SCALING_SYSTEM

<リミットのテスト>:	監視するリミットのビット指定選択(**オプション**)		
	データタイプ:		INT
	初期値:		ビット 0、1、2、3、6、7 == 1 (207)
	ビット	**10 進数**	**意味**
	0	1	ソフトウェアリミットスイッチ
	1	2	作業領域リミット
	2	4	起動した従来のプロテクションゾーン
	3	8	起動待ちの従来のプロテクションゾーン
	4	16	**<**距離**>**のソフトウェアリミットスイッチまたは作業領域リミットに違反した場合、**ケース 1** に示すような移動方向が返されます(上記参照)。
	5	32	**<**距離**>**のソフトウェアリミットスイッチまたは作業領域リミットに違反した場合、**ケース 2** に示すような移動方向が返されます(上記参照)。
	6	64	起動された干渉回避プロテクションゾーン
	7	128	起動待ちの干渉回避プロテクションゾーン
	8	256	起動された干渉回避プロテクションゾーンと起動待ちの干渉回避プロテクションゾーンのペア
[1] 複数のプロテクションゾーンに同時に違反している場合、指定された移動距離に最も大きな制限を課すプロテクションゾーンが返されます。			

例

リミット位置



この例では、X-Y 平面内の有効なソフトウェアリミットスイッチおよび作業領域リミットと、次の 3 つのプロテクションゾーンが表示されています。

- C2:工具関連のチャネル別プロテクションゾーン、有効、円形、半径= 2 mm
- C4:ワーク関連のチャネル別プロテクションゾーン、起動待ち、正方形、辺長= 10 mm
- N3:機械別のプロテクションゾーン、有効、長方形、辺長= 10 mm x 15 mm

NC プログラム

NC プログラムでは最初にプロテクションゾーンと作業領域リミットが定義されています。その後、さまざまなパラメータ設定で CALCPOSI () 機能が呼び出されます。

プログラムコード

```
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
```

4.3 プロテクションゾーン違反、作業領域リミット、およびソフトウェアリミットスイッチのチェック (CALCPOSI)

プログラムコード

```
:工具関連プロテクションゾーン C2
N70 CPROTDEF(2, TRUE, 0)
N80 G17 G1 X-2 Y0
N90 G3 I2 X2
N100 I-2 X-2
N110 EXECUTE(_PA)
; ワーク関連プロテクションゾーン C4
N120 CPROTDEF(4, FALSE, 0)
N130 G17 G1 X0 Y15
N140 X10
N150 Y25
N160 X0
N170 Y15
N180 EXECUTE(_PA)
; 機械別プロテクションゾーン N3
N190 NPROTDEF(3, FALSE, 0)
N200 G17 G1 X10 Y5
N210 X25
N220 Y15
N230 X10
N240 Y5
N250 EXECUTE(_PA)
; プロテクションゾーンを起動または起動待ちにします
N260 CPROT(2, 2, 0, 0, 0)
N270 CPROT(4, 1, 0, 0, 0)
N280 NPROT(3, 2, 0, 0, 0)
; 作業領域リミットを定義します
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.
```

プログラムコード

```
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.
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)
; N690 および N700 からフレームを再度解除します
N720 TRANS
N730 _START[0] =  0.
N740 _START[1] = 10.
N750 _START[2] =  0.
; ベクトル _DIST および _MAXDIST が移動平面を定義します
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
```

CALCPOSI()の結果

N...	<状態>	<最大距離>[0] ≙ X	<最大距離>[1] ≙ Y	備考
420	3123	8.040	4.594	N3 に違反しています。
430	1122	20.000	11.429	保護領域が監視されず、ワーキングエリアリミットに違反しています。
440	1121	30.000	17.143	ソフトウェアリミット監視のみが依然として有効です。
510	4213	0.000	0.000	起点が C4 に違反しています。
520	0000	0.000	-27.000	起動待ちの C4 が監視されていません。指定された距離をすべて移動できます。
540	2222	0.000	-25.000	_LIMIT[1] = 2 のために、移動距離が作業領域リミットによって制限されます。
620	4223	-13.000	0.000	C4 までのスペースは、C2 と _LIMIT[3]により、合計 4 mm です。C2 → N3 の 0.1 mm のスペースのために移動距離が制限されることはありません。
710	1221	0.000	21.213	平行移動と回転のあるフレームが有効です。_DIST で許容される移動距離は、平行移動および回転した WCS に適用されます。
820	102121	18.000	18.000	Y 軸のソフトウェアリミットスイッチに違反しています。詳細な移動方向の計算が<_TESTLIM> = 17 で要求されます。この方向は_DIST (0.707, 0.0, 0.707)です。これは、<_STATUS>で 100000 桁が設定されているため、有効です。

他の情報

「原点確立された」軸の状態

CALCPOSI()によって考慮されるすべての機械軸を原点確立してください。

円弧関連距離の指定

すべての円弧関連距離の指定は、**常に**半径指定として解釈されます。このことは、特に直径指定(DIAMON/DIAM90)が有効になっている径方向軸の場合に考慮してください。

移動距離の縮小

ある軸の指定された移動距離が制限されている場合は、それに比例して他の軸の移動距離も<最大距離>の戻り値で縮小されます。したがって、得られる終点はまだ指定された軌跡上にあります。

回転軸

回転軸は、モジュロ回転軸でない場合にのみ監視されます。

ソフトウェアリミットスイッチ、作業領域リミットおよびプロテクションゾーンのいずれも、関連する複数の軸に対して定義しないこともできます。

ソフトウェアリミットスイッチおよび作業領域リミットの状態

ソフトウェアリミットスイッチと作業領域リミットは、CALCPOSI()の実行中に有効である場合にのみ考慮されます。状態は、たとえば、次のものの影響を受けることがあります。

- マシンデータ:MD21020 $MC_WORKAREA_WITH_TOOL_RADIUS
- セッティングデータ: $AC_WORKAREA_CS_...
- NC/PLC インタフェース信号 DB31、... DBX12.2 / 3
- 命令:WALIMON / WALIMOF

ソフトウェアリミットスイッチと座標変換

CALCPOSI()では、さまざまなキネマティックトランスフォーメーション(TRANSMIT など)の場合、移動距離の特定の位置であいまいさが生じるため、ジオメトリ軸(WCS)の位置から機械軸(MCS)の位置を明確に特定できるとはかぎりません。通常の移動動作では、履歴、および WCS の連続移動が MCS の連続移動に対応する、という条件から一義的にならないのが一般的です。したがって、ソフトウェアリミットスイッチを監視するとき

は、このような場合のあいまいさを解決するために CALCPOSI() 実行時の機械位置が使用されます。

注記

先読み停止

座標変換と組み合わせて CALCPOSI() を使用する場合、機械軸位置の同期のために、CALCPOSI() の前に先読み停止(STOPRE)を先読みと共にプログラムすることは、使用者の責任です。

プロテクションゾーンのスペースと従来のプロテクションゾーン

従来のプロテクションゾーンでは、パラメータ<リミット>[3]で設定した安全距離が、指定された軌跡上の移動動作中、すべてのプロテクションゾーンについて保持されるという保証は**ありません**。保証されるのは、<距離>で返される終点が移動方向に安全距離だけ延長されたときに、プロテクションゾーンに違反しないということだけです。ただし、直線がプロテクションゾーンのすぐ近くを通過することがあります。

プロテクションゾーンのスペースと干渉回避プロテクションゾーン

干渉回避プロテクションゾーンでは、パラメータ<リミット>[3]で設定した安全距離が、指定された軌跡上の移動動作中、すべてのプロテクションゾーンについて保持されるという保証があります。

パラメータ<リミット>[3]で指定された安全距離は、以下が適用される場合にのみ有効になります。

<リミット>[3] **>** (MD10619 $MN_COLLISION_TOLERANCE)

ビット **4** がパラメータ<TestLim>で設定されている場合にのみ(実行中の移動方向の計算)、数十万の数値が機能の戻り値(<status>)で設定されているときに、<DIST>で受信した方向ベクトルは有効です。プロテクションゾーンに違反があったため、または座標変換が有効であるために、このような方向を決定できない場合、<DIST>の入力値は変更されないままです。追加エラーメッセージは出力されません。

特殊動作命令 5

5.1 符号化位置へのアプローチ(CAC、CIC、CDC、CACP、CACN)

次の命令で、直線軸と回転軸を、マシンデータの表に保存した固定の軸位置に、位置番号を使用して移動できます。このプログラミングタイプを「符号化位置へのアプローチ」と呼びます。

構文

```
CAC(<n>)
CIC(<n>)
CACP(<n>)
CACN(<n>)
```

意味

CAC(<n>):	位置番号 n からの符号化位置へのアプローチ
CIC(<n>):	実位置番号から前方(+n)または後方(–n)へ、符号化位置 n の場所へアプローチします
CDC(<n>):	位置番号 n から最短軌跡に沿って指定位置へアプローチします (回転軸の場合のみ)
CACP(<n>):	位置番号 n から正の方向に符号化位置へアプローチします(回転軸の場合のみ)
CACN(<n>):	位置番号 n から負の方向に符号化位置へアプローチします(回転軸の場合のみ)
<n>:	マシンデータ表内の位置番号 値の範囲:0, 1, … (表の位置の最大番号- 1)

例:位置決め軸の符号化位置へのアプローチ

プログラミングコード	コメント
N10 FA[B]=300	; 位置決め軸 B の送り速度
N20 POS[B]=CAC(10)	; 位置番号 10 の符号化位置へアプローチ
N30 POS[B]=CIC(-4)	; 「現在の位置番号」 – 4 の符号化位置へアプローチ

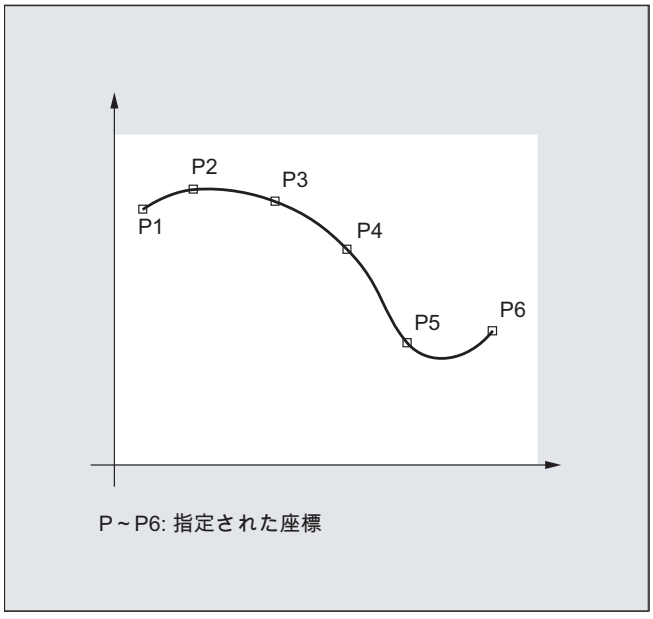
参照先

- 機能マニュアル 上級機能; 割り出し軸(T1)
- 機能マニュアル シンクロナイズドアクション

5.2 スプライン補間(ASPLINE、BSPLINE、CSPLINE、BAUTO、BNAT、BTAN、EAUTO、ENAT、ETAN、PW、SD、PL)

不規則な曲線のワーク輪郭は、解析形式で正確には定義できません。このため、表面を数値化するときなどは、これらのタイプの輪郭として、曲線上の一定の数の点を使用した近似値が求められます。数値化されたワークの表面を生成するには、曲線上の点を接続して輪郭を定義します。このために、スプライン補間を使用できます。

スプラインは、2 次または 3 次多項式で形成される曲線を定義します。スプラインの曲線上の点の特性は、**使用するスプラインタイプ**に応じて定義できます。



SINUMERIK ソリューションラインの場合は、次のスプラインタイプを使用できます。

- A スプライン
- B スプライン
- C スプライン

構文

一般:

```
ASPLINE X...Y...Z...A ...B ...C...
BSPLINE X...Y...Z...A ...B ...C...
CSPLINE X...Y...Z...A ...B ...C...
```

B スプラインでは、以下をさらにプログラム指令できます。

```
PW=<n>
SD=2
```

PL=<値>

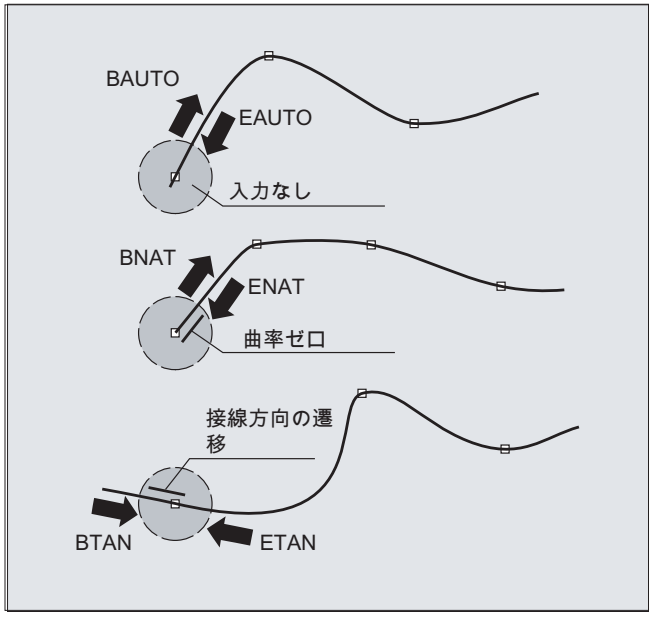
A と C スプラインでは、以下をさらにプログラム指令できます。

BAUTO / BNAT / BTAN

EAUTO / ENAT / ETAN

意味

スプライン補間タイプ:			
ASPLINE:	A スプライン補間を起動する命令		
BSPLINE:	B スプライン補間を起動する命令		
CSPLINE:	C スプライン補間を起動する命令		
	ASPLINE、BSPLINE、および CSPLINE 命令はモーダルに有効で、動作命令のグループに属します。		
曲線上の点とチェック点:			
X...Y...Z... A...B ...C...	直交座標の位置		
点の重み(B スプラインのみ):			
PW:	PW 命令を使用すると、曲線上の各点に「点の重み」をプログラム指令できます。		
<n>:	「点の重み」		
	値の範囲:	0 ≤ n ≤ 3	
	ステップ値:	0.0001	
	動作:	n > 1	曲線がチェック点に大きく引き付けられます。
		n < 1	曲線はチェック点にそれほど引き付けられません。
スプラインの次数(B スプラインのみ):			
SD:	3 次多角形を標準として使用しますが、SD=2 をプログラム指令すると、2 次多角形も使用できます。		
ノード間の距離(B スプラインのみ):			

PL:	ノード間の距離を、内部で適切に計算します。制御装置は PL 命令を使用して、いわゆるパラメータ区間の長さで指定した予約ノードスペースも加工できます。	
<値>:	パラメータ区間の長さ	
	値の範囲:	軌跡寸法と同じ
スプライン曲線の開始の遷移動作(A または C スプラインのみ):		
BAUTO:	遷移動作は指定されていません。開始を特定するのは、1 番目の点の位置です。	
BNAT:	曲率ゼロ	
BTAN:	前のブロックの接線方向の遷移(解除位置)	
スプライン曲線の終わりの遷移動作(A または C スプラインのみ):		
EAUTO:	遷移動作は指定されていません。終わりを特定するのは、最後の点の位置です。	
ENAT:	曲率ゼロ	
ETAN:	前のブロックの接線方向の遷移(解除位置)	
		

注記

プログラム指令可能な遷移動作は、B スプラインには影響しません。B スプラインは、その始点と終点で常に、指令多角形に接します。

必要条件

- 工具径補正を使用できます。
- 衝突監視が、投影された平面で実行されます。

例

例 1:B スプライン

プログラムコード 1 (重みはすべて 1 です)

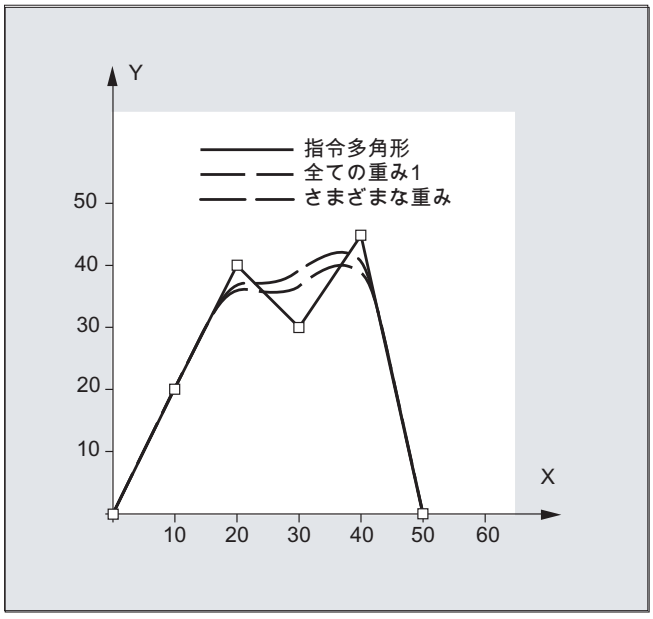
```
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
N60 X40 Y45
N70 X50 Y0
```

プログラムコード 2 (さまざまな重みがあります)

```
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
```

プログラムコード 3 (指令多角形)	コメント
N10 G1 X0 Y0 F300 G64	
N20	; 該当なし
N30 X10 Y20	
N40 X20 Y40	
N50 X30 Y30	
N60 X40 Y45	

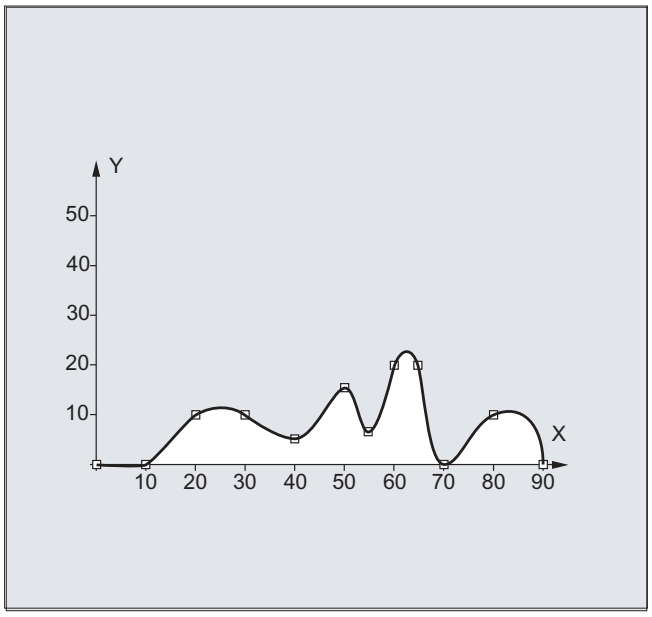
プログラムコード 3 (指令多角形)	コメント
N70 X50 Y0	



例 2 :C スプライン、始まりと終わりでの曲率ゼロ

プログラムコード

```
N10 G1 X0 Y0 F300
N15 X10
N20 BNAT ENAT
N30 CSPLINE X20 Y10
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
```



例 3 :スプライン補間(A スプライン)と座標変換(ROT)

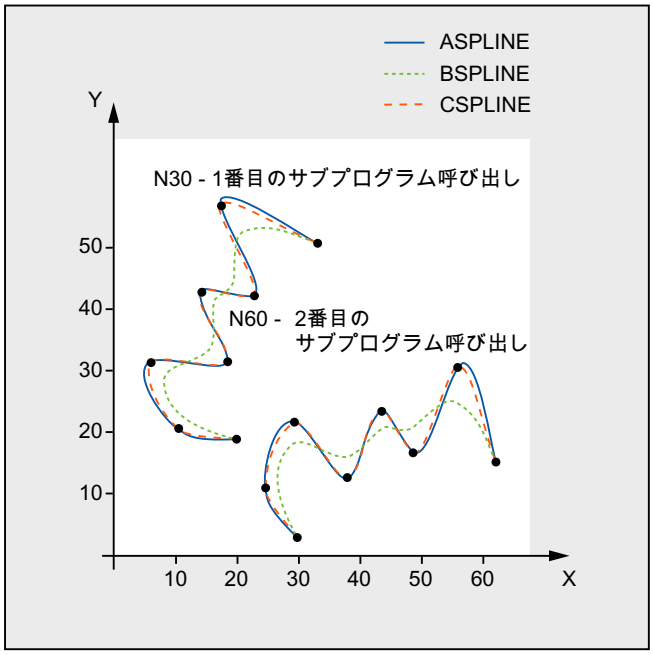
メインプログラム

プログラムコード	コメント
N10 G00 X20 Y18 F300 G64	; 起点にアプローチします。
N20 ASPLINE	; 補間タイプの A スプラインを有効にします。
N30 CONTOUR	; 1 番目のサブプログラムを呼び出します。
N40 ROT Z-45	; 座標変換:Z 軸を中心として-45°の WCS を回転します。
N50 G00 X20 Y18	; 輪郭の起点へアプローチします。
N60 CONTOUR	; 2 番目のサブプログラムを呼び出します。
N70 M30	; プログラム終了

サブプログラム「contour」(曲線上の点の座標を含む):

プログラムコード
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

上記の例のプログラム(ASPLINE)で得られたスプライン曲線に加え、以下の図には、BまたはC スプライン補間(BSPLINE、CSPLINE)を起動すると得られるスプライン曲線も含まれます。



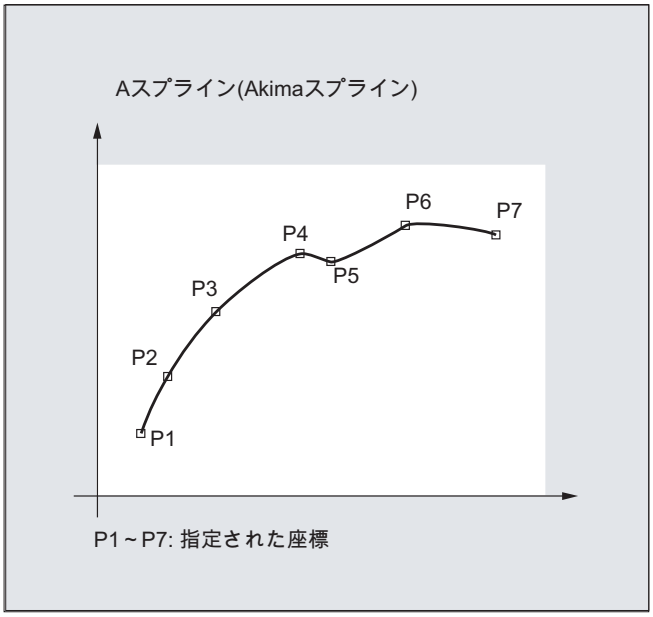
詳細情報

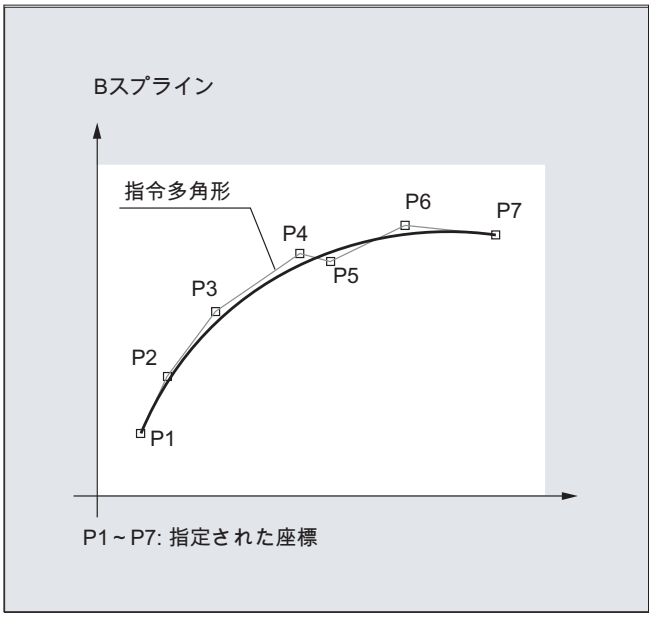
スプライン補間の長所

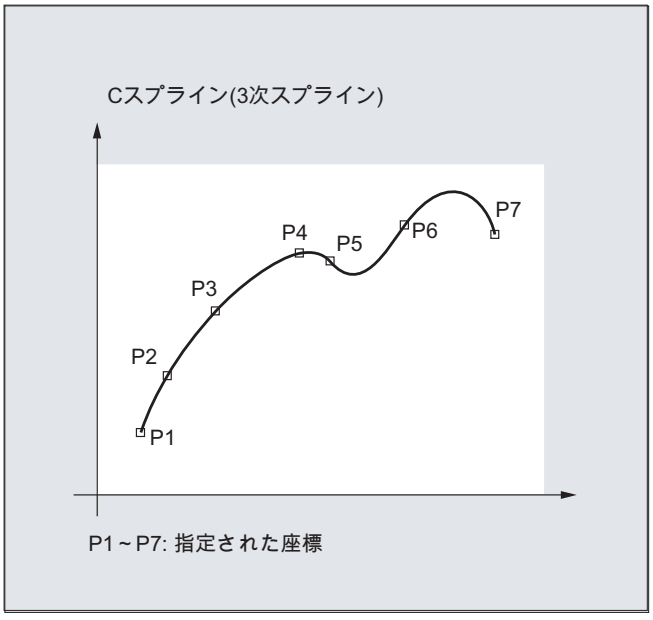
スプライン補間を使用すると、直線ブロック G01 を使用した場合とは対照的に、次の長所が得られます。

- より少ないパートプログラムブロックの数で輪郭が記述できます。
- なめらかな曲線特性により、パートプログラムブロック間遷移で、機械システムへの負担が軽減されます。

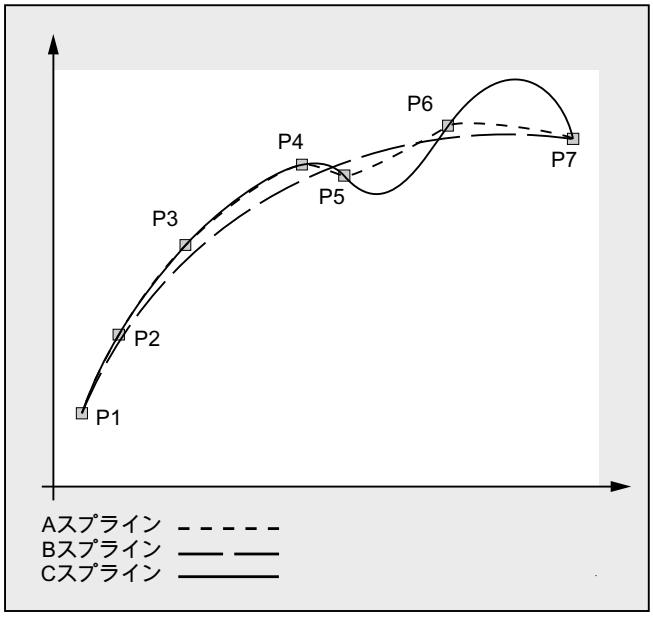
さまざまなスプラインタイプの機能と用途

スプラインタイプ	機能と使用方法
A スプライン	 **特性:** ● 曲線上の指定した中間点を正確に通過します。 ● 曲線特性は接線方向ですが、曲率は連続しません。 ● 不必要な振動はほとんど発生しません。 ● 曲線上の中間点の変化が影響する範囲は局部的です。つまり、曲線上の中間点の変化が影響するのは、隣接する 6 個までの中間点のみです。 **用途:** A スプラインは特に、大きく変化する曲線(階段状タイプの曲線特性など)の補間に適しています。

スプラインタイプ	機能と使用方法
B スプライン	 **特性:** ● 曲線上の指定した中間点は通過しませんが、その近くを通過します。曲線は中間点に引きつけられません。曲線特性はさらに、係数による中間点の重み付けにより影響されます。 ● 曲線特性は、曲率は連続して接線方向です。 ● 不必要な振動はまったく発生しません。 ● 曲線上の中間点の変化が影響する範囲は局部的です。つまり、曲線上の中間点の変化が影響するのは、隣接する 6 個までの中間点のみです。 **用途:** B スプラインの主な用途は、CAD システムとのインタフェースです。

スプラインタイプ	機能と使用方法
C スプライン	 **特性:** ● 曲線上の指定した中間点を正確に通過します。 ● 曲線特性は、曲率は連続して接線方向です。 ● 不必要な振動が、特に曲線が大きく変化する位置でよく発生します。 ● 中間点の変化が影響する範囲は全体にわたります。つまり、中間点が変化した場合は、それが曲線特性全体に影響します。 **用途:** C スプラインは、解析で定義される曲線(円、放物線、双曲線)上に中間点がある場合の使用に適しています。

同じ補間点をもつ 3 つのスプラインタイプの比較演算



最小スプラインブロックの数

G コード ASPLINE、BSPLINE、および CSPLINE は、ブロック終点をスプラインでリンクします。このために、一連のブロック(終点)は同時に計算されます。計算のバッファサイズは、標準では 10 ブロックです。すべてのブロック情報がスプライン終点となるわけではありません。ただし、制御装置には、10 ブロックごとに特定の数のスプライン終点ブロックが必要です。

スプラインタイプ	スプラインブロックの最小数
A スプライン:	各 10 ブロックのうち、**4** ブロック以上をスプラインブロックにしてください。 これには、コメントブロックやパラメータ計算は含まれません。
B スプライン:	各 10 ブロックのうち、**6** ブロック以上をスプラインブロックにしてください。 これには、コメントブロックやパラメータ計算は含まれません。
C スプライン:	必要な最小スプラインブロックの数は、以下の合計です。 MD20160 $MC_CUBIC_SPLINE_BLOCKS の値+ 1 スプラインセグメントを計算する点の数を MD20160 に入力します。初期設定は 8%です。各 10 ブロックのうち、**9** ブロック以上をスプラインブロックにしてください。

注記

許容値を下回ると同時に、スプラインで使用する軸のいずれかを位置決め軸としてプログラム指令している場合は、アラームが発生します。

短いスプラインブロックの結合

スプライン補間に短いスプラインブロックが発生し、軌跡速度が必要以上に減速するおそれがあります。「短いスプラインブロックの結合」機能を使用すると、これらのブロックが十分な長さになるように結合して、軌跡速度の減速を防止できます。

この機能は、次のチャネル別マシンデータで有効にします。

MD20488 $MC_SPLINE_MODE (スプライン補間の設定)

参照先:

機能マニュアル 基本機能; 連続軌跡モード、イグザクトストップ、先読み (B1)、
章:短いスプラインブロックの結合

5.3 スプラインングループ(SPLINEPATH)

スプラインングループで補間する軸を、SPLINEPATH 命令で選択します。スプライン補間のグループ化に使用できるのは、8 つまでの軌跡軸です。

注記

SPLINEPATH を明示的にプログラム指令していない場合は、チャネルの最初の 3 軸がスプラインングループとして移動します。

構文

スプラインングループは個別のブロックで定義します。

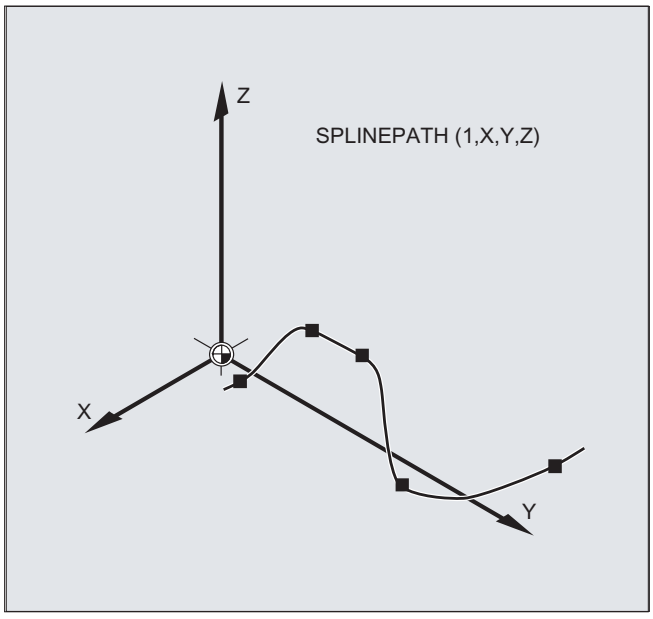
```
SPLINEPATH(n,X,Y,Z,…)
```

意味

SPLINEPATH:	スプラインングループを定義する命令
n:	=1 (固定値)
X,Y,Z,… :	スプラインングループで補間する軌跡軸の識別子

例:3 つの軌跡軸によるスプラインングループ

プログラムコード	コメント
N10 G1 X10 Y20 Z30 A40 B50 F350	
N11 SPLINEPATH(1,X,Y,Z)	; スプラインングループ
N13 CSPLINE BAUTO EAUTO X20 Y30 Z40 A50 B60	; C スプライン:
N14 X30 Y40 Z50 A60 B70	; 中間点
…	
N100 G1 X… Y…	; スプライン補間を選択解除


Z
SPLINEPATH (1,X,Y,Z)
X
Y

5.4 NC ブロック圧縮(COMPON、COMPCURV、COMPCAD、COMPSURF、COMPOF)の有効化/無効化

直線ブロック(さらにパラメータ設定に応じて、円弧、早送りブロック)を圧縮する機能は、G グループ 30 の G 命令を使用して起動/解除します。これらの命令はモーダルです。

構文

```
COMPON / COMPCURV / COMPCAD / COMPSURF
...
COMPOF
```

意味

COMPON:	コンプレッサ機能 COMPON を有効にします
COMPCURV:	コンプレッサ機能 COMPCURV を有効にします
COMPCAD:	コンプレッサ機能 COMPCAD を有効にします
COMPSURF:	コンプレッサ機能 COMPSURF を有効にします
COMPOF ：	現在動作中のコンプレッサ機能を解除します。

注記

丸み付け機能 G642 と加々速度制限 SOFT を使用して、加工面品質をさらに向上させることができます。これらの命令は、プログラムの先頭に記述してください。

例:COMPCAD

プログラムコード	コメント
N10 G00 X30 Y6 Z40	
N20 G1 F10000 G642	; 起動:丸み付け機能 G642
N30 SOFT	; 起動:加々速度制限 SOFT
N40 **COMPCAD**	; 起動:コンプレッサ機能 COMPCAD
N50 STOPFIFO	
N24050 Z32.499	; 1 番目の移動ブロック
N24051 X41.365 Z32.500	; 2 番目の移動ブロック
...	
N99999 X...Z...	; 最後の移動ブロック
COMPOF	; コンプレッサ機能をオフ

プログラムコード	コメント
...	

5.5 多項式補間(POLY、POLYPATH、PO、PL)

実際には多項式補間(POLY)が使用され、スプライン補間タイプは使用されません。その主な目的は、外部で生成され、しかもスプライン区間を直接記述できるスプライン曲線をプログラム指令するための、インタフェースとなることです。

この補間モードを使用すると、NC が多項式係数の計算処理から解放されます。これは、CAD システムまたはポストプロセッサから直接、係数が供給される場合に適切に使用できます。

構文

3 次多項式:

```
POLY PO[X]=(xe,a2,a3) PO[Y]=(ye,b2,b3) PO[Z]=(ze,c2,c3) PL=n
```

5 次多項式と新しい多項式の構文:

```
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")
```

注記

NC ブロックでプログラム指令された多項式係数と軸の合計が、ブロック毎に許可される軸の最大数を超えないようにしてください。

意味

POLY：	POLY を含むブロックによる多項式補間を適用します。
POLYPATH：	AXIS と VECT の両方の軸グループに対して多項式補間を選択可能です。
PO[軸識別子/変数]：	終点と多項式係数
X, Y, Z：	軸識別子
xe, ye, ze ：	特定の軸の終点の指定。軌跡寸法としての数値範囲

a2, a3, a4, a5：	係数 a_2、a_3、a_4、および a_5 は軌跡寸法としての数値範囲の値で記述されます。それぞれの場合の最後の係数がゼロの場合は、その係数を省略できます。
PL:	多項式が定義されたパラメータ区間の長さ(関数 f(p)の定義範囲)です。 この区間は常に 0 で始まり、p には 0 ～ PL の値を入れることができます。 PL の理論的な数値の範囲: 0.0001 ～ 99 999.9999 **注:** PL の値は、それが配置されているブロックに適用されます。PL をプログラム指令しない場合は、PL=1 が適用されます。

多項式補間の有効化/無効化

多項式補間は、パートプログラムで POLX G 命令を使用して有効にします。

POLY G 命令は、G0、G1、G2、G3、ASPLINE、BSPLINE 、および CSPLINE と共に第 1 グループに属します。

名称と終点のみでプログラム指令された軸(X10 など)は、直線で移動します。NC ブロックの軸をすべて、このようにプログラム指令した場合は、コントローラは G1 の場合と同様に動作します。

多項式補間は、第 1 G グループの別の命令(G0、G1)をプログラム指令すると、再度自動的に無効化されます。

多項式係数

PO 値(PO[]=)または...=PO(...) は、軸のすべての多項式係数を指定します。多項式の次数に対応する複数の値を、コンマで区切って指定します。1 ブロック内で、さまざまな軸にさまざまな次数の多項式を指定できます。

POLYPATH サブプログラム

POLYPATH(...)を使用すると、次のように、特定の軸グループに対して、多項式補間を選択して解放できます。

軌跡軸と追加軸のみ:	POLYPATH("AXES")
旋回軸のみ: (方向座標変換での移動時)	POLYPATH("VECT")

解放されない軸は直線で移動します。

多項式補間は両方の軸グループに対して、標準で有効になっています。

多項式補間は、POLYPATH( )がパラメータなしでプログラム指令された場合、すべての軸に対して無効です。

例

プログラムコード	コメント
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	; 多項式補間をオン
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	; 軸の混合データ
N27 PO[X]=(10,2.5) PO[Y]=(2.3)	; PL のプログラム指令なし、PL=1 を適用
N30 G1 X… Y… Z.	; 多項式補間をオフ
…	

例:新しい多項式構文

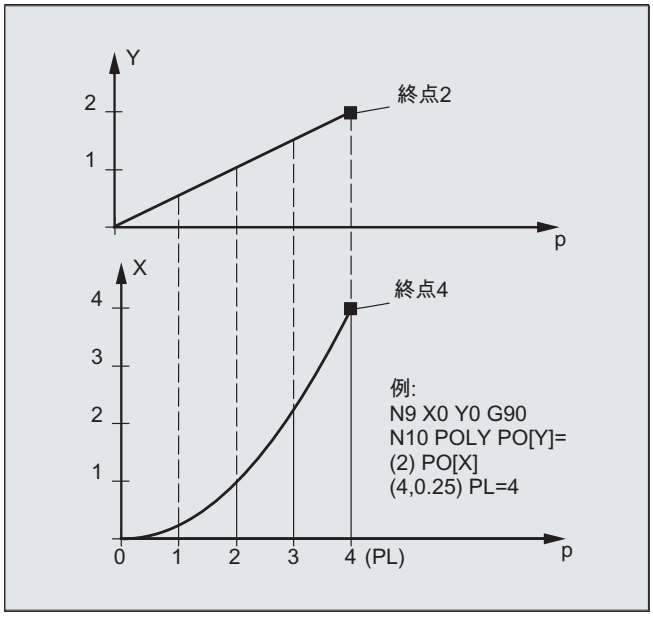
従来の多項式構文	新しい多項式構文
PO[軸識別子]=(.. , ..)	軸識別子=PO(.. , ..)
PO[PHI]=(.. , ..)	PHI=PO(.. , ..)
PO[PSI]=(.. , ..)	PSI=PO(.. , ..)
PO[THT]=(.. , ..)	THT=PO(.. , ..)
PO[]=(.. , ..)	PO(.. , ..)
PO[変数]=IC(.. , ..)	変数=PO IC(.. , ..)

例:X/Y 平面の曲線。

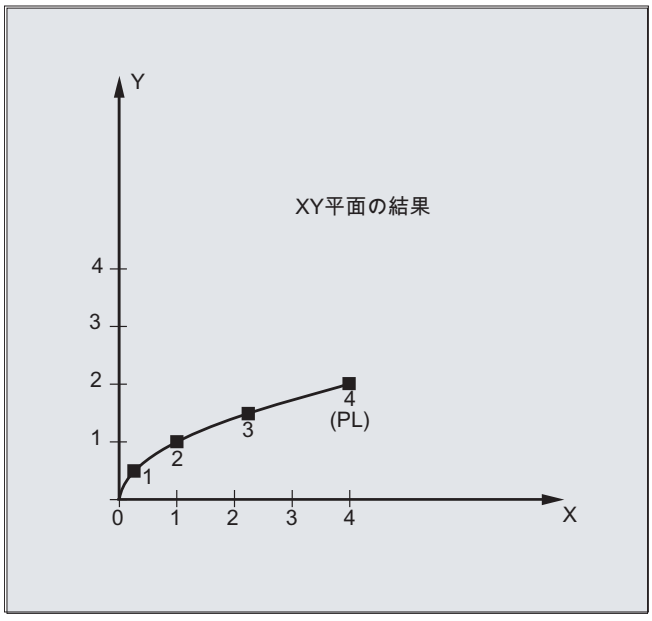
プログラミング

プログラムコード
N9 X0 Y0 G90 F100
N10 POLY PO[Y]=(2) PO[X]=(4,0.25) PL=4

曲線の形状、X(p)と Y(p)



XY 平面の曲線の形状



説明

多項式関数は通常、次の等式で表わします。

$f(p)= a_0 + a_1p + a_2p^2 + . . . + a_np^n$

意味: a_i:定数係数(i = 0, 1, ..., n)
p:パラメータ

コントローラでは、次のように、最大 5 次までの多項式をプログラム指令できます。

$f(p)= a_0 + a_1p + a_2p^2 + a_3p^3 + a_4p^4 + a_5p^5$

これらの係数に具体的な値を割り当てて、線、放物線、べき関数などのさまざまな曲線形状を生成できます。

直線は、$_2 = a_3 = a_4 = a_5 = 0$ を使用して、次のように生成します。

$f(p) = a_0 + a_1p$

以下も適用されます。

$a_{0:}$ 前のブロック終了の軸位置

$p = PL$

$a_1 = (x_E - a_0 - a_2*p^2 - a_3*p^3)/p$

G 命令 POLY を使用して有効化した多項式補間を**使用しない**多項式をプログラム指令することができます。この場合は、プログラム指令した多項式が補間されませんが、代わりに、プログラム指令した軸のすべての終点へ直線でアプローチします(G1)。プログラム指令した多項式は、パートプログラムで、指定により多項式補間を有効化した後にのみ移動します(POLY)。

特記事項:分母多項式

PO[]=(…) 命令を使用すると、 (軸名称を指定せずに) ジオメトリ軸に普通の分母多項式をプログラム指令できます。つまり、ジオメトリ軸の移動を、2 つの多項式の商として補間します。

このプログラミングオプションを使用すると、円錐曲線(円、楕円、放物線、双曲線)などの形状を正確に表現できます。

例:

プログラムコード	コメント
POLY G90 X10 Y0 F100	; ジオメトリ軸が位置 X10 Y0 へ直線で移動します。
PO[X]=(0,-10) PO[Y]=(10) PO[]=(2,1)	; ジオメトリ軸が 4 分円上を X0 Y10 へ移動します。

分母多項式の定数係数(a_0)は常に 1 とみなされます。プログラム指令した終点は、G90 / G91 には影響されません。

X(p)と Y(p)は、プログラム指令値から、次のように計算されます。

$X(p) = (10 - 10 * p^2) / (1 + p^2)$

$Y(p) = 20 * p / (1 + p^2)$

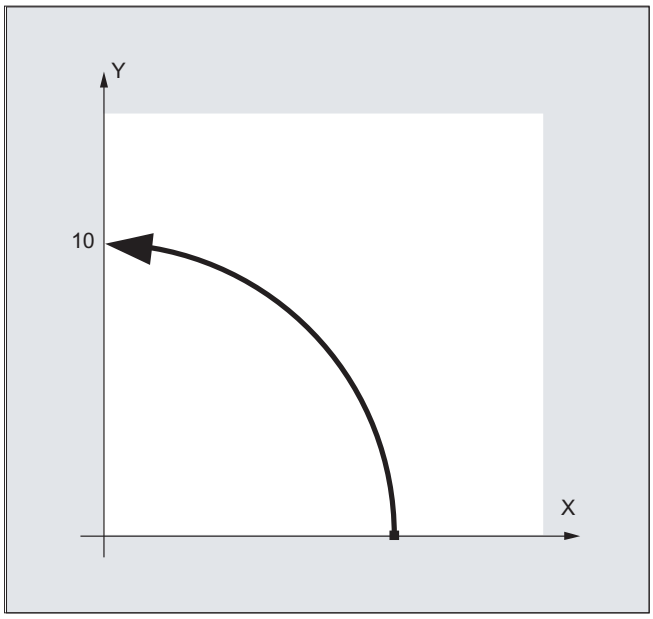
ただし、$0 \le p \le 1$

始点と終点、定数 a_2、PL=1 をプログラム指令した結果、中間結果は次のようになります。

分子(X) = $10 + 0 * p - 10 * p^2$

分子(Y) = $0 + 20 * p + 0 * p^2$

分母= $1 + p^2$



多項式補間が有効で、区間[0,PL]の範囲内がゼロで分母多項式をプログラム指令した場合は、拒否されてアラームが発生します。分母多項式は、付加軸の移動には無効です。

注記

多項式補間と組み合わせて G41、G42 で工具径補正を有効にして、直線補間モードまたは円弧補間モードと同様に適用できます。

5.6 設定可能な軌跡基準(SPATH、UPATH)

多項式補間(POLY、ASPLINE、BSPLINE、CSPLINE、COMPON、COMPCURV)の場合、軌跡軸 i の位置は、多項式 $p_i(U)$で指定されます。曲線パラメータ U は、NC ブロック内で0から 1 に移動します。

FGROUP は、軌跡送り速度 F が適用される軸(FGROUP 軸)を選択します。FGROUP 軸の軌跡 S での定速度による補間は、多項式補間時に曲線パラメータ U の変化が通常は一定でないことを意味します。結果として、FGROUP に含まれない軸を選択する場合、その軸と FGROUP 軸の関係について、以下の 2 種類の方法を使用できます。

- 軌跡 S (SPATH)に同期
- 曲線パラメータ U (UPATH)に同期

構文

```
SPATH
UPATH
```

意味

SPATH:	FGROUP に含まれない軸は、軌跡 S を基準として移動します。
UPATH:	FGROUP に含まれない軸は、曲線パラメータ U を基準として移動します。

注記

UPATH と SPATH は、軌跡移動による F ワード多項式(FPOLY、FCUB、FLIN)の相互関係も定義します。

必要条件

SPATH と UPATH は、以下の場合は無効です。

- 直線補間(G1)
- 円弧補間(G2、G3)
- ねじブロック(G33、G34、G35、G33x、G63)
- すべての軌跡軸が FGROUP に含まれる場合

例

次の例は、両タイプのモーションコントロールの相違点を示します。

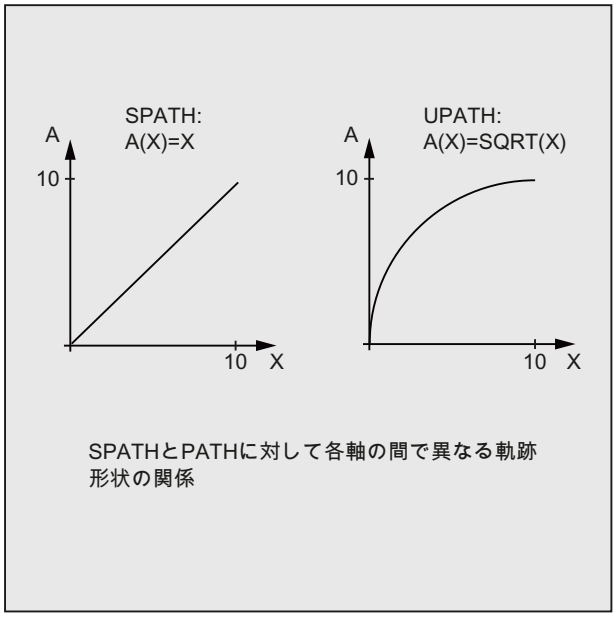
プログラムコード

```
N10 FGROUP(X,Y,Z)
N15 G1 X0 A0 F1000 SPATH                ; SPATH
N20 POLY PO[X]=(10,10) A10
```

プログラムコード

```
N10 FGROUP(X,Y,Z)
N15 G1 X0 A0 F1000 UPATH                ; UPATH
N20 POLY PO[X]=(10,10) A10
```

両方のプログラム区間で、曲線パラメータ U の二乗に応じて N20 の FGROUP 軸の軌跡 S が変わります。したがって、軌跡 X に沿った同期軸 A の位置が、SPATH と UPATH のどちらが有効かによって異なります。



SPATHとPATHに対して各軸の間で異なる軌跡形状の関係

詳細情報

リセットの制御動作、およびマシンデータ/オプションデータ

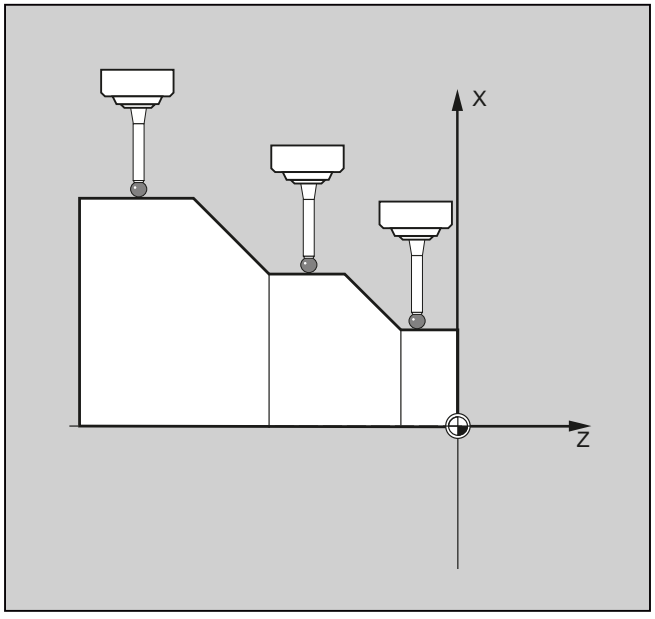
MD20150 $MC_GCODE_RESET_VALUES[44]によって定義される G 命令は、リセット後に有効になります(45 番目の G グループ)。

スムージングのタイプの初期状態は、MD20150 $MC_GCODE_RESET_VALUES[9] (10 番目の G グループ)で定義します。

軸別のマシンデータ MD33100 $MA_COMPRESS_POS_TOL[<n>]は、拡張された意味があります。これには、コンプレッサ機能とスムージング(G642 による)の許容範囲が含まれています。

5.7 タッチトリガプローブによる計測(MEAS、MEAW)

「タッチトリガブローブによる計測」を使用して、ワークの実位置へアプローチします。プローブの切り替えエッジ、計測ブロックでプログラム指令したすべての軸の位置を計測し、各軸の適切なメモリに書き込みます。



この機能のプログラミングには、次の 2 つの固定アドレスを使用できます。

- MEAS
  MEAS は、実位置と指令位置の間の残移動距離を削除します。
- MEAW
  MEAW は、常にプログラム指令位置へのアプローチが必要な計測処理に使用します。

MEAS と MEAW はノンモーダルで、移動命令と一緒にプログラム指令します。送り速度と補間タイプ(G0、G1 など)だけでなく、軸数も、それぞれの計測処理に合わせて補正してください。

構文

```
MEAS=<TE> G...X...Y...Z...
MEAW=<TE> G...X...Y...Z...
```

意味

MEAS:	命令:残移動距離を**削除する**計測	
	効果:	ノンモーダル

MEAW:	命令:残移動距離を**削除しない**計測	
	効果:	ノンモーダル
<TE>:	計測を開始する事象を起動します	
	タイプ:	INT
	値の範囲:	-2, -1, 1, 2
	意味	
	(+)1	プローブ 1 の立ち上がり(計測入力 1)
	-1	プローブ 1 の立ち下がり(計測入力 1)
	(+)2	プローブ 2 の立ち上がり(計測入力 2)
	-2	プローブ 2 の立ち下がり(計測入力 2)
	注: プローブは最大 2 個です(構成レベルにより異なります)。	
G...:	G0、G1、G2、G3 などの補間タイプ	
X...Y...Z...:	直交座標の終点	

例

プログラムコード	コメント
N10 MEAS=1 G1 F1000 X100 Y730 Z40 ...	; 1 番目の計測点でのプローブによる計測ブロックと直線補間です。先読み停止は自動的におこなわれます。

詳細情報

計測処理状態

プログラムのなかで、プローブがトリガされたかどうかを評価したい場合は、状態変数 $AC_MEA[<n>] (<n>=計測プローブの番号)を以下の値で確認できます。

値	意味
0	計測処理が完了していません。
1	計測処理が正常に完了しました(プローブがトリガされました)。

注記

プログラムの中でスイッチオンされた場合、変数は 1 に設定されます。計測ブロックの先頭で、変数は自動的にプローブの初期状態に設定されます。

計測値の読み取り

ブロックのすべての移動軌跡と位置決め軸の位置が得られます(軸の最大数は制御装置の構成により異なります)。MEAS の場合は、プローブがトリガされた後に、定義した方法で移動が減速します。

注記

ジオメトリ軸が計測ブロックにプログラム指令されている場合は、現在のすべてのジオメトリ軸の計測値が保存されます。

変換に関与する軸が計測ブロックにプログラム指令されている場合は、この変換に関与するすべての軸の計測値が記録されます。

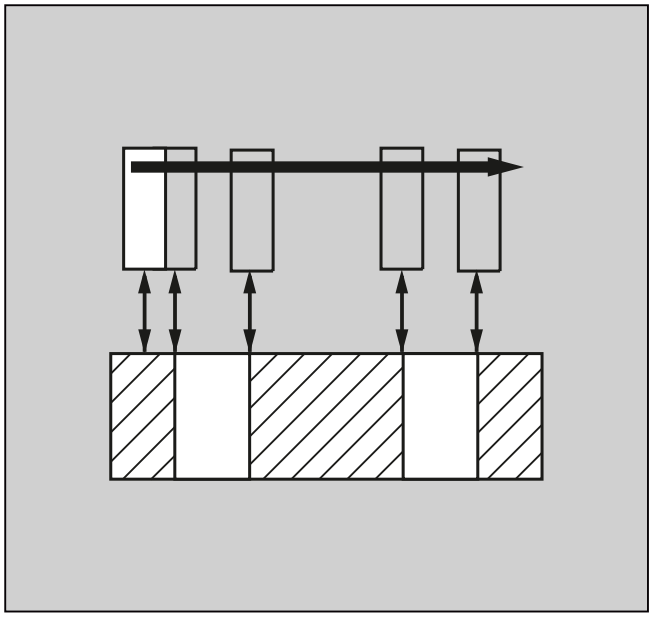
計測結果の読み取り

プローブで計測された軸の計測結果は、次のシステム変数により読み取ることができます。

- $AA_MM[<軸>]
  機械座標系の計測結果
- $AA_MW[<軸>]
  ワーク座標系の計測結果

5.8 軸別計測(MEASA、MEAWA、MEAC) (オプション)

軸別計測には、複数のプローブと複数の検出器を使用できます。



この機能のプログラミングには、キーワード MEASA、MEAWE、および MEAC を使用できます。

プログラム指令軸に MEASA または MEAWA を使用する場合は、各計測で最高 4 つの計測値が取得され、トリガ事象に対応したシステム変数に保存されます。

計測操作は MEAC で実行できます。この場合は、計測結果が FIFO 変数に保存されます。

構文

```
MEASA[<軸>]=(<モード>,<TE1>,...,<TE4>)
MEAWA[<軸>]=(<モード>,<TE1>,...,<TE4>)
MEAC[<軸>]=(<モード>,<計測メモリ>,<TE1>,...,<TE4>)
```

注記

MEASA と MEAWA はノンモーダルで、1 ブロックに一緒にプログラム指令できます。ただし、同じブロックで MEASA/MEAWA を MEAS/MEAW と一緒にプログラム指令した場合は、エラーメッセージが出力されます。

意味

MEASA:	キーワード:残移動距離を削除する軸別計測	
	効果:	ノンモーダル
MEAWA:	キーワード:残移動距離を削除しない軸別計測	
	効果:	ノンモーダル
MEAC:	キーワード:残移動距離を削除しない軸別連続計測	
	効果:	ノンモーダル
<軸>:	計測に使用するチャネル軸の名称	
<モード>:	運転モードを示す 2 桁の数字(計測モードと検出器)	
	10 進数の 1 の位(計測モード):	
	0	計測処理のキャンセル.
	1	4 種類までのトリガ事象を同時に有効にできます。
	2	4 種類までのトリガ事象を連続して有効にできます。
	3	4 種類までのトリガ事象を連続して有効にできますが、起動時にトリガ事象 1 は監視されません(アラーム 21700/21703 はマスクされます)。 **注:** このモードは MEAC ではサポートされていません。
	10 進数の 10 の位(検出器):	
	0 (またはデータなし)	有効な検出器
	1	検出器 1
	2	検出器 2
	3	両方の検出器

<TE>:	計測を開始する事象をトリガします	
	タイプ:	INT
	値の範囲:	-2, -1, 1, 2
	意味	
	(+)1	プローブ 1 の立ち上がり
	-1	プローブ 1 の立ち下がり
	(+)2	プローブ 2 の立ち上がり
	-2	プローブ 2 の立ち下がり
<計測メモリ>:	FIFO の番号(循環バッファ)	

例

例 1:モード 1 の、残移動距離を削除する軸別計測(時系列の順に評価)

a) 検出器が 1 基の場合

プログラムコード	コメント
...	
N100 MEASA[X]=(1,1,-1) G01 X100 F100	; 有効な検出器によるモード 1 の計測。プローブ 1 からの立ち上がり/立ち下がりで、X=100 への移動軌跡の計測信号を待機します。
N110 IF $AC_MEA[1]==FALSE GOTOF END	; 計測が正常におこなわれたことを確認します。
N120 R10=$AA_MM1[X]	; 1 番目にプログラム指令のトリガ事象 (立ち上がり) で得られた計測値を保存します。
N130 R11=$AA_MM2[X]	; 2 番目にプログラム指令のトリガ事象 (立ち下がり) で得られた計測値を保存します。
N140 END:	

b) 検出器が 2 基の場合

プログラムコード	コメント
...	
N200 MEASA[X]=(31,1,-1) G01 X100 F100	; 両方の検出器によるモード 1 の計測。プローブ 1 からの立ち上がり/立ち下がりで、X=100 への移動軌跡の計測信号を待機します。
N210 IF $AC_MEA[1]==FALSE GOTOF END	; 計測が正常におこなわれたことを確認します。
N220 R10=$AA_MM1[X]	; 立ち上がりで検出器 1 の計測値を保存します。
N230 R11=$AA_MM2[X]	; 立ち上がりで検出器 2 の計測値を保存します。
N240 R12=$AA_MM3[X]	; 立ち下がりで検出器 1 の計測値を保存します。
N250 R13=$AA_MM4[X]	; 立ち下がりで検出器 2 の計測値を保存します。

プログラムコード	コメント
N260 END:	

例 2 :モード 2 の、残移動距離を削除する軸別計測(時系列の順に評価)

プログラムコード	コメント
...	
N100 MEASA[X]=(2,1,-1,2,-2) G01 X100 F100	; 有効な検出器によるモード 2 の計測。X=100 への移動軌跡の間、立ち上がりプローブ 1、立ち下がりプローブ 1、立ち上がりプローブ 2、立ち下がりプローブ 2 という順に計測信号を待機します。
N110 IF $AC_MEA[1]==FALSE GOTOF PROBE2	; プローブ 1 による計測が正常におこなわれたことを確認します。
N120 R10=$AA_MM1[X]	; 1 番目にプログラム指令のトリガ事象 (プローブ 1 の立ち上がり) で得られた計測値を保存します。
N130 R11=$AA_MM2[X]	; 2 番目にプログラム指令のトリガ事象 (プローブ 1 の立ち上がり) で得られた計測値を保存します。
N140, PROBE2:	
N150 IF $AC_MEA[2]==FALSE GOTOF END	; プローブ 2 による計測が正常におこなわれたことを確認します。
N160 R12=$AA_MM3[X]	; 3 番目にプログラム指令のトリガ事象 (プローブ 2 の立ち上がり) で得られた計測値を保存します。
N170 R13=$AA_MM4[X]	; 4 番目にプログラム指令のトリガ事象 (プローブ 2 の立ち上がり) で得られた計測値を保存します。
N180 END:	

例 3 :モード 1 の軸別連続計測(時系列の順に評価)

a) 100 個までの計測値の計測

プログラムコード	コメント
...	
N110 DEF REAL MEASVALUE[100]	
N120 DEF INT loop=0	
N130 MEAC[X]=(1,1,-1) G01 X1000 F100	; 有効な検出器によるモードの計測。$AC_FIFO1 に計測値を保存します。X=1000 への移動軌跡の間に、検出器のプローブ 1 の立ち下がりを待機します。
N135 STOPRE	
N140 MEAC[X]=(0)	; 軸位置に到達すると、計測を終了します。
N150 R1=$AC_FIFO1[4]	; パラメータ R1 に計測値の累計個数を保存します。
N160 FOR loop=0 TO R1-1	
N170 MEASURED VALUE[loop]=$AC_FIFO1[0]	; $AC_FIFO1 から計測値を読み出し、保存します。
N180 ENDFOR	

b) 10 個の計測値を得た後の、残移動距離を削除する計測

プログラムコード	コメント
...	
N10 WHEN $AC_FIFO1[4]>=10 DO MEAC[x]=(0) DELDTG(x)	; 残移動距離削除
N20 MEAC[x]=(1,1,1,-1) G01 X100 F500	
N30 MEAC [X]=(0)	
N40 R1 = $AC_FIFO1[4]	; 計測値の数
...	

c) 2 つのプローブによる立ち下がり/立ち上がり歯面の計測

プログラムコード	コメント
...	
N110 DEF REAL MEASVALUE[16]	
N120 DEF INT loop=0	
N130 MEAC[X]=(1,1,-1,2) G01 X100 F100	; 有効な検出器によるモード 1 の計測。$AC_FIFO1 により計測値を保存します。X=100 への移動軌跡の間に、プローブ 1 の立ち下がり、プローブ 2 の立ち上がりの順に計測信号を待機します。
N140 STOPRE	; 先読み停止
N150 MEAC[X]=(0)	; 軸位置に到達すると、計測を終了します。
N160 R1=$AC_FIFO1[4]	; パラメータ R1 に計測値の累計個数を保存します。
N170 FOR loop=0 TO R1-1	
N180 MEASURED VALUE[loop]=$AC_FIFO1[0]	; $AC_FIFO1 から計測値を読み出し、保存します。
N190 ENDFOR	

関連情報

計測ジョブ

計測処理は、パートプログラムで、またはシンクロナイズドアクションからプログラム指令できます(「シンクロナイズドアクション (ページ 729)」の章を参照してください)。

各軸がいつでも有効にすることができるのは、1 つの計測ジョブのみであることに注意してください。

注記

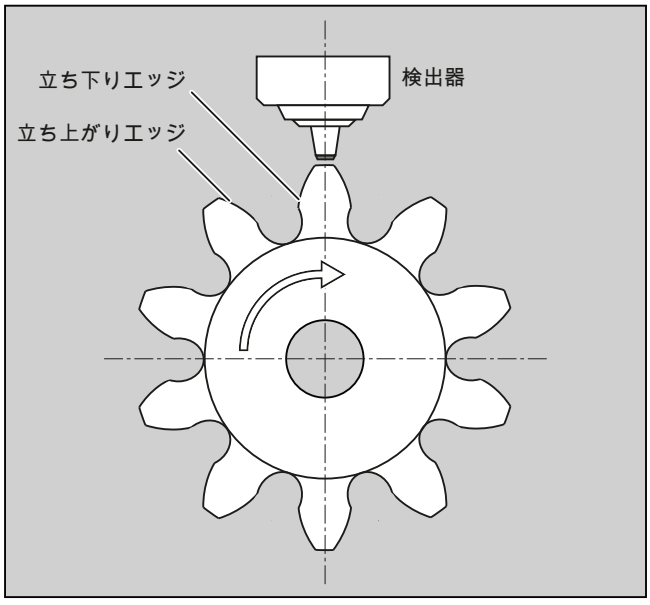
実行中の計測処理に合わせて、送り速度を調節してください。

MEASA と MEAWA の場合は、結果の正確性は送り速度に対して保証されます。この送り速度で、同じタイプの 1 つまでのトリガ事象、および異なるタイプの 4 つまでのトリガ事象のみが各位置制御周期で発生します。

MEAC による連続計測の場合は、補間クロックサイクルと位置制御サイクルの比率が 1:8 を超えないようにしてください。

トリガ事象

トリガ事象は、プローブ番号と計測信号のトリガ条件(立ち上がりまたは立ち下がり)から成ります。



それぞれの計測では、使用するプローブのトリガ事象を 4 つまで処理できます。つまり、処理できるのは、それぞれ 2 つの計測信号エッジによって 2 つまでのプローブが処理できます。トリガ事象の処理順序と最大数は、選択したモードにより異なります。

注記

計測モード 1 には以下が適用されます。同じトリガ事象は、1 回の計測処理に一度だけプログラム指令できます。

MEAC では、トリガ事象ごとの計測値の数は、PROFIBUS メッセージ 395 を使用して、各トリガ事象と位置コントローラサイクルについて立ち上がりの 8 つの計測値、立ち下がりの 8 つの計測値まで増やすことができます。

- 1 つのプローブ:立ち上がり用に 8 つの計測値、立ち下がり用に 8 つの計測値
- 2 つのプローブ:各プローブに対して、立ち上がり用に 4 つの計測値、立ち下がり用に 4 つの計測値

これは、PROFIBUS メッセージ 395 を使用して、送り速度または速度を上げられることを意味します。

参照先:
機能マニュアル 上級機能; 計測(M5)、「軸の測定」の章

運転モード

運転モードの 1 番目の桁(10 進数の 10 の位)で、必要な検出器を選択します。取り付けている検出器は 1 基のみであるが、2 基目がプログラム指令されている場合は、取り付けている検出器を自動的に選択します。

2 番目の桁(10 進数の 1 の位)で、必要な計測モードを選択します。これで、計測処理が、当該の制御装置がサポートするオプションに合わせて次のように調整されます。

- **モード 1**
  トリガ事象は、発生順に時系列で評価されます。このモードを選択すると、6 軸モジュールに対するトリガ事象を 1 つだけプログラム指令できます。複数のトリガ事象を指定した場合は、選択したモードが自動的にモード 2 に切り替わります(メッセージはありません)。
- **モード 2**
  トリガ事象は、プログラム指令順に評価されます。
- **モード 3**
  トリガ事象は、プログラム指令順に評価されますが、START 時にトリガ事象 1 の監視はおこなわれません。

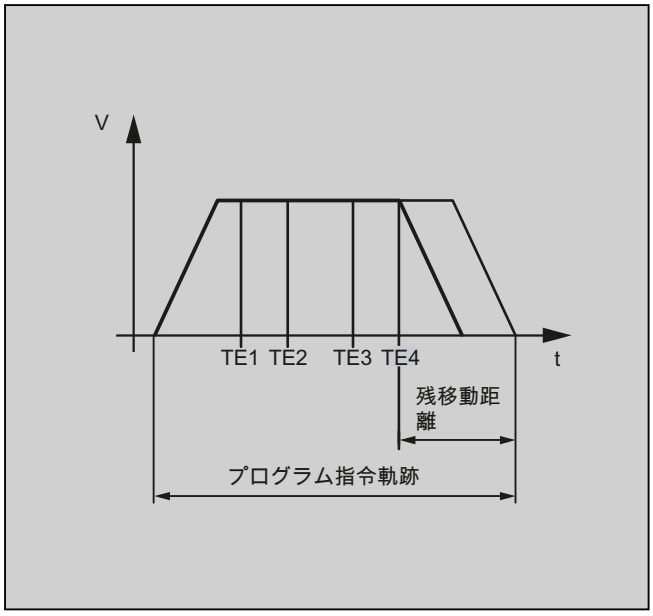
注記

2 台の検出器を使用している場合は、2 つを超えるトリガ事象をプログラム指令できません。

残移動距離を削除する計測と削除しない計測

MEASA 命令をプログラム指令した場合は、必要な計測値がすべて記録されるまでは残移動距離が削除されません。

MEAWA 機能は、常にプログラム指令位置へのアプローチが必要な特別な計測処理に使用します。



注記

MEASA は、シンクロナイズドアクションではプログラム指令できません。別の方法として、MEAWA に残移動距離削除を加えて、シンクロナイズドアクションとしてプログラム指令できます。

MEAWA による計測処理をシンクロナイズドアクションから起動した場合は、計測値は機械座標系でのみ得ることができます。

MEASA、MEAWA の計測結果

計測結果は、次のシステム変数で得ることができます。

- 機械座標系の場合:

$AA_MM1[<軸>]	トリガ事象 1 でプログラム指令した検出器の計測値
...	...
$AA_MM4[<軸>]	トリガ事象 4 でプログラム指令した検出器の計測値

- ワーク座標系の場合:

$AA_MW1[<軸>]	トリガ事象 1 でプログラム指令した検出器の計測値
...	...
$AA_MW4[<軸>]	トリガ事象 4 でプログラム指令した検出器の計測値

ジオメトリ軸/座標変換

ジオメトリ軸に対して軸計測を起動する場合は、その他のジオメトリ軸のすべてに、同じ計測ジョブを明示的にプログラム指令してください。同じことが、座標変換の関連軸にも適用されます。

例:
```
N10 MEASA[Z]=(1,1) MEASA[Y]=(1,1) MEASA[X]=(1,1) G0 Z100
```
または
```
N10 MEASA[Z]=(1,1) POS[Z]=100
```

2 基の検出器による計測ジョブ

2 基の検出器で計測ジョブを実行する場合は、当該の軸の両方の検出器で 2 つのトリガ事象を、それぞれ得ることができます。そのため、予約変数の割り当てが設定されます。

$AA_MM1[<軸>]	または $AA_MW1[<軸>]	トリガ事象 1 の検出器 1 からの計測値
$AA_MM2[<軸>]	または $AA_MW2[<軸>]	トリガ事象 1 の検出器 2 からの計測値
$AA_MM3[<軸>]	または $AA_MW3[<軸>]	トリガ事象 2 の検出器 1 からの計測値
$AA_MM4[<軸>]	または $AA_MW4[<軸>]	トリガ事象 2 の検出器 2 からの計測値

システム変数

プローブ状態は、以下のシステム変数で得ることができます。

$A_PROBE[<n>]

値	意味
1	スイッチがオンされたプローブ
0	スイッチがオフされたプローブ

プローブ制限は、以下のシステム変数で取得できます。

$A_PROBE_LIMITED[<n>]

値	意味
1	プローブ制限が有効
0	プローブ制限が無効

<n> = プローブ

参照先:
リストマニュアル、システム変数

MEASA、MEAWA の計測ジョブ状態

プログラムで評価する必要がある場合は、計測処理状態を$AC_MEA[<n>]で確認できます。ここで、<n>はプローブの番号です。同じブロックでプログラムしたプローブ<n>のすべてのトリガ事象が発生し終わると、この変数は「1」の値を返します。それ以外の場合の値は 0 です。

注記

計測がシンクロナイズドアクションから起動された場合は、$AC_MEA は更新されなくなります。この場合は、新しい PLC インタフェース信号 DB31, … DBX62.3 または同等の変数$AA_MEAACT[<軸>]を確認してください。

意味

$AA_MEAACT==1:計測が有効です。

$AA_MEAACT==0:計測が無効です。

連続計測(MEAC)

MEAC の計測値は機械座標系で得ることができ、プログラム指令した FIFO[n]メモリ(循環バッファ)に保存されます。2 個のプローブを計測用に構成している場合は、2 番目のプローブの計測値は、このために特に設定された(マシンデータで定義した)FIFO[n+1]メモリに別に格納されます。

FIFO メモリはリングバッファで、ここには、計測値が循環規則に従って$AC_FIFO 変数に書き込まれます。「シンクロナイズドアクション (ページ 729)」の章を参照してください。

注記

FIFO の内容は、循環バッファから一回だけ読み取ることができます。この計測データを複数回使用する場合は、ユーザーデータに保持
してください。

FIFO メモリの計測値の数が、マシンデータで定義した最大値を超えた場合は、計測が自動的に終了します。

計測値を周期的に読み出すと、無限の計測処理を実行できます。この場合は、新しい計測値の読み込みと同じ頻度でデータを読み出してください。

参照先:

- 機能マニュアル シンクロナイズドアクション; 詳細説明、「パラメータ($AC_FIFO)」の章
- 機能マニュアル 上級機能; 計測(M5)、「軸の測定」の章

プログラミングエラーに対する保護

以下のプログラミングエラーが検出され、適切に表示されます。

- MEAS/MEAW と MEASA/MEAWA のプログラム指令が同一ブロックにあります
  例:
  ```
  N01 MEAS=1 MEASA[X]=(1,1) G01 F100 POS[X]=100
  ```
- MEASA/MEAWA のパラメータの数が<2 または>5 です
  例:
  ```
  N01 MEAWA[X]=(1) G01 F100 POS[X]=100
  ```
- MEASA/MEAWA のトリガ事象が 1/ -1/ 2/ -2 ではありません
  例:
  ```
  N01 MEASA[B]=(1,1,3) B100
  ```
- 無効なモードの MEASA/MEAWA です
  例:
  ```
  N01 MEAWA[B]=(4,1) B100
  ```
- MEASA/MEAWA のトリガ事象が 2 回プログラム指令されています
  例:
  ```
  N01 MEASA[B]=(1,1,-1,2,-1) B100
  ```

- MEASA/MEAWA とジオメトリ軸がありません
  例:
  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) G01 X50 Y50 Z50 F100 ;GEO axis X/Y/Z
  ```
- ジオメトリ軸と計測処理が不整合です
  例:
  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) MEASA[Z]=(1,1,2) G01 X50 Y50 Z50 F100
  ```

5.9 OEM ユーザー用の応用機能(OMA1 ... OMA5、OEMIPO1、OEMIPO2、G810 ... G829)

OEM アドレス

OEM アドレスの意味は、OEM ユーザーが特定します。その機能は、コンパイルサイクルを使用して組み込まれています。5 つの OEM アドレスが予約されています (OMA1 ... OMA5)。アドレス識別子は設定できます。OEM アドレスはどのブロックにもプログラム指令できます。

予約済み G 命令呼び出し

以下の G 命令呼び出しが OEM ユーザーに予約されています。

- OEMIPO1、OEMIPO2 (G グループ 1)
- G810 ... G819 (G グループ 31)
- G820 ... G829 (G グループ 32)

その機能は、コンパイルサイクルを使用して組み込まれています。

機能とサブプログラム

また、OEM ユーザーは、予約機能とサブプログラムをパラメータ転送を使用して設定することもできます。

注記

ワークシミュレーション

SW 4.4 まではコンパイルサイクルはサポートされていません。SW 4.4 では、選択されたコンパイルサイクル(CC)のみがワークシミュレーション用にサポートされています。

したがって、サポートされていないコンパイルサイクルのパートプログラム内の言語命令(OMA1 ... OMA5、OEMIPO1/2、G810 ... G829、ユーザー自身の手順と機能)は、その結果アラームメッセージが発生し、個々の処理をおこなわずにシミュレーションがキャンセルされます。

対策:パートプログラム($P_SIM query)で、欠落している CC 用の言語要素を個別に処理します。
例:

```
N1 G01 X200 F500
IF (1==$P_SIM)
N5 X300 ;not active for CC simulation
ELSE
N5 X300 OMA1=10
ENDIF
```

5.10 コーナ減速による送り速度低下(FENDNORM、G62、G621)

自動コーナ減速を使用すると、コーナに達する前に、送り速度がベル型カーブを描いて減速します。また、セッティングデータで、加工に関連する工具動作の範囲をパラメータ設定することもできます。これには、以下が含まれます。

- 送り速度減速の開始と終了
- 送り速度を減速するオーバライド
- 当該のコーナの検出

当該のコーナは、コーナ内角の角度が、セッティングデータでパラメータ設定したコーナ角度より小さい角度です。

初期値 FENDNORM は、自動コーナオーバライドの機能を解除します。

参照先:
/FBFA/「機能説明書 G コード言語」

構文

```
FENDNORM
G62 G41
G621
```

意味

FENDNORM:	自動コーナ減速オフ
G62:	工具径補正が有効なときの内側コーナのコーナ減速
G621:	工具径補正が有効なときのすべてのコーナのコーナ減速

G62 は、以下の場合の内側コーナでのみ適用されます。

- 工具径補正 G41、G42 が動作中、および
- 連続軌跡モード G64、G641 が動作中

コーナでは、以下で得られる減速された送り速度でアプローチします。

F * (送り速度減速オーバライド) *送り速度オーバライド

送り速度低下は、工具がコーナで中心軌跡を基準として方向転換する、ちょうどその位置で最大となります。

G621 は、G62 と同様に、FGROUP で定義した軸のそれぞれのコーナで適用されます。

5.11 プログラム指令可能な動作終了条件(FINEA、COARSEA、IPOENDA、IPOBRKA、ADISPOSA)

軌跡補間(G601、G602、および G603)のブロック切り替え条件と同様に、パートプログラムのとき、またはコマンド軸/PLC 軸のシンクロナイズドアクションのときの単独軸補間の動作終了条件もプログラム指令できます。

動作終了条件の指令は、単独軸移動のよるパートプログラムブロックとテクノロジサイクルブロックが完了するまでの時間に影響します。これは、FC15/16/18 経由の PLC にも適用されます。

構文

```
FINEA[<軸>]
COARSEA[<軸>]
IPOENDA[<軸>]
IPOBRKA(<軸>[,<タイミング>])
ADISPOSA[<軸>]=(<モード>,<範囲のサイズ>)
```

意味

FINEA:	動作終了条件:「精密イグザクトストップ」	
	有効:	モーダル
COARSEA:	動作終了条件:「汎用イグザクトストップ」	
	有効:	モーダル
IPOENDA:	動作終了条件:「補間停止」	
	有効:	モーダル
IPOBRKA:	ブロック切り替え条件:減速カーブ	
	有効:	モーダル
ADISPOSA:	動作終了条件の許容範囲	
	有効:	モーダル
<軸>:	チャネル軸名称(X、Y、...)	
<タイミング>:	ブロック切り替えのタイミング、減速カーブを基準にして%で表わします。 • 100% = 減速カーブの開始 • 0% = 減速カーブの終了、IPOENDA と同じ意味です。	
	タイプ:	REAL

<モード>:	許容範囲の基準		
	値の範囲:	0	許容範囲が有効ではありません
		1	設定位置の許容範囲
		2	実位置の許容範囲
	タイプ:	INT	
<範囲のサイズ>:	許容範囲のサイズ		
	タイプ:	REAL	

例

例 1:動作終了条件:「補間停止」

プログラムコード

```
; 位置決め軸 X を 100 に移動、速度 200 m/min、 加速度 90%、
; 動作終了条件:補間停止
N110 G01 POS[X]=100 FA[X]=200 ACC[X]=90 IPOENDA[X]

; シンクロナイズドアクション:
; もし常に、入力 1 が設定された場合は
; 位置決め軸を 50 に移動、速度 200 m/min、加速度 140%、
;       動作終了条件:補間停止
N120 EVERY $A_IN[1] DO POS[X]=50 FA[X]=200 ACC[X]=140
IPOENDA[X]
```

例 2:ブロック切り替え条件:「減速カーブ」

プログラムコード	コメント
	; 初期設定が有効です。
N40 POS[X]=100	; X から位置 100 への位置決め動作です。
	ブロック切り替え条件:精密イグザクトストップ
N20 IPOBRKA(X,100)	; ブロック切り替え条件:「減速カーブ」,
	100% = 減速カーブの開始
N30 POS[X]=200	; ブロックは X 軸の減速開始と同時に切り替えられます。
N40 POS[X]=250	; X 軸は位置 200 で減速せずに、位置 250 まで移動し続けます。
	軸の減速開始と同時にブロックが切り替えられます。
N50 POS[X]=0	; X 軸が減速して位置 0 に戻ります。
	位置 0 でブロックが切り替えられ、「精密イグザクトストップ」となります。
N60 X10 F100	; X 軸が軌跡軸として位置 10 に移動します。

詳細情報

動作終了条件用のシステム変数

有効な動作終了条件を、システム変数$AA_MOTEND を使用して読み取ることができます。

参照先: /LIS2sl/ リストマニュアル, Book 2

ブロック切り替え条件: 「減速カーブ」 (IPOBRKA)

ブロック切り替え条件「減速カーブ」を有効にするときに、任意選択のブロック切り替え点のタイミングの値がプログラム指令されている場合、この値は次の位置決め動作に対して有効になり、メインランに同期してセッティングデータに書き込まれます。 ブロック切り替え点のタイミングの値が指定されていない場合、セッティングデータの現在値が有効になります。

SD43600 $SA_IPOBRAKE_BLOCK_EXCHANGE

IPOBRKA は、軸の動作終了条件(FINEA、COARSEA 、IPOENDA)がその次に軸へプログラム指令されると、対応するアクセスに対して無効になります。

追加のブロック切り替え条件: 「許容範囲」(ADISPOSA)

ADISPOSA を使用して、ブロック終点周囲の許容範囲(現在位置または指令位置)を追加のブロック切り替え条件として定義できます。 ブロック切り替えに対して以下の 2 つの条件を満たしてください。

- ブロック切り替え条件: 「減速カーブ」
- ブロック切り替え条件: 「許容範囲」

参照先

位置決め軸のブロック切り替え条件について詳しくは、以下を参照してください。

- 機能マニュアル 上級機能; 位置決め軸(P2)
- プログラミングマニュアル 基礎編;「送り速度制御」の章

座標変換(フレーム) 6

6.1 フレーム変数による座標変換

機能マニュアル 基本機能の「座標変換(フレーム)」の章で説明する ROT、AROT、SCALE などの命令の他に、フレーム変数$P_...FR (データ記憶フレーム)や$P_...FRAME (動作中のフレーム)によってワーク座標系(WCS)を変換することもできます。

次の図は、フレーム変数の構造の概要を示したものです。

- データ管理フレーム
- 動作中のフレーム
- 動作中の全体フレーム:動作中の全フレームの結合
- NCU グローバルフレーム
- チャネル別フレーム

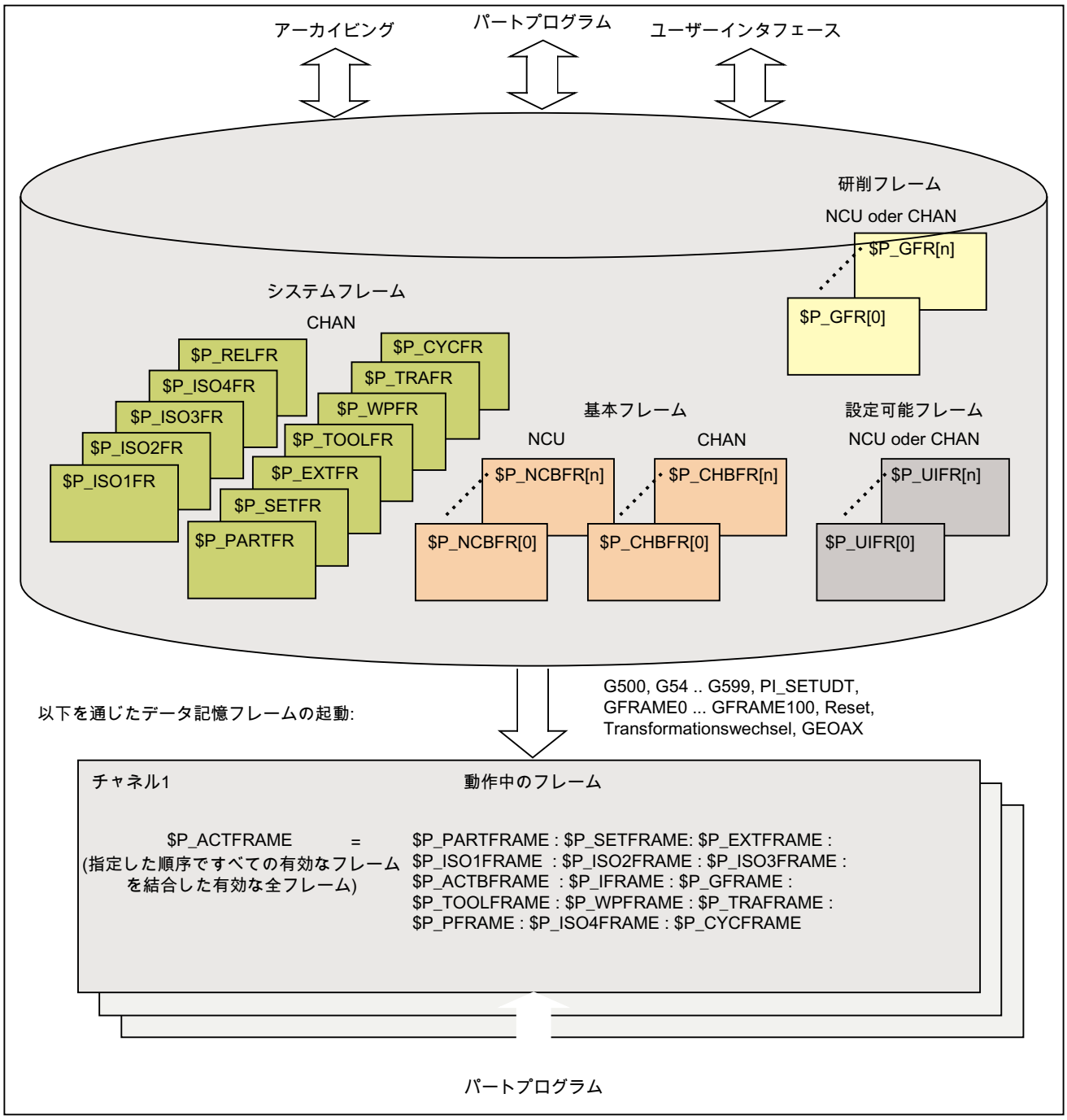


図 6-1　フレーム変数の概要

6.1.1 予約フレーム変数($P_CHBFRAME、$P_IFRAME、$P_PFRAME、$P_ACTFRAME)

有効:チャネル別基本フレーム$P_CHBFRAME[<n>] ($P_BFRAME)

注記

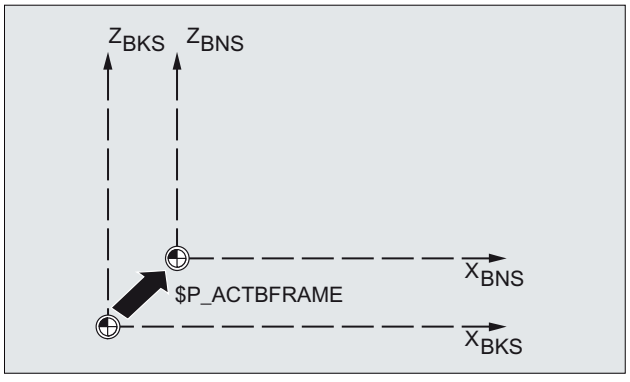
現在の基本フレーム$P_BFRAME とデータ記憶の基本フレーム$P_UBFR は、互換性の理由から保持されます。

- $P_BFRAME □ $P_CHBFRAME[0]
- $P_UBFR □ $P_CHBFR[0]

フレーム変数$P_CHBFRAME[<n>]は、基本座標系(BCS)と基本ゼロオフセットシステム(BZS)の間の位置関係を定義します。

現在のチャネル別基本フレーム$P_CHBFRAME[<n>]を、NC プログラムですぐに有効にするには、以下の方法を使用できます。

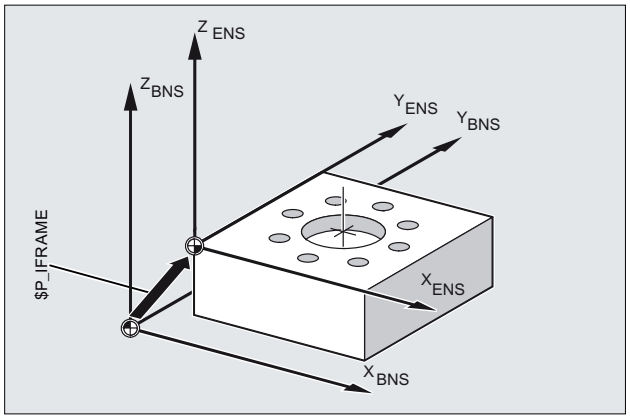
- 命令:
  - G500 (すべての設定可能フレームを無効にし、基本フレームは有効なままにします)
  - G54 ～ G599 (設定可能なゼロオフセット)
- データ記憶のチャネル別基本フレームを、現在のチャネル別基本フレームに割り当て
  $P_CHBFRAME[<n>] = $P_CHBFR[<m>]



有効:チャネル別設定可能フレーム$P_IFRAME

フレーム変数$P_IFRAME は、基本ゼロオフセットシステム(BZS)と設定可能ゼロオフセットシステム(SZS)の間の位置関係を定義します。

- $P_IFRAME は$P_UIFR[$P_IFRNUM]に対応します。
- たとえば、G54 をプログラム指令後に$P_IFRAME には、G54 で定義した平行移動、回転、スケーリング、およびミラーリングが含まれます。

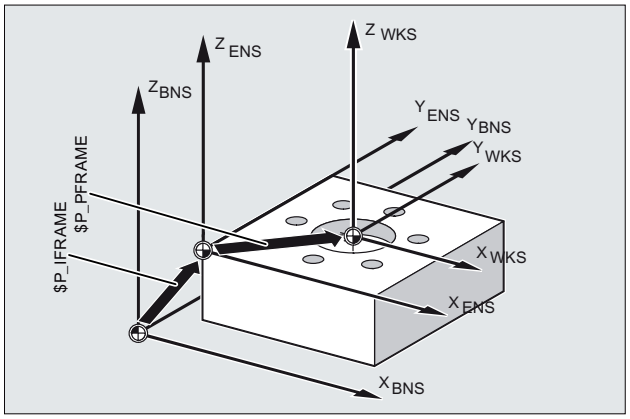


有効:チャネル別プログラマブルフレーム$P_PFRAME

$P_PFRAME フレーム変数は、設定可能ゼロオフセットシステム(SZS)とワーク座標系(WCS)の間の位置関係を定義します。

$P_PFRAME には、以下のいずれかの結果のフレームが含まれます。

- TRANS/ATRANS、ROT/AROT、SCALE/ASCALE、MIRROR/AMIRROR のプログラミングの結果、または
- **プログラム指令 FRAME への** CTRANS、CROT、CMIRROR、CSCALE の割り当ての結果



有効:全体フレーム$P_ACTFRAME

チャネル内の有効な全体フレームは、チャネル内で機能するすべてのフレームを結合したものです。

```
$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 は、現在有効なワーク座標系を記述します。

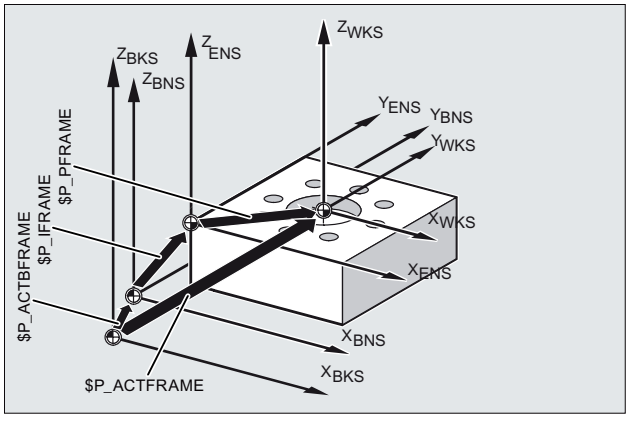
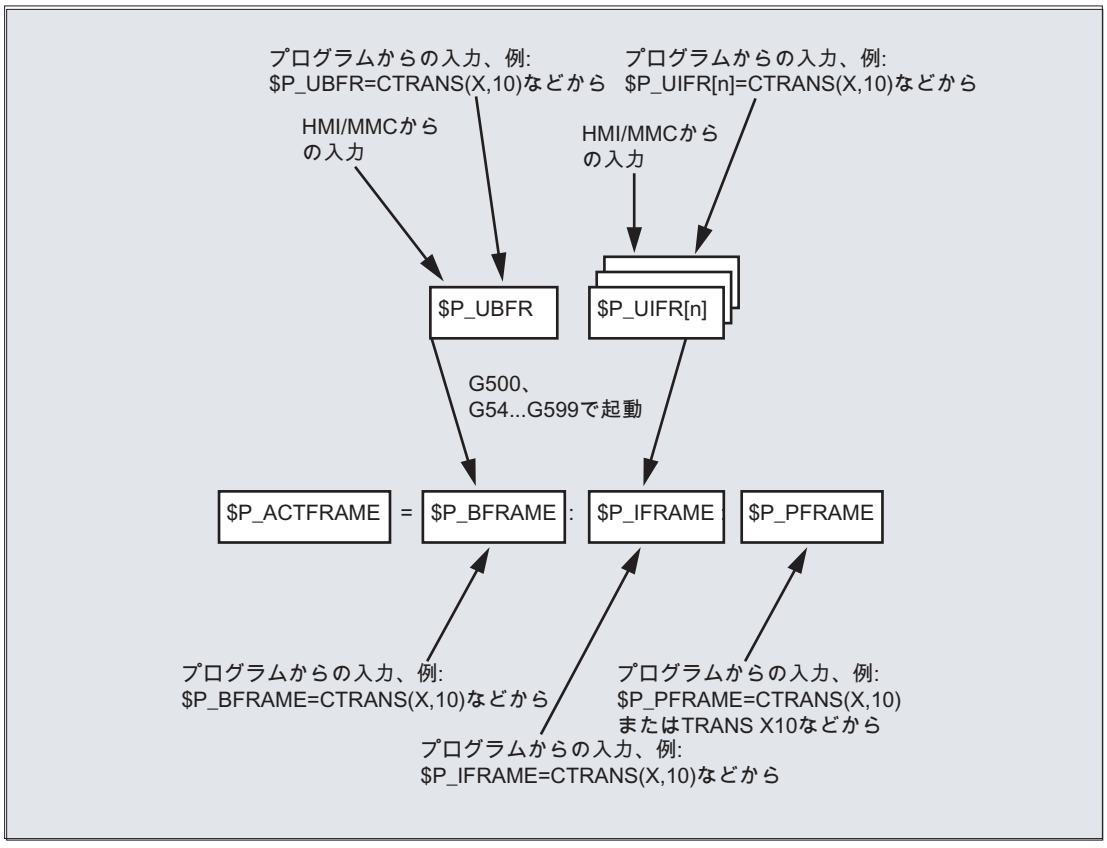


図 6-2　フレーム変数$P_ACTFRAME

$P_BFRAME / $P_CHBFRAME[<n>]、$P_IFRAME、または$P_PFRAME のいずれかのフレームが変更された場合、現在の全体フレーム$P_ACTFRAME が再計算されます。



MD 20110 RESET_MODE_MASK を以下のように設定した場合は、リセット後に基本フレームと設定可能フレームが有効になります。

ビット 0=1、ビット 14=1 --> $P_UBFR (基本フレーム)が動作します。

ビット 0=1、ビット 5=1 --> $P_UIFR [$P_UIFRNUM](設定可能フレーム)が動作します。

データ記憶:チャネル別基本フレーム$P_CHBFR[<n>]

フレーム変数$P_CHBFR[<n>]は、データ記憶の基本フレームを読み取り/書き込みます。データ記憶フレームは、書き込んだときにチャネル内ですぐには有効になりません。書き込まれたフレームは、以下により有効になります。

- チャネルリセットと MD20110 $MC_RESET_MODE_MASK、ビット 0 == 1 およびビット 14 == 1
- 命令 G500、G54 ... G57、G505 ... G599 (基本フレームの有効化/無効化と、その後の現在の全体フレームの再計算)

データ記憶:チャネル別設定可能フレーム$P_UIFR[<n>]

フレーム変数$P_UIFR[<n>]は、データ記憶の設定可能な基本フレームを読み取り/書き込みます。フレームは、書き込んだときにチャネル内ですぐには有効になりません。チャネル内の書き込まれたフレームは、以下により計算されます。

- G500 命令(すべての設定可能フレームまたはゼロオフセットを解除)
- G54 ... G57、G505 ... G599 命令(設定可能フレームまたはゼロオフセットを有効化)

有効な設定可能フレーム	データ記憶フレーム	(対応する命令)
$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

6.2 フレームへの値の割り付け

6.2.1 値の直接割り当て(軸値、角度、スケール)

NC プログラムでフレームまたはフレーム変数に直接、値を割り当てることができます。

構文

構文

```
$P_PFRAME = CTRANS(X, <オフセット値>, Y, <オフセット値>, Z, <オフセット値>, ...)
$P_PFRAME = ROT(X, <角度>, Y, <角度>, Z, <角度>, ...)
$P_UIFR[..] = CROT(X, <角度>, Y, <角度>, Z, <角度>,  ...
$P_PFRAME = CSCALE(X, <スケール>, Y, <スケール>, Z, <スケール>, ...)
$P_PFRAME = CMIRROR(X, Y, Z)
```

$P_CHBFRAME[<n>]の構文は$P_PFRAME と同じです。

意味

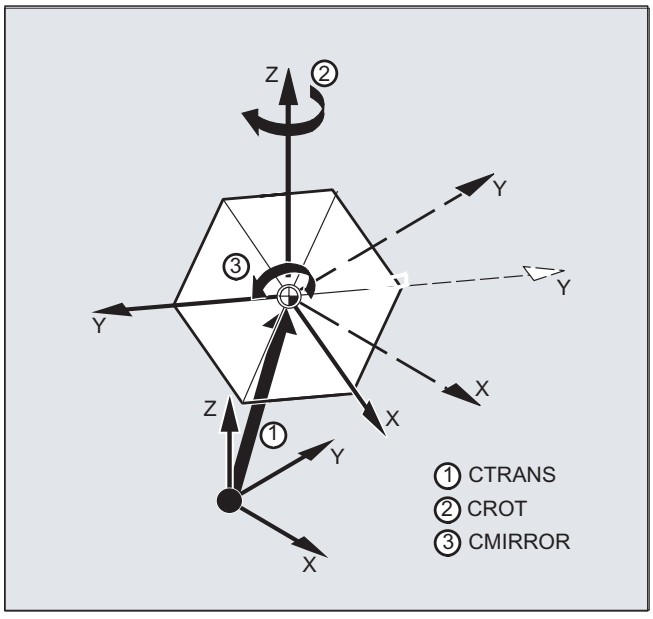
CTRANS:	指定軸の平行移動
CROT:	指定軸を中心とする回転
CSCALE:	指定軸でのスケール変更
CMIRROR:	指定軸の方向逆転
X, Y, Z :	指定ジオメトリ軸の方向のオフセット値
<オフセット値>:	オフセット値
<角度>:	回転角度
<スケール>:	スケール値

例

現在のプログラマブルフレームのフレーム成分への値割り当て

現在のプログラマブルフレームのフレーム成分の移動、回転、ミラーリングのための値割り当て

```
$P_PFRAME = CTRANS(X,10,Y,20,Z,5) :CROT(Z,45) :CMIRROR(Y)
```



フレームの回転成分の書き込み

CROT により、設定可能なデータ記憶フレーム$P_UIFR の回転成分の 3 本の軸すべてに値を割り当てます。

```
$P_UIFR[5] = CROT(X, 0, Y, 0, Z, 0)
```

別の方法として、データ記憶フレームの回転成分の関連する軸に個々の値を直接割り当てます。

```
$P_UIFR[5, Y, RT]=0
$P_UIFR[5, X, RT]=0
$P_UIFR[5, Z, RT]=0
```

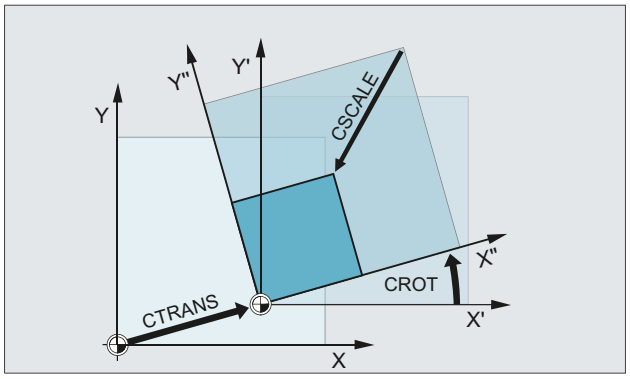
説明

結合演算子:フレームでの複数の操作を相互に結合します。操作は、左から右に連続して処理されます。

例

$P_PFRAME でのオフセット、回転、スケーリングの結合操作

```
$P_PFRAME = CTRANS(...) :CROT(...) :CSCALE...
```



6.2.2 フレーム成分の読み取りと変更(TR、FI、RT、SC、MI)

この機能を使用すると、特定のオフセット値または回転角度など、フレームの個々のデータにアクセスできます。これらの値を変更したり、別の変数に割り当てたりできます。

構文

`R10=$P_UIFR[$P_UIFNUM,X,RT]`	現在有効な設定可能ゼロオフセット$P_UIFRNUMの、X 軸を中心とする回転角度 RT を変数 R10 に割り当てます。
`R12=$P_UIFR[25,Z,TR]`	設定したフレーム No. 25 のデータセットから、Z のオフセット値 TR を変数 R12 に割り当てます。
`R15=$P_PFRAME[Y,TR]`	現在のプログラマブルフレームの Y のオフセット値 TR を変数 R15 に割り当てます。
`$P_PFRAME[X,TR] = 25`	現在のプログラムマブルフレームの X のオフセット値 TR を変更します。X25 は直ちに適用されます。

意味

`$P_UIFRNUM`:	この命令は、現在有効な設定可能ゼロオフセットを基準とした位置を自動的に確立します。
`P_UIFR[n,…,…]`：	フレーム番号 n を指定して、設定可能フレーム No. n にアクセスします。
	読み取る、または変更する成分を指定します。
`TR`:	TR 平行移動

FI:	FI 精密平行移動
RT:	RT 回転
SC:	SC スケール変更
MI:	MI ミラーリング
X, Y, Z:	対応する軸 X、Y、Z も指定されます (例を参照してください)。

RT 回転の値の範囲

1 番目のジオメトリ軸を中心とした-180° ～ +180°
回転:
2 番目のジオメトリ軸を中心とした-90° ～ +90°
回転:
3 番目のジオメトリ軸を中心とした-180° ～ +180°
回転:

説明

フレームの呼び出し

システム変数$P_UIFRNUM を指定すると、$P_UIFR または G54、G55、... で設定した現在のゼロオフセットにアクセスできます
($P_UIFRNUM には、現在設定しているフレーム番号が含まれます)。

保存されている他の設定可能な$P_UIFR フレームはすべて、適切な番号の$P_UIFR[n]を指定して呼び出します。

予約フレーム変数とユーザー定義フレームの場合は、$P_IFRAME などの名称を指定してください。

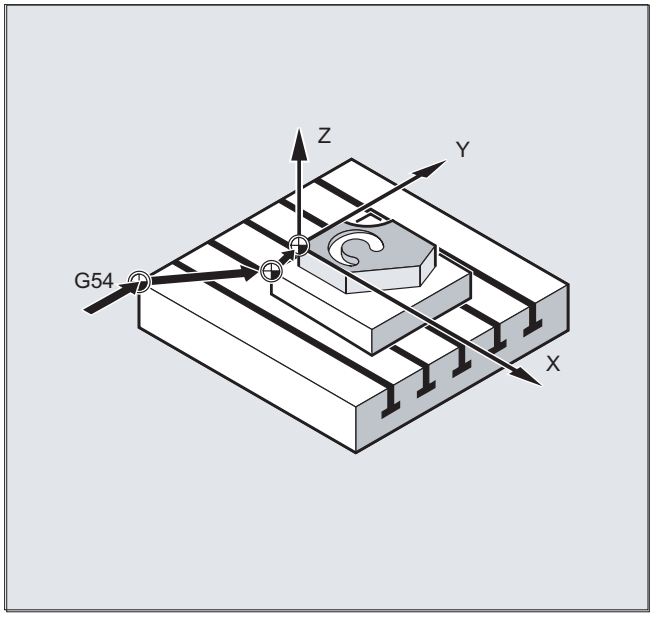
データの呼び出し

アクセスする、または変更する軸名称と値のフレーム成分は、[X, RT]や[Z, MI]のように、角括弧で囲んで記述します。

6.2.3 フレームの接続

フレームを別のフレームに割り当てたり、フレームを NC プログラムで相互に結合することができます。

このフレーム結合は、パレット上に配置され、同じ処理で加工する複数のワークの記述に適しています。



フレーム成分に含めることができるのは、パレット処理の記述のための中間値のみです。これらの値の結合により、さまざまなワーク原点を生成します。

例

割り当て

プログラムコード	コメント
DEF FRAME SETTING_1	; ローカルフレーム変数の定義
SETTING_1 = CTRANS(X,10)	; 関数の結果をフレーム変数に割り当て
$P_PFRAME = SETTING_1	; フレーム変数を現在のフレームに割り当て
DEF FRAME SETTING_4	; ローカルフレーム変数の定義
SETTING_4 = $P_PFRAME	; 現在のフレームをフレーム変数に保存
...	
$P_PFRAME = SETTING_4	; 現在のフレームをフレーム変数から取得

結合

演算子:フレームをプログラム指令順に相互に結合します。オフセットや回転などのフレーム成分が、連続して順に実行されます。

プログラムコード	コメント

`$P_IFRAME = $P_UIFR[15] : $P_UIFR[16]`	`; 2 つの設定可能なデータ記憶フレームの`
	`; 結合による結果フレームを`
	`; 動作中の設定可能な全体フレームに割り当て`
	`; 用途例 :`
	`; $P_UIFR[15]:オフセット`
	`; $P_UIFR[16]:回転`

`$P_UIFR[3] = $P_UIFR[4] : $P_UIFR[5]`	`; 2 つの設定可能なデータ記憶フレーム`
	`; の結合による結果フレームを`
	`; 別の設定可能なデータ記憶フレームに割り当て`

6.2.4 フレーム変数の定義(DEF FRAME)

予約フレーム変数に加えて、ユーザーフレーム変数も定義できます。ユーザー定義のフレーム変数は、FRAME タイプのユーザー変数です。フレームの名称は、ユーザー変数の規則に従って自由に割り当てることができます。

CTRANS、CROT、CSCALE、および CMIRROR は、ユーザー定義のフレーム変数に値を割り当てます。

構文

```
DEF FRAME <名称>
```

意味

DEF FRAME:	FRAME タイプのユーザー変数を定義します。
<名称**>**:	フレーム変数の名称

例

「PALETTE」フレーム変数の定義とオフセット値と回転値の割り当て

プログラムコード	コメント
`DEF FRAME PALETTE`	`; PALETTE フレーム変数を定義`

プログラムコード	コメント
`PALETTE = CTRANS(...) :CROT(...)`	`;` オフセットと回転のために結合の結果フレームを `;` PALETTE フレーム変数に割り当て

6.3 荒削りオフセットと仕上げオフセット(CTRANS、CFINE)

仕上げオフセット

仕上げオフセット CFINE(...)は、次のフレームに適用できます。

- 設定可能フレーム: $P_UIFR または$P_IFRAME
- 基本フレーム: $P_NCBFR[<n>]、$P_CHBFR[<n>]、$P_CHBFRAMES[<n>]または $P_ACTBFRAME
- プログラマブルフレーム: $P_PFRAME

フレームの仕上げオフセットは、CFINE(...)命令によりプログラム指令されます。

荒削りオフセット

荒削りオフセット CTRANS(...)は、すべてのフレームに適用できます。

サムオフセット

サムオフセットは、荒削りオフセットと仕上げオフセットを加算したものです。

マシンデータ

仕上げオフセットの有効化

仕上げオフセットは次のマシンデータで有効にします。

MD18600 $MN_MM_FRAME_FINE_TRANS = 1

構文

仕上げオフセット

- フレーム全体
  - <フレーム> = CFINE(<K_1>,<値>)
  - <フレーム> = CFINE(<K_1>,<値>, <K_2>, <値>)
  - <フレーム> = CFINE(<K_1>,<値>, <K_2>, <値>, <K_3>, <値>)
- フレーム成分
  - <フレーム>[<n>, <K_1>, FI] = <値>

荒削りオフセット

- フレーム全体
  - <フレーム> = CTRANS(<K_1>,<値>)
  - <フレーム> = CTRANS(<K_1>,<値>, <K_2,<値>)
  - <フレーム> = CTRANS(<K_1>,<値>, <K_2,<値>, <K_3,<値>)
- フレーム成分
  - <フレーム>[<n>,<K_1>,TR] = <値>

特にプログラマブルフレーム$P_PFRAME の場合は、次のようになります。

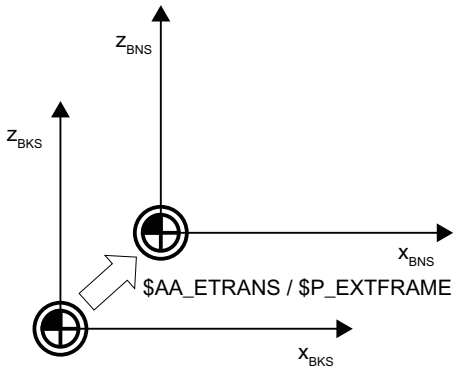
- TRANS <K_1> <値>
- TRANS <K_1> <値> <K_2> <値>
- TRANS <K_1> <値> <K_2> <値> <K_3> <値>

意味

<フレーム>:	フレーム、データ記憶の設定可能フレーム$P_UIFR[<n>]など
CFINE:	仕上げオフセットまたは追加オフセット
CTRANS:	荒削りオフセット、アブソリュートオフセット
TRANS:	プログラマブルフレームのみ:荒削りオフセット、アブソリュートオフセット
<K_n>:	座標軸 X、Y、Z
<値>:	オフセット値

6.4 外部ゼロオフセット($AA_ETRANS)

外部ゼロオフセットは、基本座標系(BCS)と基本ゼロオフセットシステム(BZS)の間の直線オフセットです。



$AA_ETRANS による外部ゼロオフセットは、マシンデータのパラメータ設定に応じて、次の 2 通りの方法で動作します。

1. NC/PLC インタフェース信号による起動後、システム変数$AA_ETRANS は直接オフセット値として動作します。
2. NC/PLC インタフェース信号による起動後、システム変数$AA_ETRANS の値は、有効なシステムフレーム$P:EXTFRAME とデータ記憶フレーム$P_EXTFR に転送されます。動作中の全体フレーム$P_ACTFRAME が再計算されます。

マシンデータ

システム変数$AA_ETRANS との組み合わせでは、次のマシンデータで選択した 2 つの処理は区別されます。

MD28082 $MC_MM_SYSTEM_FRAME_MASK、ビット 1 = <値>

<値>	意味
0	機能: $AA_ETRANS[<軸>]を PLC、HMI、または NC プログラムで直接書き込みます。 次の可能な移動ブロックで$AA_ETRANS[<軸>]のためにゼロオフセットの後退を可能にします:DB31、...DBX3.0
1	機能:有効なシステムフレーム$P:EXTFRAME とデータ記憶フレーム$P_EXTFR を起動します。 $AA_ETRANS[<軸>]のためにゼロオフセットの後退を可能にします:DB31、...DBX3.0。チャネルで以下が実行されます。 ● チャネルのすべての移動動作を停止(コマンド軸と PLC 軸以外) ● 先読み停止とその後の再解析(STOPRE) ● 動作中のフレームの荒削りオフセット$P_EXTFRAME[<軸>] = $AA_ETRANS[<軸>] ● データ記憶フレームの荒削りオフセット$P_EXTFR[<軸>] = $AA_ETRANS[<軸>] ● 動作中の全体フレームの再計算$P_ACTFRAME ● プログラム指令された軸でオフセットを後退 ● 中断されていた NC プログラムの移動動作を再開

プログラミング

- 構文
  ```
  $AA_ETRANS[<軸>] = <値>
  ```
- 意味

$AA_ETRANS:	外部ゼロオフセットを保存するためのシステム変数
<軸>:	チャネル軸
<値>:	オフセット値

NC/PLC インタフェース信号

DB31, ... DBX3.0 = 0 → 1 ⇒ $P_EXTFRAME[<軸>] = $P_EXTFR[<軸>] = $AA_ETRANS[<軸>]

6.5 原点確立状態をリセットして現在値を設定(PRESETON)

PRESETON()は、1 つ以上の軸に対して、機械座標系(MCS)で新しい現在値を設定します。これは、軸の MCS のゼロオフセットに対応します。これで軸が移動することはありません。

PRESETON は、同期による先読み停止を開始します。実際の位置は、停止状態でのみ軸に割り当てられます。

PRESETON の軸がチャネルに割り当てられていない場合、その後の処理は、軸別の軸入れ替え動作の設定によって決まります。

MD30552 $MA_AUTO_GET_TYPE

原点確立状態

機械座標系で新しい現在値を設定することにより、機械軸の原点確立状態がリセットされます。

DB31、... DBX60.4/.5 = 0 (原点確立済み/原点同期済み検出器 1/2)

このため、PRESETON はレファレンス点が不要な軸にのみ使用することをお勧めします。

元の機械座標系に復帰するには、たとえばパートプログラムからレファレンス点復帰動作により(G74)、機械軸の検出器を再度原点確立してください。

⚠ 注意

原点確立状態のリセット

PRESETON により機械座標系で新しい現在値を設定すると、機械軸の原点確立状態が「原点未確立/原点未同期」にリセットされます。

プログラミング

構文

```
PRESETON(<軸_1>, <値_1> [, <軸_2>, <値_2>, ... <軸_8>, <値_8>])
```

意味

PRESETON:	原点確立状態のリセットにより現在値を設定します	
	先読み停止:	あり
	単独ブロック指令:	あり

<軸_x>:	機械軸名称	
	タイプ:	AXIS
	データ範囲:	チャネルで定義された機械軸の名称
<値_x>:	機械座標系(MCS)での機械軸の新しい現在値 入力は現在有効な単位系(インチ/メトリック)でおこなわれます 有効な直径指定(DIAMON)が考慮されます	
	タイプ:	REAL

参照先

NC プログラムでの PRESETONS

NC プログラムでの PRESETON の詳細については、以下を参照してください。

機能マニュアル 基本機能、章「K2:軸、座標系、フレーム」> 「座標系」>「機械座標系(MCS 原点確立状態をリセットして現在値を設定(PRESETON)」

シンクロナイズドアクションでの PRESETONS

シンクロナイズドアクションでの PRESETON の詳細については、以下を参照してください。

機能マニュアル シンクロナイズドアクション、「詳細説明」>「シンクロナイズドアクションでの動作」>「原点確立状態をリセットして現在値を設定(PRESETON)」

6.6 原点確立状態をリセットせずに現在値を設定(PRESETONS)

PRESETONS()は、1 つ以上の軸に対して、機械座標系(MCS)で新しい現在値を設定します。これは、軸の MCS のゼロオフセットに対応します。これで軸が移動することはありません。

PRESETONS は、同期による先読み停止を開始します。実際の位置は、停止状態でのみ軸に割り当てられます。

PRESETONS の軸がチャネルに割り当てられていない場合、その後の処理は、軸別の軸入れ替え動作の設定によって決まります。

MD30552 $MA_AUTO_GET_TYPE

原点確立状態

PRESETONS により機械座標系(MCS)で新しい現在値を設定しても、機械軸の原点確立状態は変更され**ません**。

要件

- **エンコーダタイプ**
  PRESETONS は、次のエンコードタイプの有効な検出器に対してのみ使用できます。
  - MD30240 $MA_ENC_TYPE[<検出器>] = 0 (シミュレーションされるエンコーダ)
  - MD30240 $MA_ENC_TYPE[<検出器>] = 1 (生信号エンコーダ)
- **原点復帰動作モード**
  PRESETONS は、有効な検出器の次の原点復帰動作モードでのみ使用できます。
  - MD34200 $MA_ENC_REFP_MODE[<検出器>] = 0 (可能なレファレンス点復帰なし)
  - MD34200 $MA_ENC_REFP_MODE[<検出器>] = 1 (インクレメンタル検出器、回転検出器、またはリニア検出器のレファレンス点復帰:エンコーダトラックでゼロパルス)

プログラミング

構文

```
PRESETONS(<軸_1>, <値_1> [, <軸_2>, <値_2>, ... <軸_8>, <値_8>])
```

意味

PRESETONS:	原点確立状態をリセットせずに、現在値を設定します	
	先読み停止:	あり
	単独ブロック指令:	あり

<軸_x>:	機械軸名称	
	タイプ:	AXIS
	値の範囲:	チャネルで定義された機械軸の名称
<値_x>:	機械座標系(MCS)の機械軸の新しい現在の実際値 入力は有効な単位系(インチ/メトリック)でおこなわれます 有効な直径指定(DIAMON)が考慮されます	
	タイプ:	REAL

参照先

NC プログラムでの PRESETONS

NC プログラムでの PRESETONS の詳細については、以下を参照してください。

機能マニュアル 基本機能、章「K2:軸、座標系、フレーム」>「座標系」>「機械座標系(MCS)」>「原点確立状態をリセットせずに現在値を設定(PRESETONS)」

シンクロナイズドアクションでの PRESETONS

シンクロナイズドアクションでの PRESETONS の詳細については、以下を参照してください。

機能マニュアル シンクロナイズドアクション、「詳細説明」>「シンクロナイズドアクションでの動作」>「原点確立状態をリセットせずに現在値を設定(PRESETONS)」

6.7 空間の 3 つの計測点からのフレーム計算(MEAFRAME)

MEAFRAME 機能は、計測サイクルのサポートに使用します。これは、最適な 3 点と対応する計測点からフレームを計算します。

ワークが加工位置に位置決めされると、直交機械座標系を基準とした位置は通常、その最適な位置に対してオフセットされて回転されます。正確な加工または計測をおこなうには、高価な部品の調整をおこなうか、パートプログラムで定義した動作を変更する必要があります。

フレームは、空間の 3 点の既知の最適位置のサンプリングで定義できます。サンプリングには、タッチトリガプローブまたは光学センサを使用して、支持板またはプローブボールに正確に取り付けられた特殊な穴に接触します。

構文

```
MEAFRAME(<最適な点>,<計測点>,<品質>)
```

意味

MEAFRAME:	ファンクションコール		
<最適な点>:	2 次元最適な 3 点の座標を含む実配列		
<計測点>:	2 次元計測した 3 点の座標を含む実配列		
<品質>:	フレーム演算の品質に関する情報を返す変数		
	タイプ:	VAR REAL	
	値:	-1	最適な点がほぼ直線上にあります。フレームを計算できません。返された FRAME 変数に不明のフレームが含まれます。
		-2	計測点がほぼ直線上にあります。フレームを計算できません。返された FRAME 変数に不明のフレームが含まれます。
		-4	別の理由で、回転マトリクスを計算できませんでした。
		≥ 0.0	変形量の合計(各点間の距離)。これは計測した三角形を最適な三角形と一致する三角形に変換するために必要です。

注記

計測の品質

計測した座標を最適な座標に、回転と平行移動を使用してマッピングするには、計測した点が形成する三角形が、最適な三角形に一致する必要があります。このためには、計測した三角形を最適な三角形に再形成するために必要な偏差平方和を最小にする補正アルゴリズムを使用します。

計測点の実際の変形量を使用して、計測値の品質を判断できるため、MEAFRAME は追加の変数として、この実際の変形量を返します。

注記

MEAFRAME で作成したフレームは、ADDFRAME 機能によってフレーム結合で別のフレームに変換できます(例「ADDFRAME による結合」を参照してください)。

例

例 1:

パートプログラム 1:

プログラムコード
...
DEF FRAME CORR_FRAME

計測点の設定:

プログラムコード	コメント
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)	; テスト用
DEF REAL FIT_QUALITY=0	
DEF REAL ROT_FRAME_LIMIT=5	; 部品位置を最大 5°だけ回転できます。
DEF REAL FIT_QUALITY_LIMIT=3	; 最適な三角形と計測した三角形の間に最大 3 mm のオフセットが可能です。
DEF REAL SHOW_MCS_POS1[3]	
DEF REAL SHOW_MCS_POS2[3]	
DEF REAL SHOW_MCS_POS3[3]	

プログラムコード	コメント
N100 G01 G90 F5000	
N110 X0 Y0 Z0	
N200 CORR_FRAME=MEAFRAME(IDEAL_POINT,MEAS_POINT,FIT_QUALITY)	
N230 IF FIT_QUALITY < 0	
SETAL(65000)	
GOTOF NO_FRAME	
ENDIF	
N240 IF FIT_QUALITY > FIT_QUALITY_LIMIT	
SETAL(65010)	
GOTOF NO_FRAME	
ENDIF	
N250 IF CORR_FRAME[X,RT] > ROT_FRAME_LIMIT	; 1 番目の RPY 角を制限しています。
SETAL(65020)	
GOTOF NO_FRAME	
ENDIF	
N260 IF CORR_FRAME[Y,RT] > ROT_FRAME_LIMIT	; 2 番目の RPY 角を制限しています。
SETAL(65021)	
GOTOF NO_FRAME	
ENDIF	
N270 IF CORR_FRAME[Z,RT] > ROT_FRAME_LIMIT	; 3 番目の RPY 角を制限しています。
SETAL(65022)	
GOTOF NO_FRAME	
ENDIF	
N300 $P_IFRAME=CORR_FRAME	; サンプルフレームを設定可能フレームで起動します。 ; ジオメトリ軸を最適な点に位置決めしてフレームを確認します。
N400 X=IDEAL_POINT[0,0] Y=IDEAL_POINT[0,1] Z=IDEAL_POINT[0,2]	
N410 SHOW_MCS_POS1[0]=$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	; 設定可能フレームをゼロフレームと同じように解除します (入力値もプリセット値もありません)。

プログラムコード	コメント
No_FRAME	; ゼロフレームでプリセットされるように設定可能フレームを解除します (入力値はありません)。
M0	
M30	

例 2: フレームの結合

オフセットの MEAFRAME の結合

MEAFRAME 機能はオフセットフレームを返します。このオフセットフレームを、機能(G54 など)の呼び出し時に有効であった設定可能フレーム$P_UIFR[1]に結合すると、設定可能フレームが得られ、移動や加工でさらに変換することができます。

ADDFRAME による結合

フレーム結合のこのオフセットフレームを別の位置で適用する場合、または他のフレームが設定可能フレームより前に有効になっている場合は、ADDFRAME 機能を使用して、チャネル基本フレームのいずれかまたはシステムフレームに結合できます。

このフレームでは以下を無効にしてください。

- MIRROR によるミラーリング
- SCALE によるスケーリング

指令値と現在位置の入力パラメータはワーク座標です。これらの座標は、必ずメトリック単位またはインチ単位(G71/G70)で、半径を基準として(DIAMOF)、制御装置の基本単位系で指定してください。

参照先:
ADDFRAME について詳しくは、次の文書を参照してください。
機能マニュアル 基本機能; K2:軸タイプ、座標系、フレーム

6.8 NCU グローバルフレーム

各 NCU のすべてのチャネルに使用する NCU グローバルフレームは 1 セットのみです。NCU グローバルフレームは、すべてのチャネルから読み取りと書き込みを実行できます。NCU グローバルフレームは、それぞれのチャネルで有効にします。

グローバルフレームを使用して、オフセットを含む**チャネル軸と機械軸**のスケーリングとミラーリングを実行できます。

幾何学的関係とフレーム結合

グローバルフレームを使用する場合、各軸間の幾何学的関係はありません。したがって、回転を実行することも、ジオメトリ軸識別子をプログラム指令することもできません。

- グローバルフレームでは、回転を使用できません。回転のプログラミングでは、次のアラームが発生して、拒否されます。「18310 Channel %1 Block %2 Frame:rotation not allowed」(18310 チャネル%1 ブロック%2 フレーム: 回転できません)
- グローバルフレームとチャネル別フレームは結合できます。結果として得られるフレームには、すべての軸の回転を含むすべてのフレーム成分が含まれます。回転成分を含むフレームをグローバルフレームに割り当てると、アラーム「Frame:rotation not allowed」(フレーム: 回転できません)とともに拒否されます。

NCU グローバルフレーム

NCU グローバル基本フレーム$P_NCBFR[n]

8 個までの NCU グローバル基本フレームを設定できます。

チャネル別基本フレームも使用できます。

グローバルフレームは、NCU のすべてのチャネルから読み取りと書き込みを実行できます。グローバルフレームに書き込むときは、チャネル協調をおこなってください。これは、待機マーク(WAITMC)などを使用して実行できます。

工作機械メーカ

グローバル基本フレームの数は、マシンデータを介して設定します。

参照先:
機能マニュアル 基本機能; 軸、座標系、フレーム(K2)

NCU グローバル設定可能フレーム$P_UIFR[n]

すべての設定可能フレーム G500、G54...G599 は、NCU グローバルに、またはチャネル毎に設定できます。

工作機械メーカ

すべての設定可能フレームは、マシンデータ
MD18601 $MN_MM_NUM_GLOBAL_USER_FRAMES を利用して、グローバルフレームとして再設定できます。

チャネル軸識別子と機械軸識別子は、フレームプログラム命令の軸識別子として使用できます。ジオメトリ軸識別子をプログラミングすると、アラームで拒否されます。

6.8.1 チャネル別フレーム($P_CHBFR、$P_UBFR)

設定可能フレームまたは基本フレームは、オペレータ操作と PLC による OPI およびパートプログラムを介して読み書きできます。

グローバルフレームには仕上げオフセットも使用できます。グローバルフレームのマスクは、G53、G153、SUPA と G500 で実行されます。これは、チャネル別フレームの場合も同様です。

工作機械メーカ

チャネル内の基本フレームの数は、マシンデータ MD28081
$MC_MM_NUM_BASE_FRAMES を介して設定できます。標準構成は、チャネル毎に 1 個以上のフレームがあることを前提に構成されています。1 チャネルあたり 8 個までの基本フレームがサポートされています。8 個の基本フレームに加え、8 個の NCU グローバル基本フレームをチャネルで使用できます。

チャネル別フレーム

$P_CHBFR[n]

システム変数$P_CHBFR[n]を使用すると、基本フレームの読み取りと書き込みを実行できます。基本フレームに書き込んだときに、結合された基本フレーム全体は、G500、G54 ～ G599 命令が実行されるまで無効です。この変数は主として HMI または PLC から基本フレームへの書き込み操作を格納するために使用されます。これらのフレーム変数は、データバックアップにより保存されます。

チャネルの 1 番目の基本フレーム

配列インデックスが 0 の基本フレームは、予約変数$P_UBFR への書き込み時に同時に起動されるのではなく、G500、G54...G599 命令の実行時にのみ起動されます。この変数の読み取りと書き込みはプログラムでも実行できます。

$P_UBFR

$P_UBFR は$P_CHBFR[0]と同じです。システム変数に旧バージョンとの互換性があるよう、初期設定では、常に基本フレームがチャネルに 1 個あります。チャネル別基本フ

レームがない場合は、読み取り/書き込み時に次のアラームが発生します。
「Frame:statement not permissible」(フレーム: 命令を実行できません)

6.8.2 チャネルの有効フレーム

チャネルの有効フレームは、パートプログラムから、これらのフレームに関連のあるシステム変数を使用して入力します。これにはシステムフレームも含まれます。現在のシステムフレームの読み取りと書き込みは、これらのシステム変数を使用してパートプログラムで実行できます。

チャネルの現在有効なフレーム

一覧

現在のシステムフレーム	用途:
$P_PARTFRAME	TCARR と PAROT
$P_SETFRAME	現在位置設定と接触計測
$P_EXTFRAME	外部ゼロオフセット
$P_NCBFRAME[n]	現在の NCU グローバル基本フレーム
$P_CHBFRAME[n]	現在のチャネル基本フレーム
$P_BFRAME	現在の、チャネルの 1 番目の基本フレーム
$P_ACTBFRAME	基本フレーム全体
$P_CHBFRMASK と$P_NCBFRMASK	基本フレーム全体
$P_IFRAME	現在の設定可能フレーム
現在のシステムフレーム	用途:
$P_TOOLFRAME	TOROT と TOFRAME
$P_WPFRAME	ワークの基準点
$P_TRAFRAME	座標変換
$P_PFRAME	現在のプログラマブルフレーム
現在のシステムフレーム	用途:
$P_CYCFRAME	サイクル
P_ACTFRAME	現在のフレーム全体
フレーム結合	現在のフレームが基本フレーム全体から構成されます

$P_NCBFRAME[n] 現在の NCU グーバル基本フレーム

システム変数$P_NCBFRAME[n]を使用して、現在のグローバル基本フレームのフィールド要素を読み書きできます。結果として得られる基本フレーム全体は、チャネルでの書き込み処理により計算されます。

変更されたフレームは、そのフレームがプログラムされたチャネル内でのみ起動されます。NCU のすべてのチャネルについてフレームを変更する場合は、$P_NCBFR[n]と$P_NCBFRAME[n]を同時に書き込んでください。そのあと、その他のチャネルで G54 などでフレームを起動します。基本フレームに書き込むたびに、基本フレーム全体が再計算されます。

$P_CHBFRAME[n] 現在のチャネル基本フレーム

システム変数$P_CHBFRAME[n]を使用して、現在のチャネル基本フレームのフィールド要素を読み書きできます。結果として得られる基本フレーム全体は、チャネルでの書き込み処理により計算されます。基本フレームに書き込むたびに、基本フレーム全体が再計算されます。

$P_BFRAME 現在の、チャネルの 1 番目の基本フレーム

予約フレーム変数$P_BFRAME を使用して、チャネル内で有効な配列インデックスが 0 の現在の基本フレームを、パートプログラム内で読み書きできます。書き込んだ基本フレームは、直ちに計算されます。

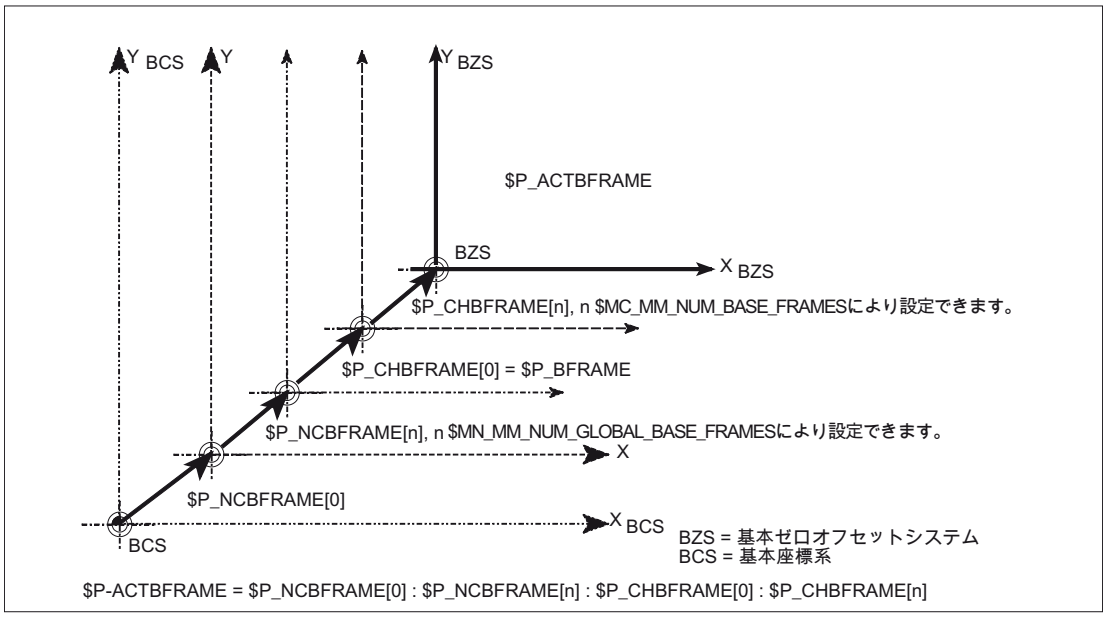
$P_BFRAME は$P_CHBFRAME[0]と同じです。システム変数には常に、有効な初期値を保持します。チャネル別基本フレームがない場合は、読み取り/書き込み時に次のアラームが発生します。「Frame:statement not permissible」(フレーム: 命令を実行できません)

$P_ACTBFRAME 基本フレーム全体

$P_ACTFRAME 変数は、結合された基本フレーム全体を特定します。この変数は読み取り専用です。

$P_ACTFRAME は、以下に対応します。

$P_NCBFRAME[0] : ... : $P_NCBFRAME[n] : $P_CHBFRAME[0] : ... : $P_CHBFRAME[n]



$P_CHBFRMASK と$P_NCBFRMASK 基本フレーム全体

システム変数$P_CHBFRMASK と$P_NCBFRMASK を使用して、どの基本フレームを基本フレーム「全体」の計算に含めるかを選択できます。これらの変数はプログラム内にのみプログラム指令でき、OPI を介して読み取ることができます。変数の値はビットマスクとして解釈され、$P_ACTFRAME のどの基本フレームフィールドの要素が計算に含まれるかを指定します。

$P_CHBFRMASK を使用して、どのチャネル別基本フレームを計算に含めるかを指定できます。また、$P_NCBFRMASK を使用して、どの NCU グローバル基本フレームを計算に含めるかを指定できます。

変数をプログラム指令すると、基本フレーム全体とフレーム全体が再計算されます。リセット後および初期設定では、$P_CHBFRMASK と$P_NCBFRMASK の値は次のようになります。

$P_CHBFRMASK = $MC_CHBFRAME_RESET_MASK

$P_NCBFRMASK = $MC_CHBFRAME_RESET_MASK

例:

```
$P_NCBFRMASK = 'H81' ;$P_NCBFRAME[0] : $P_NCBFRAME[7]
$P_CHBFRMASK = 'H11' ;$P_CHBFRAME[0] : $P_CHBFRAME[4]
```

$P_IFRAME 現在の設定可能フレーム

予約フレーム変数$P_IFRAME を使用して、チャネル内で有効な現在の設定可能フレームを、パートプログラム内で読み書きできます。書き込んだ設定可能フレームは、すぐに計算されます。

NCU グローバル設定可能フレームの場合、変更されたフレームは、そのフレームがプログラムされたチャネル内でのみ機能します。NCU のすべてのチャネルについてフレームを変更する場合は、P_UIFR[n]と$P_IFRAME を同時に書き込んでください。それから、その他のチャネルで G54 などで対応するフレームを起動してください。

$P_PFRAME 現在のプログラマブルフレーム

$P_PFRAME は、TRANS/ATRANS、G58/G59、ROT/AROT、SCALE/ASCALE、MIRROR/AMIRROR のプログラム指令の結果、またはプログラマブルフレームへの CTRANS、CROT、CMIRROR、CSCALE の割り当ての結果のプログラマブルフレームです。

設定可能ゼロオフセットシステム(SZS)とワーク座標系(WCS)の間の位置関係を確立する現在のプログラマブルフレーム変数。

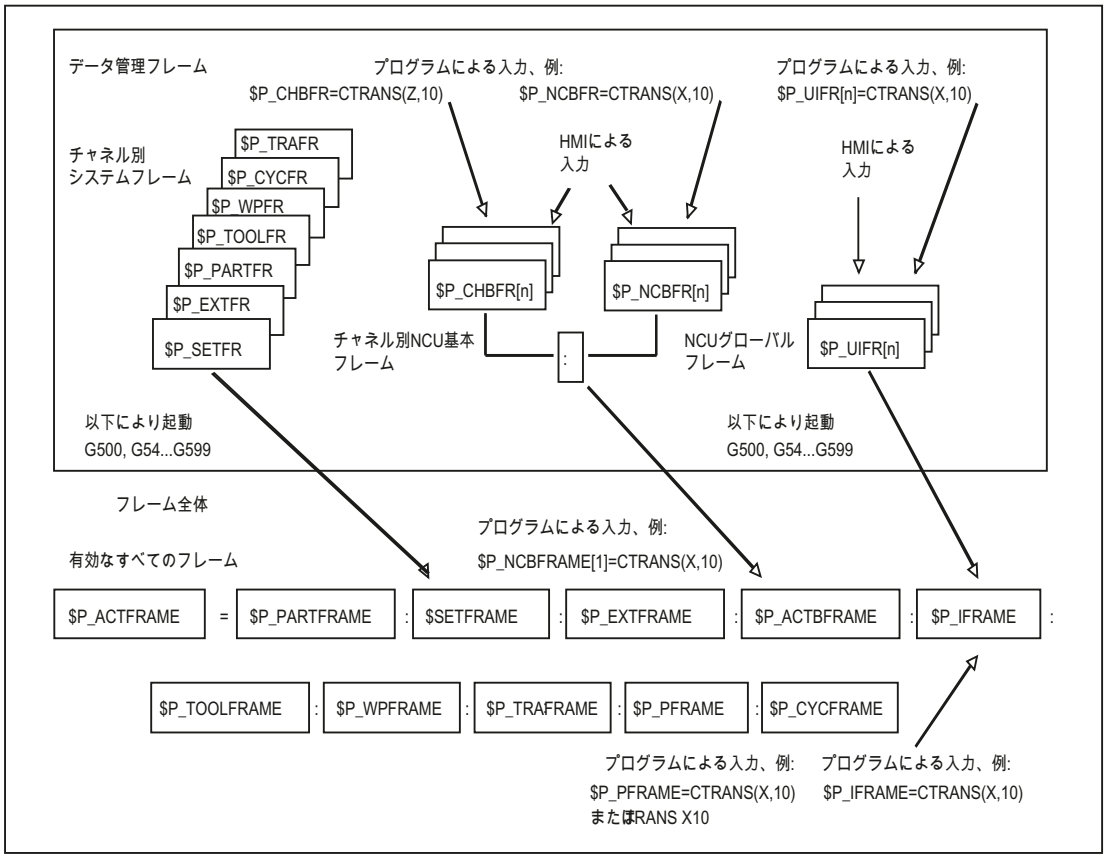
P_ACTFRAME 現在のフレーム全体

結果として得られる現在のフレーム全体$P_ACTFRAME は、基本フレームのすべて、現在の設定可能フレーム、およびプログラマブルフレームを結合したものです。現在のフレームは、フレーム成分の変更のたびに必ず更新されます。

$P_ACTFRAME は、以下に対応します。

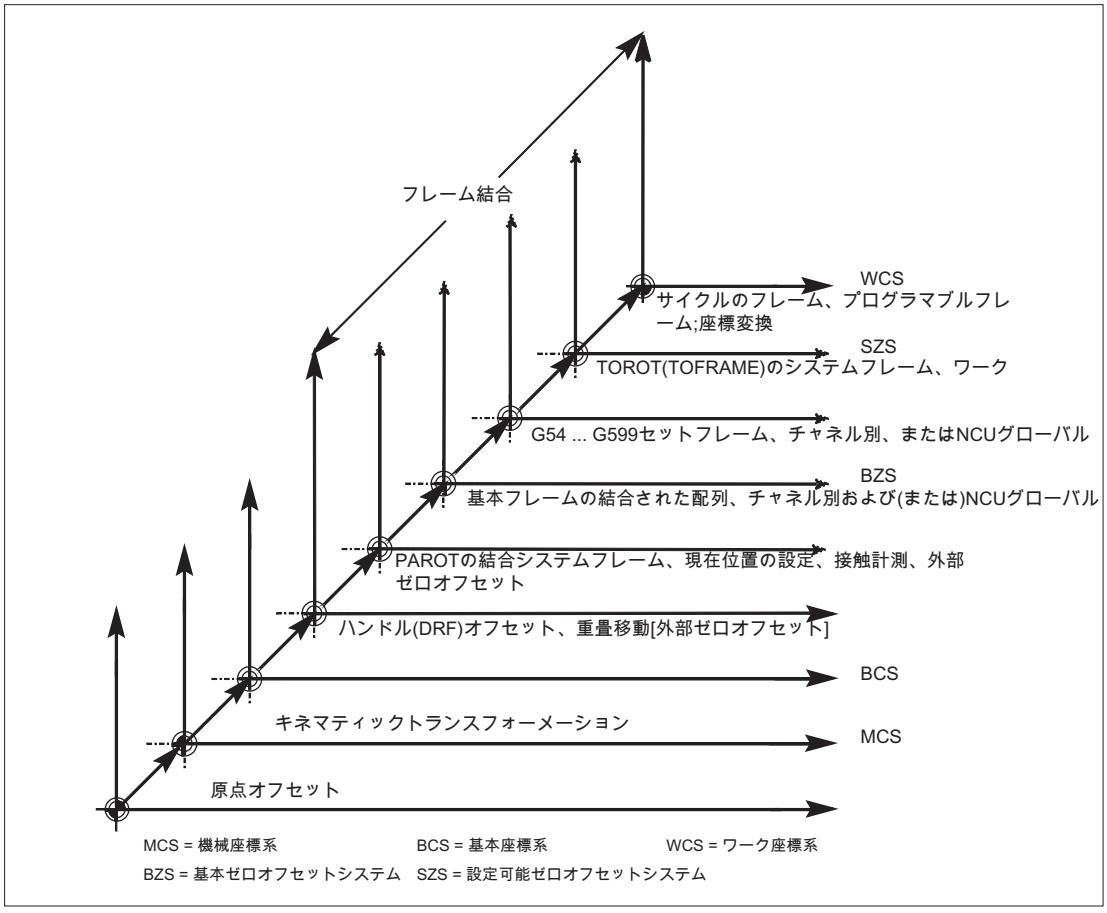
```
$P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME : $P_ACTBFRAME :
$P_IFRAME :

$P_TOOLFRAME : $P_WPFRAME : $P_TRAFRAME : $P_PFRAME : $P_CYCFRAME
```


データ管理フレーム
プログラムによる入力、例:
$P_CHBFR=CTRANS(Z,10)
$P_NCBFR=CTRANS(X,10)
プログラムによる入力、例:
$P_UIFR[n]=CTRANS(X,10)
チャネル別
システムフレーム
$P_TRAFR
$P_CYCFR
$P_WPFR
$P_TOOLFR
$P_PARTFR
$P_EXTFR
$P_SETFR
HMIによる
入力
HMIによる
入力
$P_CHBFR[n]
$P_NCBFR[n]
$P_UIFR[n]
チャネル別NCU基本
フレーム
:
NCUグローバル
フレーム
以下により起動
G500, G54...G599
以下により起動
G500, G54...G599
フレーム全体
有効なすべてのフレーム
プログラムによる入力、例:
$P_NCBFRAME[1]=CTRANS(X,10)
$P_ACTFRAME
=
$P_PARTFRAME
:
$SETFRAME
:
$P_EXTFRAME
:
$P_ACTBFRAME
:
$P_IFRAME
:
$P_TOOLFRAME
:
$P_WPFRAME
:
$P_TRAFRAME
:
$P_PFRAME
:
$P_CYCFRAME
プログラムによる入力、例:
$P_PFRAME=CTRANS(X,10)
またはRANS X10
プログラムによる入力、例:
$P_IFRAME=CTRANS(X,10)

フレーム結合

現在のフレームは、上記で指定した現在のフレーム全体に従って、基本フレーム全体、設定可能フレーム、システムフレーム、およびプログラマブルフレームから構成されています。



座標変換 7

7.1 座標変換タイプの一般的なプログラミング

一般機能

さまざまな機械のキネマティックスにコントローラを適合させるために、適切なパラメータを使って座標変換タイプをプログラム指令できます。 これらのパラメータを使用すると、空間の工具の向きと回転軸の旋回移動の両方を、選択した座標変換に合わせて宣言できます。

3 軸、4 軸、および 5 軸座標変換では、プログラム指令した位置データが常に工具の先端と対応し、空間でその工具は加工される平面に直角に追従します。 直交座標系は、基本座標系から機械座標系に変換され、ジオメトリ軸に対応します。 これらにより、動作点が記述されます。 空間の工具の向きを記述する仮想回転軸は、TRAORI でプログラム指令します。

キネマティックトランスフォーメーションの場合は、位置を直交座標系でプログラム指令できます。 コントローラは、TRANSMIT、TRACYL 、および TRAANG でプログラム指令した直交座標系の移動動作を、実際の機械軸の移動動作に割り当てます。

プログラミング

3 軸、4 軸、5 軸座標変換(TRAORI)

宣言した方向座標変換は、TRAORI 命令と、使用可能な 3 個のパラメータ(座標変換番号、配向ベクトル、回転軸オフセット)で有効にします。

TRAORI (座標変換番号、配向ベクトル、回転軸オフセット)

キネマティックトランスフォーメーション

TRANSMIT (座標変換番号) で宣言した座標変換は、キネマティックトランスフォーメーションの例です。

TRACYL (作業直径, 座標変換番号)

TRAANG (オフセット軸の角度, 座標変換番号)

動作中の座標変換の解除

TRAFOOF を使用すると、現在動作中の座標変換を解除できます。

方向座標変換

3 軸、4 軸、5 軸座標変換(TRAORI)

機械の作業領域の空間に構成された表面を最適に加工するために、工作機械には、3 つの直線軸 X、Y、および Z に加え、追加の軸が必要です。空間の向きを記述するこれらの付加軸は、後続の章で旋回軸と呼ばれます。 これらは、さまざまなキネマティックスを備えた 4 種類の機械で回転軸として使用できます。

1. 固定式工具テーブル上の直線軸に平行な 1 つの回転軸付きカルダン式工具ヘッドなどの 2 軸スイベルヘッド。
2. 2 つの軸を中心に回転可能な工具テーブル付き固定式スイベルヘッドなどの 2 軸回転テーブル。
3. 1 つの軸を中心に回転可能な 1 基の工具テーブル用の 1 つの回転工具付き回転式スイベルヘッドなどの単独軸スイベルヘッド、および単独軸回転テーブル。
4. 1 つの軸を中心に回転可能な工具テーブル上にあり、工具自身を中心に回転可能な工具付きの 1 基の回転式スイベルヘッドなどの、2 軸スイベルヘッド、および単独軸回転テーブル。

3 軸と 4 軸座標変換は 5 軸座標変換の特殊なタイプで、5 軸座標変換と同じ方法でプログラム指令します。

「**汎用 3 軸/4 軸/5 軸/6 軸座標変換**」の対象機能は、直交回転軸の座標変換とユニバーサルミリングヘッドの座標変換の両方に対応しており、他のすべての方向座標変換と同様に、これらの 4 種類の機械に対しても TRAORI で有効にできます。 汎用 5 軸/6 軸座標変換では、工具オリエンテーションにさらに 3 次元の自由度があり、これによって工具を、空間で必要な向きになるよう、工具方向に対応する工具軸を中心として回転できます。

参照先: /FB3/機能マニュアル応用機能; 3 軸～ 5 軸座標変換(F2)

キネマティックスに影響しない工具オリエンテーションの初期設定

ORIRESET

TRAORI による方向座標変換が動作中の場合は、ORIRESET を使用して、3 つまでの旋回軸の初期設定を任意選択パラメータ A、B、C で指定できます。プログラム指令パラメータを回転軸に割り当てる順序は、座標変換で定義した旋回軸の順序により異なります。 ORIRESET(A, B, C)をプログラム指令すると、各旋回軸がその現在の位置から指定した初期設定位置へと、直線的に同期しながら移動します。

キネマティックトランスフォーメーション

TRANSMIT と TRACYL

旋削機械のフライス加工の場合:

1. TRANSMIT による回転式チャックでの正面加工、または
2. TRACYL による、円筒上の任意の軌跡の溝加工

宣言した座標変換に対して、上記のいずれかをプログラム指令できます。

TRAANG

傾斜切り込みの切り込み軸のオプション設定が必要な場合(研削加工の場合など)は、TRAANG を使用して、宣言した座標変換に対して設定可能な角度をプログラム指令できます。

直交 PTP 移動

キネマティックトランスフォーメーションには、8 つまでの異なる関節継手位置 STAT= をプログラム指令できる、「直交 PTP 移動」と呼ばれる移動も含まれます。 位置は直交座標系でプログラム指令しますが、機械の移動は機械座標でおこなわれます。

参照先:
/FB2/機能マニュアル 上級機能; キネマティックトランスフォーメーション(M1)

座標変換重畳

2 つの座標変換は、連続して切り替えることができます。 ここで結合された 2 番目の座標変換では、軸の移動部分は 1 番目の座標変換から取得されます。

1 番目の変換は、次のいずれかです。

- 方向座標変換 TRAORI
- 極座標変換 TRANSMIT
- 円筒座標変換 TRACYL
- 傾斜軸座標変換 TRAANG

2 番目の座標変換には、傾斜軸用の TRAANG タイプの座標変換を使用してください。

7.1.1 座標変換の旋回移動

移動と旋回移動

プログラム指令の向きの移動動作は、主に機械の種類で特定されます。TRAORI による3 軸、4 軸、5 軸タイプの座標変換の場合は、回転軸、または直線軸まわりの回転で工具の旋回移動を記述します。

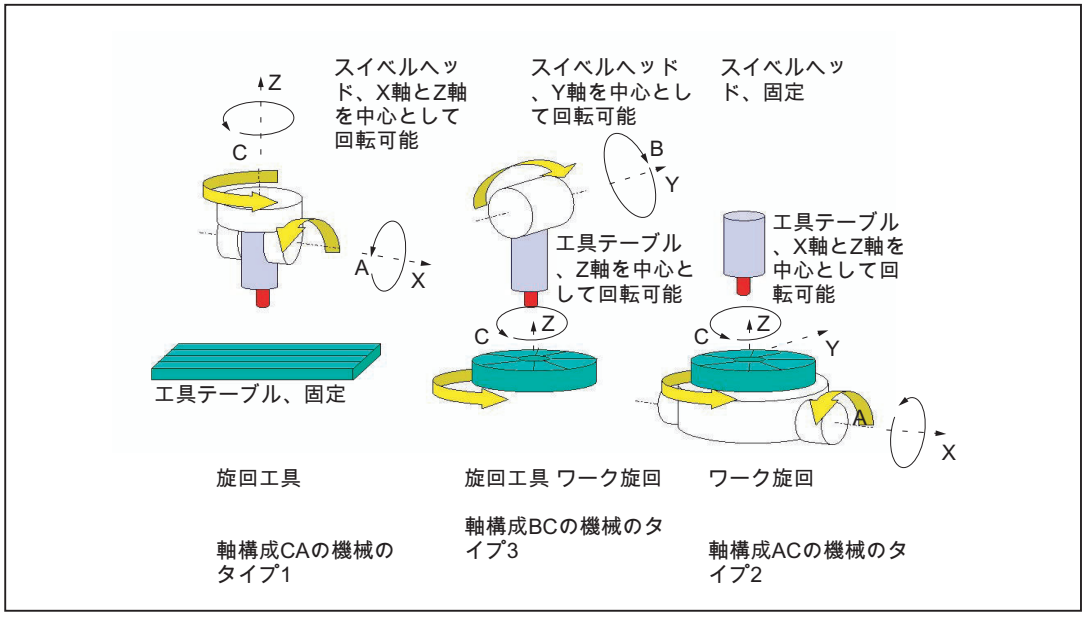
方向座標変換に関わる回転軸の位置の変更により、その他の機械軸で補正移動が発生します。工具先端の位置はそのまま変更されません。

工具の旋回移動は、用途に合わせて仮想軸の回転軸識別子 A...、B...、C...を使用してプログラム指令できます。この場合、オイラー角または RPY 角、方向ベクトルまたは面法線ベクトル、円錐の回転軸、または円錐面上の中間旋回用に正規化ベクトルを入力します。

TRANSMIT、TRANSMIT、TRACYL 、および TRAANG によるキネマティックトランスフォーメーションの場合は、コントローラが、プログラム指令した直交座標系の移動を、実際の機械軸の移動に割り当てます。

3 軸、4 軸、5 軸座標変換(TRAORI)の機械のキネマティックス

工具または工具テーブルは、最大 2 つの回転軸を使用して回転できます。スイベルヘッドと回転テーブル(いずれも単独軸のもの)の組合わせも可能です。



機械のタイプ	方向のプログラミング
3 軸座標変換の機械のタイプ 1 とタイプ 2	工具オリエンテーションのプログラミングは、回転軸に垂直な平面上のみです。 **2 つの**平行移動の軸(直線軸)、および **1 つの**回転の軸(回転軸)があります。
4 軸座標変換の機械のタイプ 1 と 2	工具オリエンテーションのプログラミングは、回転軸に垂直な平面上のみです。 **3 つの**平行移動の軸(直線軸)、および **1 つの**回転の軸(回転軸)があります。
5 軸座標変換機械タイプ 3 単独軸スイベルヘッド、および単独軸回転テーブル	方向座標変換のプログラミング **3 つの**直線軸、および **2 つの**直交する回転軸によるキネマティックスです。 回転軸は、3 つの直線軸の 2 つに平行です。2 つの直交する直線軸により、1 番目の回転軸が移動します。これにより、3 つ目の直線軸が工具と一緒に回転します。2 番目の回転軸は、ワークを回転します。

汎用 5/6 軸座標変換

機械のタイプ	方向座標変換のプログラミング
汎用 5/6 軸座標変換機械タイプ 4 工具自身を中心に回転する工具付き 2 軸スイベルヘッド、および単独軸回転テーブル	方向座標変換のプログラミング **3 つの**直線軸、および **3 つの**直交する回転軸によるキネマティックス。 回転軸は、3 つの直線軸の 2 つに平行です。2 つの直交する直線軸により、1 番目の回転軸が移動します。これにより、3 つ目の直線軸が工具と一緒に回転します。2 番目の回転軸は、ワークを回転します。基本の工具オリエンテーションは、THETA 回転角度による、工具自身を中心とした追加回転でもプログラム指令できます。

「汎用 3 軸、4 軸、および 5/6 軸座標変換」を呼び出すと、基本の工具オリエンテーションをおこなうこともできます。回転軸の方向に関する制限事項は、ここでは適用されません。回転軸が互いに正確に垂直ではないか、既存の回転軸が直線軸に正確に平行はでない場合は、「汎用 5/6 軸座標変換」により、工具オリエンテーションに関しては良好な結果が得られます。

キネマティックトランスフォーメーション TRANSMIT、TRACYL、および TRAANG

旋削機械のフライス加工、または研削中に傾斜切り込みに設定できる軸の場合は、初期設定では宣言している座標変換に従って、スレーブ軸の配置が適用されます。

TRANSMIT	極座標変換の適用
回転式チャックでの正面加工	回転軸 回転軸に垂直な切り込み軸 回転軸に平行な長手軸

TRACYL	円筒補間の適用
円筒上の任意の軌跡の溝加工	回転軸 回転軸に垂直な切り込み軸 回転軸に平行な長手軸

TRAANG	傾斜軸座標変換の適用
傾斜切り込み軸による加工	回転軸 パラメータ設定可能な角度の切り込み軸 回転軸に平行な長手軸

直交 PTP 移動

機械は機械座標で移動し、以下を使用してプログラム指令します。

TRAORI	座標変換の適用
PTP ポイントツーポイント移動	直交座標系(MCS)位置へのアプローチ
CP	BCS の直交軸の軌跡移動
STAT	関節継手の位置は座標変換により異なります
TU	軸が最短軌跡で移動するときの角度

汎用 5 軸/6 軸座標変換による PTP 移動

機械が、機械座標と工具オリエンテーションを使用して移動します。この移動は、キネマティックスまたは方向ベクトルに関係なく、回転軸の位置、およびオイラー角ベクトルと RPY 角ベクトルのは両方を使用してプログラム指令できます。

このような場合は、回転軸補間、大円弧補間のベクトル補間、または円錐面の配向ベクトルの補間が可能です。

例:ユニバーサルミリングヘッドでの 3 軸～ 5 軸座標変換

工作機械には、軸が 5 軸以上あります。

- 直線移動のための 3 つの平行移動軸。この軸で、作業領域の任意の位置に動作点を移動します。
- 設定可能な角度(通常は 45°)に配置された 2 つの回転する旋回軸。この軸で、45°刻みで半球に制限されている空間内の位置で工具を旋回します。

7.1.2 方向座標変換 TRAORI の一覧

TRAORI と組み合わせて使用できるプログラミングタイプ

機械のタイプ	動作中の座標変換 TRAORI によるプログラミング
機械タイプ 1、2、または 3 の 2 軸スイベルヘッドまたは 2 軸回転テーブル、または単独軸スイベルヘッドと単独軸回転テーブルの組合わせです。	旋回軸の軸の順序と工具の旋回方向は、機械のキネマティックスに応じて、マシンデータを使用して **機械別**基準で設定するか、 または、機械のキネマティックスとは無関係に、 **ワーク別**基準で変更可能な向きに 設定することができます。 基準系の旋回軸の回転方向は、次のようにプログラム指令します。 - ORIMKS 基準系=機械座標系 - ORIWKS 基準系=ワーク座標系 初期設定は ORIWKS です。 旋回軸は、次のようにプログラミングします。 直接の機械軸位置 A、B、C A2、B2、C2 仮想軸のプログラム指令角度 - オイラー角(標準)による ORIEULER - RPY 角による ORIRPY - 1 番目の定義の仮想旋回軸による ORIVIRT1 - 2 番目の定義の仮想旋回軸による ORIVIRT2 これは次の補間タイプ間で違いがあります: **直線補間** - 旋回軸または機械軸の ORIAXES **大半径円弧補間**(配向ベクトルの補間) - 旋回軸からの ORIVECT ベクトル成分の A3、B3、C3(方向/面法線) の指定による旋回軸のプログラミング 得られた工具オリエンテーションのプログラミング ブロック始点の面法線ベクトルの A4、B4、C4 ブロック終点で面に垂直なベクトルの A5、B5、C5 LEAD 工具オリエンテーションのリード角 TILT 工具オリエンテーションの傾斜角

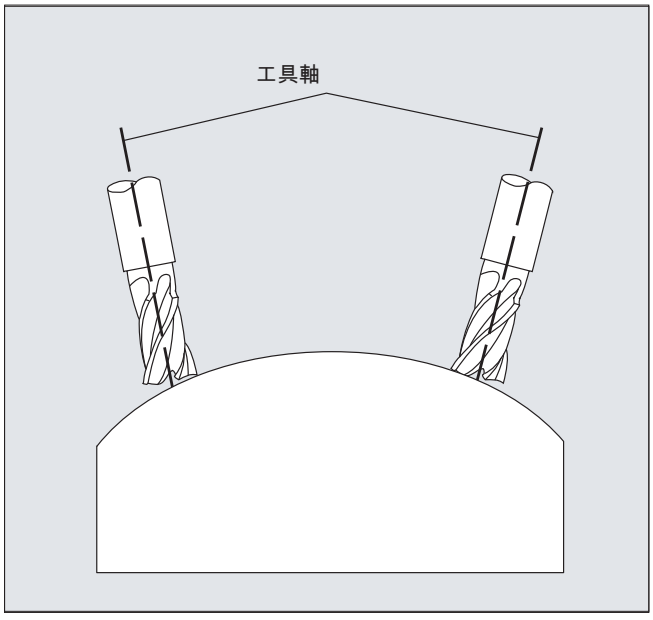
機械のタイプ	動作中の座標変換 TRAORI によるプログラミング
	円錐面の配向ベクトルの補間 次の補間を使用した、**空間の任意の場所で**円錐面への向きの変更: - 平面での ORIPLANE(大半径円弧補間) - 円錐面での右回り方向への ORICONCW - 円錐面での左回り方向への ORICONCCW 、A6、B6、C6 方向ベクトル(円錐の回転軸) -円錐面での OICONIO 補間で、 A7、B7、C7 中間ベクトル(最初または最終の向き)による、または - 円錐面での ORICONTO、接線方向の遷移 以下による、**軌跡に対する向きの**変更 - ORICURVE 以下による 2 つの接点の移動の指定 PO[XH]=(xe, x2, x3, x4, x5)旋回多項式(5 次以下) PO[YH]=(ye, y2, y3, y4, y5)旋回多項式(5 次以下) PO[ZH]=(ze, z2, z3, z4, z5)旋回多項式(5 次以下) - ORIPATHS 後退移動のときの工具の方向、および軌跡長に対応する工具の A8、B8、C8 の再旋回状態の 旋回処理のスムージング。

機械のタイプ	動作中の座標変換 TRAORI によるプログラミング
機械タイプ 1 と 3 付加工具自身がその工具を中心として回転する、その他の機械タイプには、3 番目の回転軸が必要です。 方向座標変換は汎用 6 軸座標変換などです。 配向ベクトルの回転。	工具オリエンテーションの回転のプログラミング LEAD 角、面法線ベクトルに対する角度 PO[PHI] 5 次までの多項式のプログラミング TILT 軌跡タンジェント(Z 方向)を中心とした回転角度 PO[PSI] 5 次までの多項式のプログラミング THETA 回転角度(Z の工具方向を中心とした回転) THETA=ブロック終点の到達値 THETA=AC(...)指令へのアブソリュートノンモーダル切り替え THETA=IC(...)結合指令へのノンモーダル切り替え THETA=Θ_e プログラム指令角度 G90/G91 の補間 PO[THT]=(..) 5 次までの多項式のプログラミング 回転ベクトルのプログラミング - ORIROTA アブソリュート回転ベクトル - ORIROTR 相対回転ベクトル - ORIROTT 接線方向の回転ベクトル
軌跡に対する向きの変更のための軌跡に対する旋回、または軌跡の接線方向の回転ベクトルの回転	以下による、軌跡に対する向きの 変更 - ORIPATH 軌跡に対する工具オリエンテーション - ORIPATHS 旋回処理が不連続変化した場合の回転ベクトルのプログラミング - ORIROTC 接線方向の回転ベクトル、軌跡タンジェントに対する回転

7.2 3 軸、4 軸、5 軸座標変換(TRAORI)

7.2.1 ユニバーサル工具ヘッドの一般的な関係

3 次元曲線の加工で最適な切削条件を得るために、工具の設定角度を変えることができます。

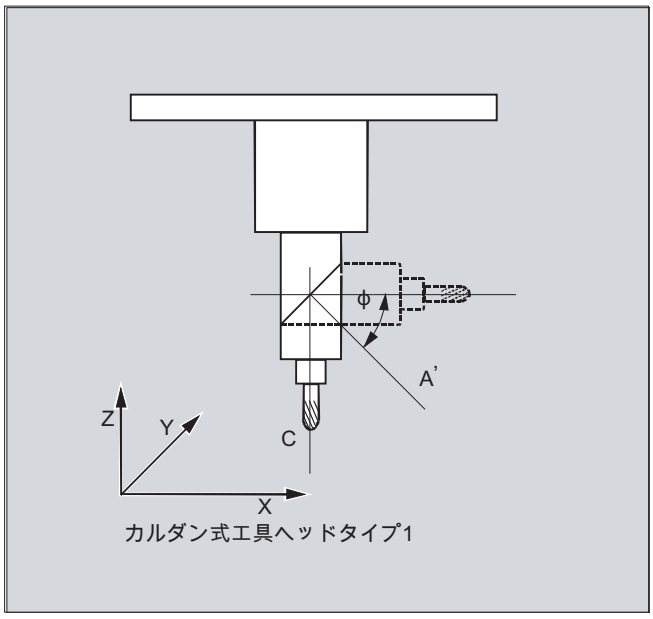


これを可能にする機械構成が軸データに格納されています。

5 軸座標変換

カルダン式工具ヘッド

ここで、3 つの直線軸(X、Y、Z)および 2 つの旋回軸(C、A)は、工具の設定角度と動作点を定義します。2 つの旋回軸の 1 つは傾斜軸として作成し、ここでの A'の例では、多くの場合、45°の角度で配置します。

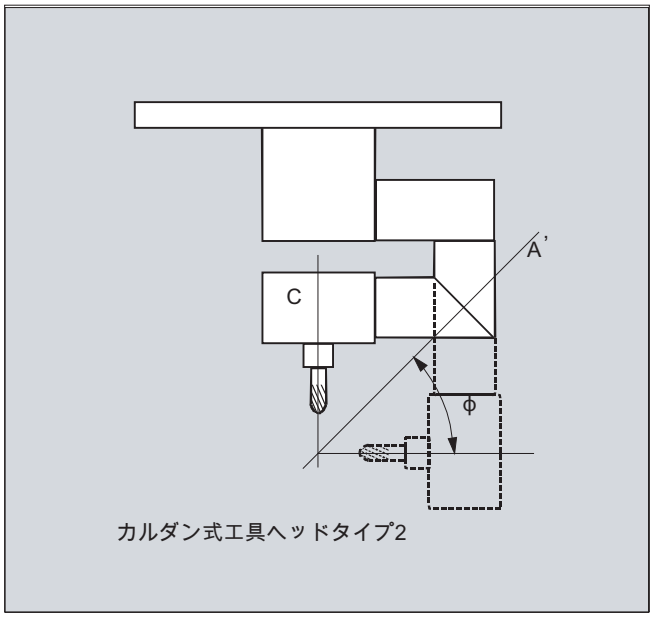


カルダン式工具ヘッドタイプ1

ここで示す例では、その配置は、カルダン式工具ヘッド付き CA タイプの機械のキネマティックスで説明されています。

工作機械メーカ

旋回軸の軸の順序と工具の旋回方向は、機械のキネマティックスに合わせてマシンデータを使用して設定できます。



カルダン式工具ヘッドタイプ2

この例では、A'は X 軸に対して角度 φ の位置にあります。

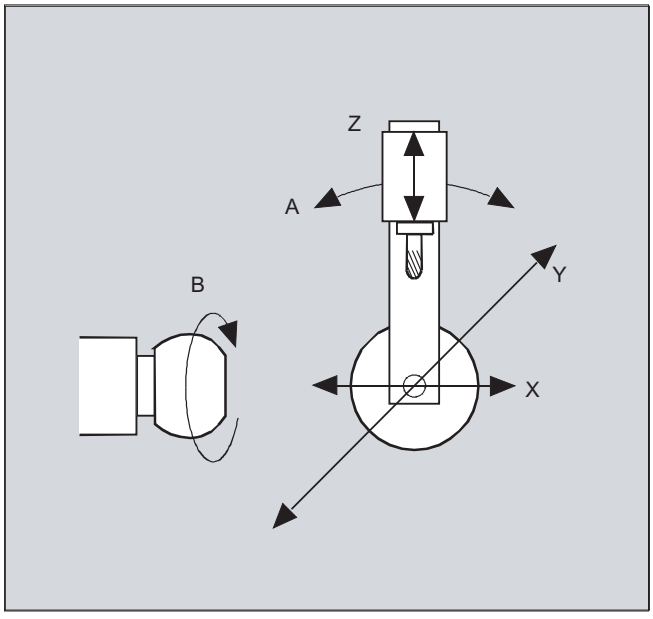
次の関係が一般的に有効です。

A'は X 軸に対して角度 φ の位置にあります　　X 軸
B'は Y 軸に対して角度 φ の位置にあります　　Y 軸
C'は Z 軸に対して角度 φ の位置にあります　　Z 軸

角度 φ は、マシンデータを使用して、0°～+89°の範囲で設定できます。

直線軸旋回機構の場合

これは、移動するワークと移動する工具の配置です。キネマティックスは、3 つの直線軸(X、Y、Z)、および 2 つの直交する回転軸から成ります。たとえば、1 番目の回転軸が 2 つの直線軸の合成スライドで移動し、工具は 3 番目の直線軸に平行に停止しているとします。2 番目の回転軸がワークを回転します。3 番目の直線軸(旋回軸)は合成スライドの平面にあります。



回転軸の軸の順序と工具の旋回方向は、機械のキネマティックスに合わせてマシンデータを使用して設定できます。

以下の関係が可能です。

軸:	**軸の順序:**
1. 回転軸	A A B B C C
2. 回転軸	B C A C A B
直線軸旋回機構	Z Y Z X Y X

工具の旋回方向について設定可能な軸の順序について詳しくは、以下を参照してください。

参照先: /FB3/機能マニュアル 応用機能; 3 ～ 5 軸座標変換(F2)、ユニバーサルミリングヘッド、「パラメータ設定」の章

7.2.2 3 軸、4 軸、5 軸座標変換(TRAORI)

2 つまたは 3 つの平行移動軸、および 1 つの回転軸は、次のように構成できます。座標変換では、回転軸が旋回平面と直交していることが前提です。

工具の旋回は回転軸に垂直な平面上でのみ可能です。座標変換は、可動式工具と可動式ワークの機械タイプをサポートしています。

3 軸と 4 軸座標変換は、5 軸座標変換と同じ方法で設定され、プログラム指令します。

参照先:
機能マニュアル 応用機能; 多軸座標変換(F2)

構文

```
TRAORI(<n>)
TRAORI(<n>,<X>,<Y>,<Z>,<A>,<B>)
TRAFOOF
```

意味

TRAORI:	1 番目に指定した方向座標変換を起動します	
TRAORI(<n>):	n で指定した方向座標変換を起動します	
<n>:	座標変換の番号	
	値:	1 または 2
	例: TRAORI(1)は方向座標変換 1 を有効にします	
<X>,<Y>,<Z>:	工具が指す配向ベクトル成分	
<A>,<B>:	回転軸のプログラマブルオフセット	
TRAFOOF:	座標変換の解除	

工具の向き

工具に対して選択された旋回方向に応じて、工具長補正が工具オリエンテーションの方向に働くように、有効な作業平面(G17、G18、G19)を NC プログラムで設定してください。

注記

座標変換が有効なときは、位置決めデータ(X、Y、Z)は常に工具の先端に対応しています。座標変換に関わる回転軸の位置を変更すると、それに合わせて、残りの機械軸が補正移動をおこないます。このため、工具先端の位置は変わりません。

方向座標変換は常に、工具先端から工具アダプタまでを示します。

旋回軸のオフセット

方向座標変換を有効にすると、旋回軸に対して追加オフセットを直接プログラム指令できます。

正しい順序でプログラム指令している場合は、パラメータを省略できます。

例:

`TRAORI(, , , ,A,B)` (オフセットを 1 つだけ入力する場合)

直接プログラミングに代わる方法として、旋回軸の追加オフセットを、現在動作中のゼロオフセットから自動的に転送することもできます。転送はマシンデータに設定します。

例

```
TRAORI(1,0,0,1)          ; 工具の基本の向きは Z 方向です
TRAORI(1,0,1,0)          ; 工具の基本の向きは Z 方向です
TRAORI(1,0,1,1)          ; 工具の基本の向きは Y/Z 方向です(-45°の位置になります)
```

7.2.3 向きのプログラミングのタイプと初期設定(ORIRESET)

TRAORI による工具オリエンテーションのプログラミング

プログラム指令可能な TRAORI 方向座標変換と組み合わせるには、直線軸 X、Y、Z の他に回転軸識別子 A..、B...、C...も使用して、軸位置、または角度またはベクトル成分で仮想軸をプログラム指令できます。旋回軸と機械軸では、さまざまなタイプの補間を実行できます。現在有効な PO[角度]旋回多項式、および PO[軸]軸多項式に関係なく、異なったタイプの多くの多項式をプログラム指令できます。これには G1、G2、G3、CIP または POLY が含まれます。

工具オリエンテーションの変更は、配向ベクトルでプログラム指令できる場合もあります。このような場合では、各ブロックの最終的な向きを、ベクトルの直接プログラミングまたは回転軸位置のプログラミングによって設定できます。

3 軸～ 5 軸座標変換の向きのプログラミングタイプ

以下に示す向きのプログラムのバージョンは、相互に排他的です。

A、B、C	回転軸位置の直接入力。
A2、B2、C2	オイラー角または RPY 角を使用した仮想軸の角度のプログラム
A3、B3、C3	ベクトル成分項目
LEAD、TILT	軌跡と面に対するリード角と傾斜角の指定
A4、B4、C4 A5、B5、C5	ブロックの始点とブロックの終点にある、面法線ベクトル
A6、B6、C6 A7、B7、C7	テーパ面変換における配向ベクトルの補間
A8、B8、C8	工具の方向転換、後退運動の方向と軌跡長さ

工具オリエンテーションの初期設定のアプローチ(ORIRESET)

ORIRESET(...)を介して、関連する機械のキネマティックスの旋回軸を、現在の位置から、プログラムされた初期状態位置へ、線形的かつ同期的に移動します。軸の基本位置がプログラムされていない場合、関連するマシンデータ
$MC_TRAFO5_ROT_AX_OFFSET_1/2 の位置が使用されます。

回転軸の有効なフレームは無視されます。

機械のキネマティックス CA(チャネル軸名称 C、A)の例

命令	説明
ORIRESET(90, 45)	軸 C:90° 軸 A:45°
ORIRESET(, 30)	軸 C: $MC_TRAFO5_ROT_AX_OFFSET_1/2[0] 軸 A:30°
ORIRESET( )	軸 C: $MC_TRAFO5_ROT_AX_OFFSET_1/2[0] 軸 A: $MC_TRAFO5_ROT_AX_OFFSET_1/2[1]

機械のキネマティックス CAC(チャネル軸名称 C、A、B)の例

命令	説明
ORIRESET(90, 45, 90)	軸 C:90° 軸 A:45° 軸 B:90°
ORIRESET( )	軸 C: $MC_TRAFO5_ROT_AX_OFFSET_1/2[0] 軸 A: $MC_TRAFO5_ROT_AX_OFFSET_1/2[1] 軸 B: $MC_TRAFO5_ROT_AX_OFFSET_1/2[2]

注記

ORIRESET...)による工具オリエンテーションの初期状態への移動は、有効なオリエンテーション座標変換 TRAORI...)とともにのみ実行できます。

LEAD、TILT、および THETA 回転のプログラミング

リード角 LEAD と傾斜角 TILT。

3 軸～ 5 軸座標変換に関しては、工具オリエンテーションの回転を LEAD 角と TILT 角でプログラム指令します。

回転角度 THETA

3 番目の回転軸による座標変換の場合、工具自身の回転を、ベクトル成分のオリエンテーション、および角度 LEAD と TILT のプログラムのために、THETA 回転角度でプログラムできます。

7.2.4 工具オリエンテーションのプログラミング(A...、B...、C...、LEAD、TILT):

工具オリエンテーションのプログラム指令のときは、次のオプションを使用できます。

1. 回転軸の移動の直接プログラミング。基本座標系または機械座標系では、向きの変更が常に発生します。同期軸として旋回軸を移動します。
2. A2, B2, C2 を使用した角度定義に従ったオイラー角または RPY 角のプログラミング。
3. A3, B3, C3 を使用した方向ベクトルのプログラミング。方向ベクトルは、工具先端から工具アダプタの方向を指します。
4. A4, B4, C4 によるブロック始点、および A5, B5, C5 によるブロック終点の面法線ベクトルのプログラミング(正面削り)。
5. リード角 LEAD と傾斜角 TILT を使用したプログラミング。

6. A6, B6, C6 を使用した正規化ベクトルとしての円錐の回転軸、または A7, B7, C7, を使用した円錐面の中間の向きのプログラミング。「円錐面に沿った向きのプログラミング(ORIPLANE、ORICONxx)」を参照してください。
7. A8, B8, C8, を使用した後退移動のときの再旋回、工具の方向と軌跡長のプログラミングについては、
「旋回処理のスムージング(ORIPATHS A8=, B8=, C8=)」を参照してください。

注記

すべての場合において、方向座標変換が動作中の場合にのみ、向きをプログラム指令できます。

長所:これらのプログラムは、あらゆる機械のキネマティックスに適用できます。

G 命令による工具オリエンテーションの定義

注記

工作機械メーカ

マシンデータを使用すると、オイラー角と RPY 角を切り替えることができます。従って、マシンデータを適切に設定した場合は、有効なグループ 50 の G 命令に対応した形でも、対応していない形でも切り替えが可能です。以下の設定オプションを選択できます。

1. 旋回軸と旋回角度を定義する両方マシンデータを G 命令でゼロに設定している場合は、A2, B2, C2 を使用してプログラム指令された角度が **マシンデータにより異なります。** また、向きのプログラミングの角度の定義が、オイラー角または RPY 角として解釈されます。
2. 旋回軸を定義するマシンデータを G 命令で 1 に設定した場合は、動作中のグループ 50 の G 命令によって、切り替えが
異なります。
A2, B2, C2 を使用してプログラムされた角度は、動作中の G 命令 ORIEULER、ORIRPY、ORIVIRT1、ORIVIRT2、ORIAXPOS、および ORIPY2 に従って解釈されます。旋回軸でプログラムした値も、動作中のグループ 50 の G 命令に従って旋回角度として解釈されます。
3. 旋回角度を定義するマシンデータを G 命令で 1 に、旋回軸を定義するマシンデータを G 命令で 0 に設定した場合は、動作中のグループ 50 の G 命令によって、切り替えは**違いません**。
A2, B2, C2 を使用してプログラムした角度は、動作中の G 命令 ORIEULER、ORIRPY、ORIVIRT1、ORIVIRT2、ORIAXPOS、および ORIPY2 のいずれかに従って解釈されます。旋回軸でプログラムした値は、有効なグループ 50 の G 命令にかかわらず、常に回転軸の位置として解釈されます。

構文

回転軸の位置

```
G1 X<値> Y<値> Z<値> A<値> B<値> C<値>
```

オイラー角

```
G1 X<値> Y<値> Z<値> A2<値> B2<値> C2<値>
```

方向ベクトル

```
G1 X<値> Y<値> Z<値> A3<値> B3<値> C3<値>
```

ブロック始点での面法線ベクトル

```
G1 X<値> Y<値> Z<値> A4<値> B4<値> C4<値>
```

ブロック終点での面法線ベクトル

```
G1 X<値> Y<値> Z<値> A5<値> B5<値> C5<値>
```

リード角

```
LEAD=<値>
```

傾斜角

```
TILT=<値>
```

意味

G1:	直線補間
X, Y, Z:	直線軸の位置
A、B、C:	回転軸の位置
A2=、B2=、C2=:	角度のプログラム(オイラー角または RPY 角)
A3=、B3=、C3=:	WCS による X、Y、Z 座標の方向ベクトル。
A4=、B4=、C4=:	WCS による X、Y、Z 座標のブロック始点の面法線ベクトル。
A5=、B5=、C5=:	WCS による X、Y、Z 座標のブロック終点の面法線ベクトル。
LEAD= :	リード角 1)
TILT= :	傾斜角 1)
1) 角度の指定の解釈は、MD21094 $MC_ORIPATH_MODE での設定によって異なります。	

詳細情報

5 軸プログラムは通常、CAD/CAM システムで生成され、制御装置では入力されません。そのため、次の説明は主に、ポストプロセッサのプログラマを対象としています。

向きのプログラムには、以下のコマンドを使用できます。

命令	意味
ORIEULER:	回転処理 ZX'Z"のオイラー角
ORIRPY:	回転処理 XY'Z"の RPY 角
ORIRPY2:	回転処理 ZY'X"の RPY 角
ORIVIRT1:	以下を通じて自由に定義可能な、回転処理を伴う仮想旋回軸: MD21120 $MC_ORIAX_TURN_TAB_1
ORIVIRT2:	以下を通じて自由に定義可能な、回転処理を伴う仮想旋回軸: MD21130 $MC_ORIAX_TURN_TAB_2
ORIAXPOS:	回転軸位置を伴う仮想旋回軸

注記

工作機械メーカはマシンデータを使用して、さまざまなタイプを定義できます。工作機械メーカの取扱説明書を参照してください。

オイラー角のプログラム ORIEULER、回転処理 Z X' Z"

A2、B2、C2 で向きプログラム ORIEULER を行うときのプログラム指令値は、オイラー角(° 単位)として解釈されます。

新しい向きのベクトルは、元の向きのベクトルにおける以下 3 つの回転の結果として生じます。

1. 座標軸 Z を中心とする回転軸 A2
2. 新しい座標軸 X'を中心とする回転軸 B2
3. 座標軸 Z"を中心とする回転軸 C2

この場合、C2 の値(新しい Z 軸を中心とする回転)は無意味であり、プログラム指令する必要はありません。

RPY 角のプログラム ORIRPY、回転処理 X Y' Z"

A2、B2、C2 で向きのプログラム RPY を行うときのプログラム指令値は、RPY 角(° 単位)として解釈されます。

注記

ORIEULER を使用したプログラムとは対照的に、ORIRPY では 3 つの値がすべて配向ベクトルに作用します。

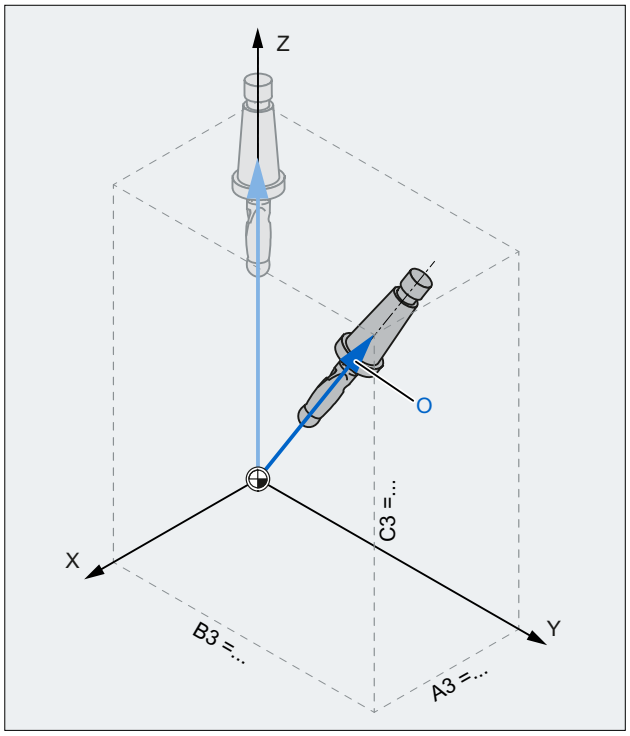
新しい向きのベクトルは、元の向きのベクトルにおける以下 3 つの回転の結果として生じます。

1. 座標軸 X を中心とする回転軸 A2
2. 新しい座標軸 Y'を中心とする回転軸 B2
3. 座標軸 Z"を中心とする回転軸 C2

方向ベクトルのプログラミング

方向ベクトル成分は、A3, B3, C3 でプログラム指令します。ベクトルは工具アダプタの方向を指し、ベクトル長には意味はありません。

プログラム指令していないベクトル成分は、ゼロとに設定されます。



X、Y、Z WCS の座標軸

A3、B3、C3 方向ベクトルのコンポーネント

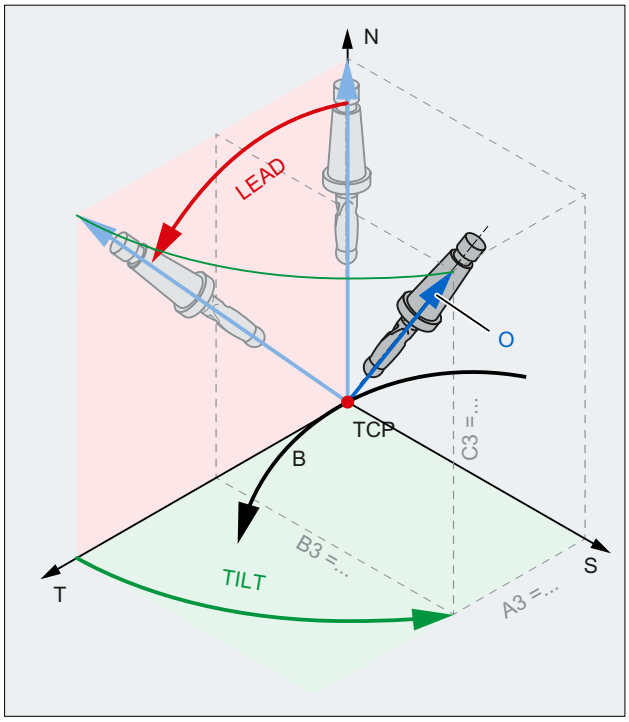
O 配向ベクトル

図 7-1 方向ベクトルのプログラム

LEAD と TILT による工具オリエンテーションのプログラム

得られる工具オリエンテーションは以下により特定されます。

- 軌跡タンジェント
- 面法線ベクトル
  A4, B4, C4 のブロック始点、および A5, B5, C5 のブロック終点
- リード角 LEAD
  軌跡タンジェントと面法線ベクトルで定義した平面での角度
- ブロック終点での傾き角 TILT
  面法線ベクトルに対する軌跡タンジェントに垂直な平面上の角度



T　軌跡タンジェント

S　軌跡タンジェントと直交

N　面法線

B　パス

TCP　工具中心点

O　配向ベクトル

図 7-2　LEAD TILT のプログラム

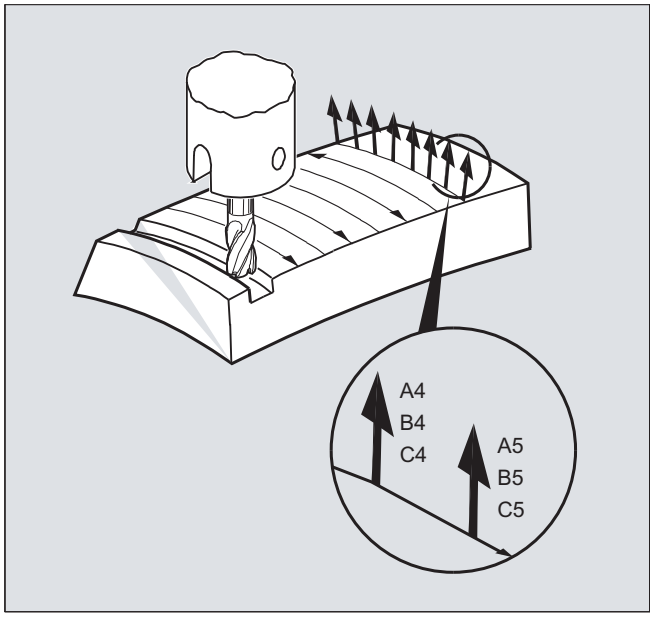
注記

3 次元工具オフセットでの内側コーナの動作

ブロックが内側コーナで短縮されると、プログラムされた工具オリエンテーションもブロック終点で調整されることになります。

7.2.5 正面削り(A4、B4、C4、A5、B5、C5)

正面削りを使用すると、あらゆる種類の曲面を加工できます。



このタイプの 3 次元加工の場合は、ワーク表面の 3 次元軌跡を行単位で記述してください。

通常は CAM で実行される計算では、工具の形状と寸法が考慮されます。その後、計算を完了した NC ブロックが、ポストプロセッサ経由で制御装置に読み取られます。

軌跡曲率のプログラミング

面の記述

軌跡曲率は、以下の成分を使用した面法線ベクトルで記述します。

`A4, B4, C4` ブロック始点の開始ベクトル

`A5, B5, C5` ブロック終点の終了ベクトル

ブロックに開始ベクトルのみが含まれる場合、面法線ベクトルはブロック全体を通じて一定です。ブロックに終了ベクトルのみが含まれる場合は、前のブロックの終了値から、大円弧補間で、プログラム指令の終了値へと補間が実行されます。

開始ベクトルと終了ベクトルをプログラム指令した場合は、大円弧補間で、2 方向の間で補間が実行されます。これにより、連続的に滑らかな軌跡を作成できます。

有効な G17 ～ G19 平面にかかわらず、初期設定では、面法線ベクトルは Z 方向を指します。

ベクトルの長さには意味はありません。

プログラム指令していないベクトル成分は、ゼロに設定されます。

ORIWKS が有効な場合(「旋回軸の基準(ORIWKS、ORIMKS): (ページ 401)」を参照してください)、面法線ベクトルは動作中のフレームを基準とし、フレーム回転でも回転されます。

機械製造メーカ

面法線ベクトルは、マシンデータで設定した制限値内で軌跡タンジェントに直交している必要があります。直交していない場合は、アラームが発生します。

7.2.6 旋回軸の基準(ORIWKS、ORIMKS):

ワーク座標系で向きをプログラム指令するには、以下のいずれかを使用できます。

- オイラー角または RPY 角
- 配向ベクトル

上記を使用した場合は、ORIMKS/ORIWKS を使用して、回転移動の進路を設定できます。

注記

工作機械メーカ

旋回補間のタイプは、次のマシンデータで設定します。

MD21104 $MC_ORI_IPO_WITH_G_CODE

= FALSE:G 命令 ORIWKS と ORIMKS で基準が設けられます。

= TRUE:51 番目のグループの G 命令(ORIAXES、ORIVECT、ORIPLANE など)で基準が設定されます。

構文

```
ORIMKS=...
ORIWKS=...
```

意味

ORIMKS:	機械座標系の回転
ORIWKS:	ワーク座標系の回転

注記

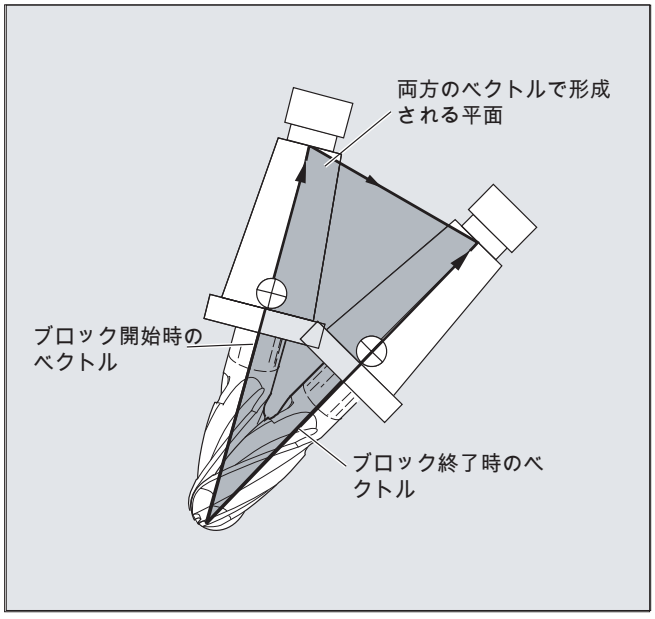
基本設定は ORIWKS です。どの機械で実行するかが明確でない 5 軸プログラムの場合は、常に ORIWKS を選択してください。機械が実際にどのように移動を実行するかは、機械のキネマティックスにより異なります。

ORIMKS を使用すると、(機器類との衝突を避けるためなどに)実際の機械の移動をプログラム指令できます。

詳細情報

ORIMKS を使用した場合、工具がおこなう移動は機械のキネマティックスにより**異なります**。空間の固定点で工具先端の向きが変わった場合は、各回転軸位置の間で直線補間がおこなわれます。

ORIWKS を使用した場合、工具がおこなう移動は機械のキネマティックスには**影響されません**。固定された工具先端の向きが変わると、開始ベクトルと終了ベクトルで設定された平面で工具が移動します。



特異点

注記

ORIWKS

5 軸機械が特異点設定領域で旋回移動をおこなうには、機械軸が大きく移動する必要があります。(たとえば、C を回転軸、A を旋回軸として回転するスイベルヘッドでは、A = 0 のすべての位置は特異点です。)

機械製造メーカ

機械軸への過負荷を避けるために、特異点位置近傍の工具軌跡速度は速度制御により大幅に減速されます。

使用するマシンデータ

$MC_TRAFO5_NON_POLE_LIMIT

$MC_TRAFO5_POLE_LIMIT

上記のマシンデータで座標変換のパラメータを設定すると、極近傍の旋回移動で極を通過するため、高速加工をおこなうことができます。

特異点は MD $MC_TRAFO5_POLE_LIMIT でのみ処理できます。

参照先:
/FB3/機能マニュアル 応用機能; 3 ～ 5 軸座標変換(F2)、
「特異点とその処理方法」の章

7.2.7 旋回軸のプログラミング(ORIAXES、ORIVECT、ORIEULER、ORIRPY、ORIRPY2、ORIVIRT1、ORIVIRT2)

空間の工具の向きを記述する「旋回軸」機能は、回転軸のオフセットをプログラム指令して実行します。工具自身の回転もおこなうと、さらに 3 次元の自由度が提供されます。この場合は、3 番目の回転軸により空間で工具の向きが決まります。これには 6 軸座標変換が必要です。工具自身の回転は、回転ベクトルの補間タイプに従って、THETA の回転角度を使用して定義します(「工具オリエンテーションの回転(ORIROTA、ORIROTR、ORIROTT、ORIROTC、THETA) (ページ 414)」を参照してください)。

軸識別子 A2、B2、および C2 を使用して旋回軸をプログラム指令します。

構文

```
N...ORIAXES/ORIVECT                ; 直線補間または大円弧補間
N...G1 X Y Z  A B C

N...ORIPLANE                       ;平面の旋回補間

N...ORIEULER/ORIRPY/ORIRPY2        :旋回角度 オイラー角/RPY 角
N...G1 X Y Z A2= B2= C2=           ;仮想軸の角度のプログラミング

N...ORIVIRT1/ORIVIRT2              ; 仮想旋回軸の定義 1/2
N...G1 X Y Z A3= B3= C3=           ; 方向ベクトルのプログラミング
```

注記

旋回軸の他の回転軸オフセットをプログラム指令して、空間の円錐面に沿って向きを変更できます。「円錐面に沿った向きのプログラミング(ORIPLANE、ORICONCW、ORICONCCW、ORICONTO、ORICONIO) (ページ 406)」を参照してください。

意味

ORIAXES:	機械軸または旋回軸の直線補間
ORIVECT:	大円弧補間(ORIPLANE と同じ)
ORIMKS: ORIWKS:	機械座標系の回転 ワーク座標系の回転 詳細については、「旋回軸の基準(ORIWKS、ORIMKS): (ページ 401)」を参照してください。
A= B= C=:	機械軸位置のプログラミング
ORIEULER:	オイラー角による向きのプログラミング
ORIRPY:	RPY 角による向きのプログラミング 回転の順序は XYZ で、 • A2 は X を中心とする回転の角度 • B2 は Y を中心とする回転の角度 • C2 は Z を中心とする回転の角度

ORIRPY2:	RPY 角による向きのプログラミング 回転の順序は ZYX で、 • A2 は Z を中心とする回転の角度 • B2 は Y を中心とする回転の角度 • C2 は X を中心とする回転の角度
A2= B2= C2=:	仮想軸の角度のプログラミング
ORIVIRT1/ORIVIRT2:	仮想旋回軸を使用した向きのプログラミング 定義 1: MD21120 $MC_ORIAX_TURN_TAB_1 による定義 定義 2: MD21130 $MC_ORIAX_TURN_TAB_2 による定義
A3= B3= C3=:	方向軸の方向ベクトルのプログラミング

詳細情報

機械製造メーカ

MD21102 $MC_ORI_DEF_WITH_G_CODE は、プログラム指令角度 A2、B2、C2 の定義方法を指定します。

定義は、MD21100 $MC_ORIENTATION_IS_EULER (標準)、または G グループ 50 (ORIEULER、ORIRPY、ORIVIRT1、ORIVIRT2)に従います。

MD21104 $MC_ORI_IPO_WITH_G_CODE は、有効な補間モードタイプを定義します。ORIWKS/ORIMKS と ORIAXES/ORIVECT のどちらにするかを定義します。

JOG モード

この動作モードの旋回角度の補間は常に直線補間です。移動キーによる連続移動とステップ移動時は、1 つの旋回軸のみ移動できます。複数の旋回軸を、手動パルス発生器を使用して同時に移動できます。

旋回軸を手動で移動する場合は、チャネル別の送り速度オーバライドスイッチ、または早送りオーバライドスイッチが早送りオーバライドで機能します。

次のマシンデータを使用すると、個別の速度を設定できます。

MD21160 $MC_JOG_VELO_RAPID_GEO

MD21165 $MC_JOG_VELO_GEO

MD21150 $MC_JOG_VELO_RAPID_ORI

MD21155 $MC_JOG_VELO_ORI

注記

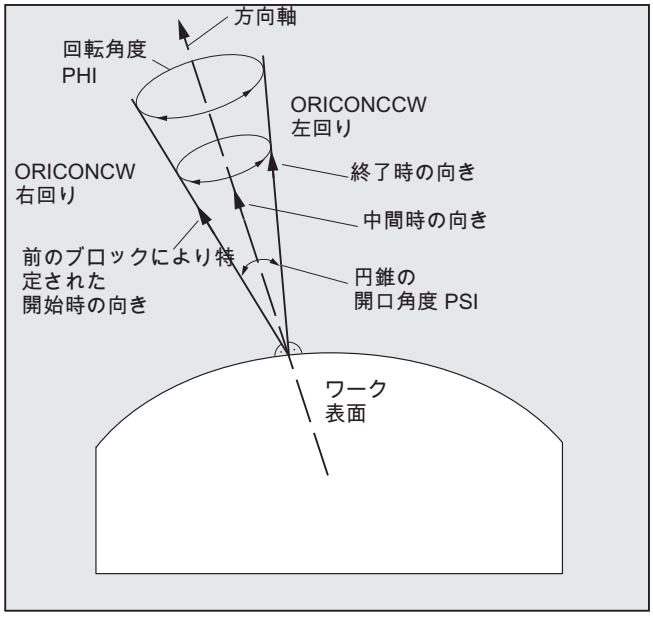
「マテハン装置用座標変換」を備えた SINUMERIK 840D sl

「手動の直交移動」機能を使用して、JOG モードでの複数のジオメトリ軸の平行移動を、基準系 MCS、WCS、および TCS で、それぞれに個別に設定できます。

参照先:
機能マニュアル 上級機能; キネマティックトランスフォーメーション(M1)

7.2.8 円錐面に沿った向きのプログラミング(ORIPLANE、ORICONCW、ORICONCCW、ORICONTO、ORICONIO)

拡張旋回をおこなうと、空間の円錐面に沿って向きの変更を実行できます。ORICONxx モーダル命令を使用して、配向ベクトルを円錐面で補間します。旋回の終了は、平面上の補間のための ORIPLANE でプログラム指令できます。旋回の開始は通常、前のブロックで定義します。



プログラミング

旋回の終了は、A2, B2, C2 によるオイラー角または RPY 角での角度のプログラミングの指定、または A, B, C による回転軸位置のプログラミングにより定義します。円錐面に沿った旋回軸には、さらに以下の詳細なプログラミングが必要です。

- A6、B6、C6 による円錐の回転軸のベクトル
- 識別子 NUT による開口角度 PSI
- A7, B7, C7 による円錐面の中間旋回

注記

円錐の回転軸のための方向ベクトル A6, B6, C6 のプログラミング

旋回の終了は必ずプログラミングする必要はありません。旋回の終了を指定していない場合は、360°の円錐面全体が補間されます。

NUT=角度による、円錐の開口角度のプログラミング

旋回の終了を指定してください。

この方法では、360°の円錐面全体は補間できません。

円錐面の中間旋回 A7, B7, C7 のプログラミング

旋回の終了を指定してください。向きの変更と回転方向を、旋回の開始、旋回の終了、および旋回の中間という 3 つのベクトルで一義的に定義します。この 3 つはそれぞれ、異なるベクトルです。プログラム指令した旋回の中間が、旋回の開始または終了に平行である場合は、直線の旋回大円弧補間が、開始ベクトルと終了ベクトルで定義された平面で実行されます。

円錐面での拡張旋回補間

`N...ORICONCW` または `ORICONCCW`	円錐面の**補間**
`N...A6= B6= C6= A3= B3= C3=`	円錐の右回り/左回り方向の方向ベクトルと旋回の終了、または
または	
`N...ORICONTO`	接線方向の遷移と
`N...G1 X Y Z A6= B6= C6=`	旋回の終了の指定、
または	
`N...ORICONIO`	または
`N...G1 X Y Z A7= B7= C7=`	旋回の終了と
	円錐面の中間旋回の指定で
`N...PO[PHI]=(a2, a3, a4, a5)`	回転角度の多項式と
`N...PO[PSI]=(b2, b3, b4, b5)`	開口角度の多項式を使用

パラメータ

ORIPLANE:	平面での補間(大円弧補間)
ORICONCW:	円錐面での右回り方向の補間
ORICONCCW:	円錐面での左回り方向への補間
ORICONTO:	接線方向の遷移による円錐面の補間
A6= B6= C6=:	円錐の回転軸のプログラミング(正規化ベクトル)
NUT=角度:	円錐の開口角度(° 単位)
NUT=+179:	180°以下の移動角度
NUT=-181:	180°以上の移動角度
ORICONIO:	円錐面での補間
A7= B7= C7=:	中間旋回(正規化ベクトルとしてのプログラミング)
PHI:	円錐の方向軸を中心とした向きの回転角度
PSI:	円錐の開口角度
使用可能な多項式 PO[PHI]=(a2, a3, a4, a5) PO[PSI]=(b2, b3, b4, b5)	それぞれの角度とは別に、 5 次までの多項式もプログラム指令できます

例:さまざまな向きの変更

プログラムコード	コメント
…	
N10 G1 X0 Y0 F5000	
N20 TRAORI(1)	;方向座標変換をオンします
N30 ORIVECT	; 工具オリエンテーションをベクトルとして補間します
…	; 平面の工具オリエンテーション
N40 ORIPLANE	; 大円弧補間を選択します
N50 A3=0 B3=0 C3=1	
N60 A3=0 B3=1 C3=1	; Y/Z 平面での向きは 45°回転し、ブロック終点で目的の向き(0、1/√2、1/√2)に到達します。
…	
N70 ORICONCW	; 円錐面の向きのプログラミングです。
N80 A6=0 B6=0 C6=1 A3=0 B3=0 C3=1	配向ベクトルは、方向(0,0,1)の円錐面で目標の向き(1/√2,0,1/√2)まで右回りに補間されます。回転角度は 270°です。
N90 A6=0 B6=0 C6=1	; 工具オリエンテーションは、同じ円錐面で完全に 1 回転します。

詳細情報

空間の任意の位置での円錐面に沿った向きの変更を記述する場合は、工具オリエンテーションの回転の中心となるベクトルが必要です。旋回の開始と終了も指定してください。前のブロックから旋回の開始を得て、旋回の終了はプログラム指令、または他の条件を使用して定義してください。

ORIVECT に対応する ORIPLANE 平面のプログラミング

角度多項式による大半径円弧補間のプログラミングは、輪郭の直線補間と多項式補間に対応します。工具オリエンテーションは、旋回の開始と終了で定義した平面で補間されます。さらに、多項式をプログラム指令する場合は、配向ベクトルを平面から傾けることもできます。

平面の円弧のプログラミング G2/G3、CIP、および CT

拡張旋回は、平面上の円弧の補間に対応します。G2/G3 などの中心または半径のある円弧、中間点 CIP をによる円弧、および接線方向の円弧 CT の、対応するプログラミングオプションについて詳しくは、以下を参照してください。

参照先:プログラミング説明書 基本編;「移動指令のプログラミング」

旋回のプログラミング

円錐面上の配向ベクトルの補間 ORICONxx

円錐面上の向きを補間するために、G グループ 51 から、以下の 4 つの補間タイプを選択できます。

1. 旋回の終了と円錐の方向、または開口角度の指定による円錐面の右回り方向の補間 ORICONCW。方向ベクトルは、識別子 A6, B6, C6、および識別子 NUT= 0 ～ 180°の範囲の円錐の開口角度でプログラム指令します。
2. 旋回の終了と円錐の方向、または開口角度の指定による円錐面の左回り方向の補間 ORICONCCW。方向ベクトルは、識別子 A6, B6, C6、および識別子 NUT= 0 ～ 180°の範囲の円錐の開口角度でプログラム指令します。
3. 旋回の終了、および中間旋回の指定による円錐面の補間 ORICONIO。中間旋回は識別子 A7, B7, C7 でプログラム指令します。
4. 接線方向の遷移、および旋回の終了の指定による円錐面の補間 ORICONTO。方向ベクトルは、識別子 A6, B6, C6 でプログラム指令します。

7.2.9 2 つの接点の向きの指定(ORICURVE、PO[XH]=、PO[YH]=、PO[ZH]=)

空間の、もう 1 つの曲線を使用した向き変更 ORICURVE のプログラミング

空間で、曲線に沿って工具先端を移動する以外に、向きの変更をプログラム指令するもう 1 つの方法は、ORICURVE を使って、工具の 2 番目の接点の移動をプログラム指令します。この方法を使うと、工具ベクトル自身をプログラム指令したときと同様に、工具オリエンテーションの変更を一義的に定義できます。

機械製造メーカ

軸識別子に関する工作機械メーカの注釈を参照してください。マシンデータで軸識別子を設定すると、工具の旋回軌跡をもう 1 つプログラム指令できます。

プログラミング

このタイプの補間を使用すると、空間の 2 つの曲線を表わす点(G1 によって)または多項式(POLY を使用)をプログラム指令できます。円弧とインボリュート曲線はプログラム指令できません。BSPLINE スプライン補間と「短いスプラインブロックの結合」機能も有効です。

参照先:
機能マニュアル 基本機能; 連続軌跡モード、イグザクトストップと先読み(B1)、「短いスプラインブロックの結合」の章

別のスプラインタイプ、ASPLINE と CSPLINE、および COMPON、COMPCURV、または COMPCAD を使用したコンプレッサの適用はできません。

座標の旋回多項式をプログラミングするときに、工具の 2 つの接点の移動を 5 次まで事前設定できます。

空間の追加曲線による拡張旋回補間と座標の多項式

```
N...ORICURVE
N...PO[XH]=(xe, x2, x3, x4, x5)
N...PO[YH]=(ye, y2, y3, y4, y5)
N...PO[ZH]=(ze, z2, z3, z4, z5)
```

工具の 2 番目の接点の移動の指定と当該座標の追加多項式

パラメータ

ORICURVE	工具の 2 つの接点間の移動を指定した旋回補間。
XH YH ZH	空間の曲線としての工具追加輪郭の 2 番目の接点の座標識別子

使用可能な多項式 `PO[XH]=(xe, x2, x3, x4, x5) PO[YH]=(ye, y2, y3, y4, y5) PO[ZH]=(ze, z2, z3, z4, z5)`	当該のそれぞれの終点を使用する以外に、空間の曲線を多項式でプログラム指令することもできます。
`xe, ye, ze`	空間の曲線の終点
`xi, yi, zi`	5 次までの多項式の係数

注記

2 番目の旋回軌跡をプログラムするための識別子 XH YH ZH

識別子は、他の識別子や直線軸と競合しないように選択してください。

X Y Z の各軸

、および以下のような回転軸

A2 B2 C2 オイラー角または RPY 角

A3 B3 C3 方向ベクトル

A4 B4 C4 または A5 B5 C5 面法線ベクトル

A6 B6 C6 回転ベクトルまたは A7 B7 C7 中間点座標

またはその他の補間パラメータ

7.3 旋回多項式(PO[角度]、PO[座標])

G グループ 1 による、現在有効な多項式補間にかかわらず、2 つのタイプの 5 次までの旋回多項式をプログラムして、3 軸から 5 軸の座標変換をおこなうことができます。

1. **角度**の多項式:
   旋回の開始と終了で定義した平面に対するリード角 LEAD と傾斜角 TILT です。
2. **座標**の多項式:工具の基準点の工具オリエンテーションのための空間の 2 番目の曲線の XH、YH、ZH です。

6 軸座標変換を使用すると、回転ベクトル THT の回転を 5 次までの多項式でプログラム指令して、工具オリエンテーションだけでなく、工具自身の回転も実行できます。

構文

タイプ 1 **角度**の旋回多項式

`N… PO[PHI]=(a2, a3, a4, a5)`	3 軸～ 5 軸座標変換
`N… PO[PSI]=(b2, b3, b4, b5)`	

タイプ 2 **座標**の旋回多項式

`N… PO[XH]=(xe, x2, x3, x4, x5)`	工具オリエンテーションの 2 番目の旋回軌跡の座標識別子
`N… PO[YH]=(ye, y2, y3, y4, y5)`	
`N… PO[ZH]=(ze, z2, z3, z4, z5)`	

上記のいずれの場合も、6 軸座標変換を実行すると、以下のような多項式をプログラム指令して、**回転**実行することもできます。

`N… PO[THT]=(c2, c3, c4, c5)`	軌跡に対する回転の補間
または	
`N… PO[THT]=(d2, d3, d4, d5)`	配向ベクトルの向きの変更に対するアブソリュート補間、相対補間、および接線方向の補間

of the orientation vector.これが可能なのは、THETA の回転角度によるプログラム指令と補間が可能なオフセットを含む回転ベクトルを座標変換がサポートしている場合のみです。

意味

PO[PHI]	旋回の開始と終了の間にある平面の角度
PO[PSI]	旋回の開始と終了の間の平面からの、向きの傾斜角
PO[THT]	THETA でプログラムしたグループ 54 の G 命令のうちいずれかの回転ベクトルの回転により生じた回転角度
PHI	リード角 LEAD
psi	傾斜角 TILT
THETA	Z の工具方向を中心とする回転
PO[XH]	工具の基準点の X 座標
PO[YH]	工具の基準点の Y 座標
PO[ZH]	工具の基準点の Z 座標

詳細情報

次の場合は、旋回多項式をプログラム指令できません。

- ASPLINE、BSPLINE、CSPLINE スプライン補間が有効である。
  タイプ 1(角度の旋回多項式)は、スプライン補間を除く、すべてのタイプの補間に使用できます。つまり、早送り G00 または送り速度 G01 による直線補間、
  POLY を使用した多項式補間、および
  円弧/インボリュート補間 G02、G03、CIP、CT、INVCW、および INCCCW に使用できます
  。
  ただし、タイプ 2(座標の旋回多項式)は、
  早送り G00 または送り速度 G01 による直線補間、または
  POLY を使用した多項式補間が有効な場合にかぎり、使用できます。
- ORIAXES の軸補間を使用して向きを補間する。この場合は、旋回軸 A と B に対して多項式の PO[A]と PO[B]を使用して直接プログラム指令できます。

ORIVECT、ORIPLANE、および ORICONxx を使用したタイプ 1 の旋回多項式

タイプ 1 の旋回多項式は、ORIVECT、ORIPLANE、および ORICONxx を使用して大半径円弧補間、および円錐面の補間に使用できます。

ORICURVE を使用したタイプ 2 の旋回多項式

空間の追加曲線による補間の ORICURVE が有効な場合は、配向ベクトルの直交成分が補間され、使用できるのはタイプ 2 の旋回多項式のみとなります。

7.4 工具オリエンテーションの回転(ORIROTA、ORIROTR、ORIROTT、ORIROTC、THETA)

可動式工具を備えた機械タイプの工具の向きも変更する場合は、各ブロックに旋回の終了をプログラム指令してください。機械のキネマティックスに応じて、旋回軸の旋回方向または配向ベクトル THETA の回転方向をプログラム指令できます。以下の回転ベクトルに対して、さまざまな補間タイプをプログラム指令できます。

- ORIROTA:アブソリュート回転方向への回転角度です。
- ORIROTR:旋回の開始と終了の間で形成される平面に対する回転角度です。
- ORIROTT:配向ベクトルの変更に対する回転角度です。
- ORIROTC:軌跡タンジェントに対する接線方向の回転角度です。

構文

補間タイプ ORIROTA が有効な場合にのみ、回転角度または回転ベクトルを、4 つのモードすべてで次のようにプログラム指令できます。

1. 直接、回転軸位置 A, B, C として
2. A2, B2, C2 によるオイラー角(° 単位)
3. A2, B2, C2 による RPY 角(° 単位)
4. A3, B3, C3 による方向ベクトル(THETA=<値>を使用した回転角度)

ORIROTR または ORIROTT が有効な場合は、回転角度を THETA でのみ、直接プログラム指令できます。

向きを変更せずに、別のブロックで回転をプログラム指令することもできます。この場合、ORIROTR と ORIROTT は無効です。この場合は、回転角度は常に、アブソリュート方向(ORIROTA)を基準として解釈されます。

N...ORIROTA	回転ベクトルの補間を定義します
N...ORIROTR	
N...ORIROTT	
N...ORIROTC	
N...A3= B3= C3= THETA=<値>	配向ベクトルの回転を定義します
N...PO[THT]=(d_2, d_3, d_4, d_5)	回転角度を 5 次多項式で補間します

意味

ORIROTA:	アブソリュート回転方向への回転角度
ORIROTR:	旋回の開始と終了間の平面に対する回転角度
ORIROTT:	向きの変更に対する接線方向の回転ベクトルの回転角度です。
ORIROTC:	軌跡タンジェントに対する接線方向の回転ベクトルの回転角度です。
THETA:	配向ベクトルの回転
THETA=<値>:	ブロック終点へ到達するまでの回転角度(° 単位)
THETA=Θe:	回転ベクトルの終了角度が Θ_e の回転角度
THETA=AC(…):	アブソリュート指令へのノンモーダル切り替え
THETA=AC(…):	インクレメンタル指令へのノンモーダル切り替え
Θe:	G90 によるアブソリュート回転ベクトル、かつ G91 (インクレメンタル指令)による相対回転ベクトルの両方の終了角度が有効です
PO[THT]=(....):	回転角度の多項式

例:向きの回転

プログラムコード	コメント
N10 TRAORI	; 方向座標変換を起動します。
N20 G1 X0 Y0 Z0 F5000	; Z 軸の回転
N30 A3=0 B3=0 C3=1 THETA=0	; Z 方向に回転角度 0
N40 A3=1 B3=0 C3=0 THETA=90	; X 方向に約 90°回転
N50 A3=0 B3=1 C3=0 PO[THT]=(180,90)	; 向き
N60 A3=0 B3=1 C3=0 THETA=IC(-90)	; Y 方向に約 180°回転
N70 ORIROTT	; そのまま一定で、90°回転
N80 A3=1 B3=0 C3=0 THETA=30	; 向きの変更に対する回転角度
	; X/Y 平面に対して 30°の角度の回転ベクトル

ブロック N40 の補間時には、0°(初期値)から 90°(最終値)までの回転角度を直線的に補間します。ブロック N50 では、回転角度が放物線 θ(u) = +90u2 に従って、90°から 180°まで変化します。N60 では、向きを変更せずに、回転を実行することもできます。

N80 を使用すると、工具オリエンテーションは Y 方向から X 方向に向かって回転します。向きの変更は X/Y 平面でおこなわれ、回転ベクトルはこの平面に対して 30°の角度を表わします。

詳細情報

ORIROTA

回転角度 THETA は、空間のアブソリュート方向を基準として補間されます。回転の基本方向はマシンデータで定義します。

ORIROTR

回転角度 THETA は、旋回の開始と終了で定義した平面に対して補間されます。

ORIROTT

回転角度 THETA は、向きの変更に対して補間されます。THETA=0 の場合は、向きの変更に対して接線方向に回転ベクトルが補間されます。また、向きの「傾斜角 PSI」に対して 1 つ以上の多項式をプログラム指令している場合を除き、ORIROTR と同じです。この場合の結果は、向きは変更されますが、平面では実行されません。その後、さらに回転角度 THETA を使用して、回転ベクトルを補間すると、常に向きの変更を示す特定の値を作成します。

ORIROTC

回転ベクトルは、オフセットを使用して軌跡タンジェントに対して補間されます。このオフセットは THETA 角度でプログラム指令可能です。5 次以下の多項式 PO[THT]=(c2, c3, c4, c5)をオフセット角に対してプログラム指令することもできます。

7.5 軌跡に対する向き

7.5.1 軌跡に対する向きのタイプ

この拡張機能を使用すると、相対的な向きがブロック終点だけでなく、軌道全体にわたって成立します。前のブロックで成立した向きを、大円弧補間によって、プログラム指令した旋回の終了へ渡します。軌跡に対して求める向きをプログラミングする方法は、基本的に次の 2 つです。

1. 工具回転と同様に、工具オリエンテーションは、ORIPATH、ORPATHTS を使用して軌跡に対して補間されます。
2. 配向ベクトルは、通常の方法でプログラム指令し、補間します。配向ベクトルの回転を、ORIROTC を使用して軌跡タンジェントに対して開始します。

構文

旋回補間のタイプと工具の回転は、以下を使用してプログラム指令します。

N...ORIPATH	軌跡に対する向き
N...ORIPATHS	旋回処理のスムージングによる、軌跡に対する向き
N...ORIROTC	軌跡に対する回転ベクトルの補間

軌道のコーナで発生する向きの不連続変化は、ORIPATHS でスムージングできます。後退移動の方向と軌跡長は、A8=X, B8=Y C8=Z 成分を使用してベクトルを使用してプログラム指令します。

ORIPATH/ORIPATHS を使用すると、軌道全体に対して、次の 3 種類の角度を使用して軌跡タンジェントに対してさまざまな基準をプログラム指令できます。

- LEAD= 軌跡と面に対するリード角を指定します
- TILT= 軌跡と面に対する傾斜角を指定します
- THETA=起動全体に対する

回転角度ですまた THETA 回転角度の他に、PO[THT]=(...)で 5 次以下の多項式をプログラム指令できます。

注記

工作機械メーカ

工作機械メーカの取扱説明書を参照してください。軌跡に対する向きに、その他の設定をおこなうには、マシンデータとセッティングデータを使用して設定できます。詳細については、以下を参照してください。

参照先:
/FB3/機能マニュアル 応用機能; 3 軸～ 5 軸の座標変換(F2)、
「オリエンテーション」の章

意味

角度 LEAD 角と TILT 角の補間には、マシンデータを使用して、以下のさまざまな設定が可能です。

- LEAD と TILT でプログラム指令した工具オリエンテーションの基準は、ブロック全体にわたって保持されます。
- リード角 LEAD:接線と法線ベクトルに垂直な方向を中心とする回転 TILT:法線ベクトルを中心とする向きの回転
- リード角 LEAD:接線と法線ベクトルに垂直な方向を中心とする回転 傾斜角 TILT:軌跡タンジェント方向の向きの回転。
- 回転角度 THETA:6 軸座標変換で旋回軸として動作する、3 番目の付加回転軸による工具自身の回転

注記

軌跡に対する向きの OSC、OSS、OSSE、OSD、および OST との組み合わせの禁止

軌跡に対する旋回補間、つまり ORIPATH または ORIPATHS、および ORIOTC は、グループ 34G 命令による旋回処理のスムージングと組み合わせてプログラムすることはできません。このため、OSOF を有効にしてください。

7.5.2 軌跡に対する工具オリエンテーションの回転(ORIPATH、ORIPATHS、回転角度)

6 軸座標変換を使用すると、3 番目の回転軸で工具自身を回転させ、空間内の求める向きに工具を配置できます。ORIPATH または ORIPATHS を使用して、軌跡に対して工具オリエンテーションを回転させると、THETA を使用した回転角度で、さらに回転をプログラム指令できます。または、工具方向と垂直な平面にあるベクトルを使って、LEAD 角と TILT 角をプログラム指令できます。

機械製造メーカ

工作機械メーカの説明書を参照してください。LEAD 角と TILT 角の補間は、マシンデータを使用して、別々の設定にすることができます。

構文

工具オリエンテーションと工具の回転

軌跡に対する工具オリエンテーションのタイプは、ORIPATH または ORIPATHS で有効にします。

`N...ORIPATH`	軌跡に対する旋回のタイプを有効にします
`N...ORIPATHS`	旋回処理のスムージングにより、軌跡に対する旋回のタイプを有効にします

下記の 3 つの回転角度を有効にします:

`N...LEAD=`	面法線ベクトルに対する、プログラム指令の向きの角度
`N...TILT=`	面法線ベクトルに対する軌跡タンジェントに直交する平面上の、プログラム指令の向きの角度
`N...THETA=`	3 番目の回転軸の工具方向の向きの変更に対する回転角度

ブロック終点での角度の値は、`LEAD=`値、`TILT=`値、または `THETA=`値でプログラム指令します。定数の角度にの他に、3 つの角度のすべてに対して 5 次までの多項式をプログラム指令できます。

`N...PO[PHI]=(a2, a3, a4, a5)`	リード角 LEAD の多項式
`N...PO[PSI]=(b2, b3, b4, b5)`	傾斜角 TILT の多項式
`N...PO[THT]=(d2, d3, d4, d5)`	回転角度 THETA の多項式

より高次の多項式係数がゼロのときは、省略してプログラミングできます。
例:PO[PHI]=a2 の結果は、LEAD 角の放物線となります。

意味

軌跡に対する工具オリエンテーション

ORIPATH:	軌跡に対する工具オリエンテーション
ORIPATHS:	軌跡に対する工具オリエンテーション、旋回処理の不連続変化をスムージングします
LEAD:	軌跡タンジェントと面法線ベクトルで定義された平面の、面法線ベクトルに対する角度
TILT:	Z 方向の向きの回転、または軌跡タンジェントを中心とした回転
THETA:	Z へ向かう工具方向を中心とする回転
PO[PHI]:	LEAD 角の旋回多項式
PO[PSI]:	TILT 角の旋回多項式
PO[THT]:	THETA の回転角度の旋回多項式

注記

回転角度 THETA

旋回軸が動作する 3 番目の回転軸で、工具自身を回転させるには、6 軸座標変換が必要です。

7.5.3 軌跡に対する工具回転の補間(ORIROTC、THETA)

回転ベクトルによる補間

ORIROTC でプログラム指令した、軌跡タンジェントに対する工具回転の回転ベクトルは、回転角度 THETA でプログラム指令できるオフセットを使用して補間することもできます。したがって、PO[THT]を使用して、5 次以下の多項式をオフセット角に対してプログラム指令できます。

構文

`N...ORIROTC`	軌跡タンジェントに対する工具の回転を開始します
`N...A3= B3= C3= THETA=値`	配向ベクトルの回転を定義します
`N...A3= B3= C3= PO[THT]=(c2, c3, c4, c5)`	5 次以下の多項式でオフセット角度を補間します

向きを変更せずに、個別のブロックで回転をプログラム指令することもできます。

意味

6 軸座標変換の軌跡に対する工具回転の補間

`ORIROTC`:	軌跡タンジェントに対する接線方向の回転ベクトルを開始します
`THETA=`値:	ブロック終点までの到達回転角度(° 単位)
`THETA=`Θe:	回転ベクトルの終了角度が Θ_e の回転角度
`THETA=AC(…)`:	アブソリュート指令へのノンモーダル切り替え
`THETA=IC(…)`:	インクレメンタル指令へのノンモーダル切り替え
`PO[THT]=(c2, c3, c4, c5)`:	5 次多項式でオフセット角度を補間します

注記

回転ベクトルの補間 ORIROTC

軌跡タンジェントに対する工具回転を、工具オリエンテーションとは逆方向に開始できるのは、6 軸座標変換の場合のみです。

ORIROTC が動作中の場合

回転ベクトル ORIROTA はプログラム指令できません。プログラム指令すると、アラーム 14128「Absolute programming of tool rotation with active ORIROTC」(ORIROTC が動作中に工具回転のアブソリュートプログラム指令がされました)が発生します。

3 軸～ 5 軸座標変換の工具の旋回方向

工具の旋回方向は、3 軸～ 5 軸座標変換の場合と同様に、オイラー角、RPY 角、または方向ベクトルでプログラム指令できます。空間の工具の向きの変更は、大円弧補間 ORIVECT、旋回軸の直線補間 ORIAXES、円錐面上のすべての補間 ORICONxx、そし

て、さらに工具の 2 つの接点による空間の曲線の補間 ORICURVE をプログラミングして実行することもできます。

G....:	回転軸移動の詳細
X, Y, Z:	直線軸の詳細
ORIAXES:	機械軸または旋回軸の直線補間
ORIVECT:	大円弧補間(ORIPLANE と同じ)
ORIMKS: ORIWKS:	機械座標系での回転 ワーク座標系での回転 説明については、「工具オリエンテーションの回転」の章を参照してください。
A= B= C=:	機械軸位置のプログラミング
ORIEULER:	オイラー角による向きのプログラミング
ORIRPY:	RPY 角による向きのプログラミング
A2= B2= C2=:	仮想軸の角度のプログラミング
ORIVIRT1: ORIVIRT2:	仮想旋回軸を使用した向きのプログラミング (定義 1)、マシンデータ$MC_ORIAX_TURN_TAB_1 による定義 (定義 2)、マシンデータ$MC_ORIAX_TURN_TAB_2 による定義
A3= B3= C3=:	方向軸の方向ベクトルのプログラミング
ORIPLANE:	平面での補間(大円弧補間)
ORICONCW:	円錐面での右回り方向の補間
ORICONCCW:	円錐面での左回り方向への補間
ORICONTO:	接線方向の遷移による円錐面の補間
A6= B6= C6=:	円錐の回転軸のプログラミング(正規化ベクトル)
NUT=角度	円錐の開口角度(° 単位)
NUT=+179	180°以下の移動角度
NUT=-181	180°以上の移動角度
ORICONIO:	円錐面での補間

A7= B7= C7=:	中間旋回(正規化ベクトルとしてのプログラミング)
ORICURVE XH YH ZH (例: 多項式 PO[XH]=(xe, x2, x3, x4, x5)などによる)	工具の 2 つの接点間の移動を指定した旋回補間。終点だけでなく、追加の曲線多項式もプログラム指令できます。

注記

ORIAXES が有効で、工具オリエンテーションを旋回軸で補間した場合は、回転角度はブロック終点で、軌跡に対して開始されるだけです。

7.5.4 旋回処理のスムージング(ORIPATHS A8=, B8=, C8=)

向きの変更を輪郭上で一定の加速度で実行すると、軌跡移動が、特に輪郭のコーナ部で不必要に中断される場合があります。結果として生じる旋回処理の不連続変化は、別の中間ブロックを挿入してスムージングできます。再旋回時に ORIPATHS が有効な場合は、向きの変更が一定の加速でおこなわれます。この段階で、工具を後退できます。

機械製造メーカ

この機能を起動するための、事前設定されたマシンデータとセッティングデータについては、工作機械メーカの注意事項を参照してください。

次のマシンデータを使用すると、後退ベクトルの解釈方法を設定できます。

1. TCS では、工具方向で Z 座標を定義します。
2. WCS では、有効平面で Z 座標を定義します。

「軌跡に対する向き」機能の詳細については、以下を参照してください。

参照先:

機能マニュアル 応用機能; 多軸座標変換(F2)

構文

輪郭のコーナでは、軌跡に対する工具オリエンテーション全体を一定に保つために、さらに詳細なプログラミングが必要です。この移動の方向と軌跡長は、A8=X, B8=Y C8=Z 成分を使用してベクトルでプログラム指令します。

```
N...ORIPATHS A8=X B8=Y C8=Z
```

意味

ORIPATHS:	軌跡に対する工具オリエンテーション; 旋回処理の不連続変化がスムージングされます
A8= B8= C8=:	方向と軌跡長のベクトル成分
X, Y, Z:	工具方向への後退移動

注記

方向ベクトル A8、B8、C8 のプログラミング

このベクトルの長さが正確にゼロの場合は、後退移動は実行されません。

ORIPATHS

軌跡に対する工具オリエンテーションは、ORIPATHS で有効にします。有効にしない場合は、直線大円弧補間で、向きが旋回の開始から旋回の終了に渡されます。

7.6 旋回の圧縮(COMPON、COMPCURV、COMPCAD、COMPOF、COMPSURF)

方向座標変換(TRAORI)が有効で、工具オリエンテーションがプログラム指令されている(タイプを問わず)NC プログラムは、指定制限内であれば、圧縮できます。

プログラミング

工具の向き

方向座標変換(TRAORI)が有効な場合は、5 軸の機械では、工具の向きを次のようにプログラム指令できます(キネマティックスは影響しません)。

- 方向**ベクトル**のプログラミング:
  `A3=<...> B3=<...> C3=<...>`
- オイラー**角**または RPY **角**のプログラミング:
  `A2=<...> B2=<...> C2=<...>`

工具の回転

6 軸の機械では、工具オリエンテーションに加えて工具の回転もプログラム指令できます。

回転角度は、以下のようにプログラム指令します。

`THETA=<...>`

「工具オリエンテーションの回転 (ページ 414)」を参照してください。

注記

追加の回転をプログラム指令した NC ブロックは、回転角度が**一次で**変化する場合にのみ圧縮できます。つまり、PO[THT]=(...)を使用した多項式は回転角度にプログラム指令することはできません。

圧縮できる NC ブロックの一般的な構成

したがって、圧縮可能な NC ブロックの一般的な構文は、以下のようになります。

`N...X=<...> Y=<...> Z=<...> A3=<...> B3=<...> C3=<...> THETA=<...> F=<...>`

または

```
N...X=<...> Y=<...> Z=<...> A2=<...> B2=<...> C2=<...> THETA=<...> F=<...>
```

注記

位置データは直接(X90 など)、またはパラメータ設定で間接的に(X=R1*(R2+R3))入力できます。

回転軸位置を使用した工具オリエンテーションのプログラミング

工具オリエンテーションは、回転軸位置を使用して、次のような構文で指定することもできます。

```
N...X=<...> Y=<...> Z=<...> A=<...> B=<...> C=<...> THETA=<...> F=<...>
```

この場合は、大半径円弧補間がおこなわれるかどうかに応じて、2 つの異なる方法で圧縮が実行されます。大半径円弧補間をおこなわない場合は、圧縮された向きの変更が、回転軸の軸多項式により通常の方法で表わされます。

輪郭精度

選択した圧縮モード(MD20482 $MC_COMPRESSOR_MODE)に応じて、設定した軸別の許容範囲(MD33100 $MA_COMPRESS_POS_TOL)、または以下のチャネル別の許容範囲(セッティングデータで設定)が、ジオメトリ軸と旋回軸の圧縮で有効になります。

SD42475 $SC_COMPRESS_CONTUR_TOL (最大輪郭誤差)

SD42476 $SC_COMPRESS_ORI_TOL (工具オリエンテーションの最大角度誤差)

SD42477 $SC_COMPRESS_ORI_ROT_TOL (工具の回転角度の最大角度誤差) (6 軸の機械でのみ使用可能です)

参照先:

機能マニュアル 基本機能; 3 軸～ 5 軸の座標変換(F2)、
「旋回の圧縮」の章

起動と解除

コンプレッサ機能は、モーダル G 命令 COMPON、COMPCURV、COMPCAD、または COMPSURF を使用して起動できます。

コンプレッサ機能は COMPOF で終了します。

「NC ブロック圧縮(COMPON、COMPCURV、COMPCAD、COMPOF) (ページ 309)」を参照してください。

注記

旋回動作が圧縮されるのは、大半径円弧補間が有効なときのみです(つまり、旋回開始と旋回終了で特定される平面で工具オリエンテーションが変更されます)。

大半径円弧補間は、次の条件下で実行されます。

- MD21104 $MC_ORI_IPO_WITH_G_CODE = 0、
  ORIWKS が有効で、
  向きを (A3, B3, C3 または A2, B2, C2 を使用した) ベクトルとしてプログラム指令している。
- MD21104 $MC_ORI_IPO_WITH_G_CODE = 1 であり、
  ORIVECT または ORIPLANE が有効である。
  工具オリエンテーションは、方向ベクトルとして、または回転軸位置を使用してプログラム指令できます。G 命令 ORICONxx と ORICURVE のいずれかが有効な場合、または旋回角度の多項式(PO[PHI]と PO[PSI])をプログラムしている場合は、大半径円弧補間は実行されません。

例

以下のプログラムの例では、多角形定義により作成される円弧を圧縮します。工具オリエンテーションは同時に、円錐面を移動します。プログラム指令の複数の向きの変化が、不連続な処理にもかかわらず連続して実行された場合は、コンプレッサ機能により、旋回動作が滑らかになります。

プログラミング	コメント
DEF INT NUMBER=60	
DEF REAL RADIUS=20	
DEF INT COUNTER	
DEF REAL ANGLE	
N10 G1 X0 Y0 F5000 G64	
$SC_COMPRESS_CONTUR_TOL=0.05	; 輪郭の最大誤差= 0.05 mm
$SC_COMPRESS_ORI_TOL=5	; 旋回の最大誤差= 5°
TRAORI	
COMPCURV	; 移動は、多角形から作成された円弧を表わします。旋回は、Z 軸を中心として、開口角度 45°の円錐上を移動します。
N100 X0 Y0 A3=0 B3=-1 C3=1	
N110 FOR COUNTER=0 TO NUMBER	
N120 ANGLE=360*COUNTER/NUMBER	
N130 X=RADIUS*cos(角度) Y=RADIUS*sin(角度)	

プログラミング	コメント
A3=sin(角度) B3=-cos(角度) C3=1	
N140 ENDFOR	

7.7 向きの特性の有効化と無効化(ORISON、ORISOF)

「向きの特性のスムージング」は、G グループ 61 の命令を使用して、パートプログラムで有効化または無効化できます。これらの命令はモーダルです。

必要条件

- 5/6 軸座標変換を使用したシステム
- コンプレッサ機能 COMPCAD が有効

構文

```
ORISON
...
ORISOF
```

意味

ORISON:	向きの特性のスムージングを起動
ORISOF:	向きの特性のスムージングを解除

例

プログラムコード	コメント
...	
TRAORI()	; 方向座標変換の適用。
COMPCAD	; COMPCAD コンプレッサ機能の起動
ORISON	; オリエンテーションのスムージングの起動.
$SC_ORISON_TOL=1.0	; 工具オリエンテーションの最大角度誤差= 1.0°
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	

プログラムコード	コメント
...	
ORISOF	; 旋回スムージングの解除。
...	

向きが XZ 平面上で-45°から+45°へと、90°旋回します。旋回処理のスムージングにより、向きが、最大角度値の-45°または+45°のどちらにも達しなくなります。

7.8 キネマティックトランスフォーメーション

7.8.1 面端座標変換(TRANSMIT)の起動

前面座標変換(TRANSMIT)は、TRANSMIT 命令を使用してパートプログラムまたはシンクロナイズドアクションで有効になります。

構文

```
TRANSMIT
TRANSMIT(<n>)
```

意味

TRANSMIT:	1 番目の TRANSMIT データセットで TRANSMIT を有効にします。
TRANSMIT(n):	n 番目の TRANSMIT データセットで TRANSMIT を有効にします。

注記

チャネルで TRANSMIT 座標変換が有効になり、次のことが適用されます:

- 座標変換の解除:TRAFOOF
- 別の座標変換の適用:例: TRACYL、TRAANG、TRAORI

7.8.2 円筒補間(TRACYL)の起動

円筒座標変換(TRACYL)は、TRACYL 命令を使用してパートプログラムまたはシンクロナイズドアクションで有効になります。

構文

```
TRACYL(<d>)
TRACYL(<d>,<n>)
TRACYL(<d>,<n>,<k>)
```

意味

TRACYL(<d>):	1 番目の TRACYL データセットと作業直径**<d>**で TRACYL を有効にします。	
TRACYL (<d>,<n>):	**<n>**番目の TRACYL データセットと作業直径**<d>**で TRACYL を有効にします。	
<d>:	基準または作業直径 この値は 1 より大きくします。	
<n>:	TRACYL データセット番号(任意選択)	
	値の範囲:	1, 2
<k>:	パラメータ**<k>**は座標変換タイプ 514 にのみ関連します。	
	k = 0:	溝壁補正なし
	k = 1:	溝壁補正あり
	パラメータが指定されていない場合、次のパラメータ設定された基本位置が適用されます: $MC_TRACYL_DEFAULT_MODE_<n> <n> = TRACYL データセット番号	

注記

チャネル内で有効な TRACYL 座標変換は、以下によってオフに切り替えることができます。

- 座標変換の解除:TRAFOOF
- 別の座標変換の有効化:TRAANG、TRANSMIT、TRAORI など

例

プログラムコード	コメント
...	
N40 TRACYL(40.)	; 1 番目の TRACYL データセットと作業直径 40 mm で TRACYL を有効にします。
...	

詳細情報

プログラム構成

TRACYL 座標変換 513 (溝壁オフセットありの TRACYL)を使用して溝を加工するパートプログラムは通常、以下の手順から成ります。

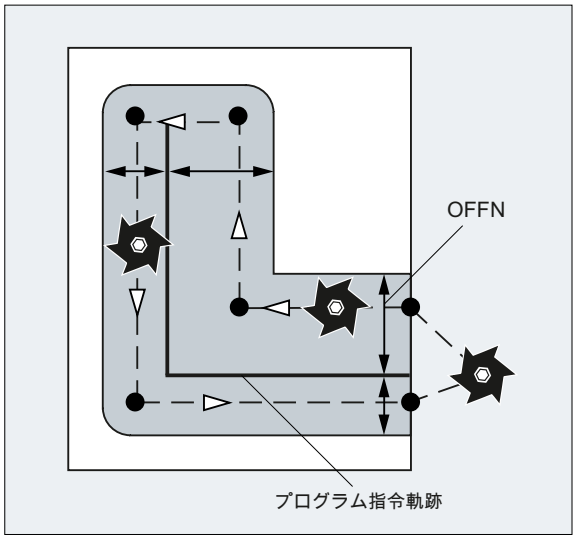
1. 工具の選択
2. TRACYL の選択
3. 適切な座標オフセット(フレーム)の選択
4. 位置決め
5. OFFN のプログラム指令
6. TRC の選択
7. 移動ブロック(TRC を位置決め後に溝壁へアプローチする)
8. 溝の中心線の輪郭
9. TRC の選択解除
10. 後退ブロック(TRC を後退後に溝壁から離れる)
11. 位置決め
12. TRAFOOF
13. オリジナルの座標シフト(フレーム)の再選択

輪郭オフセット(OFFN

TRACYL 座標変換 513 を使用して溝を加工するには、パートプログラムで、溝の中心線と**溝幅の半分**を OFFN アドレスを使用してプログラム指令します。

溝壁の損傷を避けるために、OFFN は工具径補正が有効なときにのみ機能します。

パートプログラム内で OFFN を変更できます。これにより、溝の中心線を中心からオフセットすることができます。



注記

溝壁の反対側の損傷を避けるために、OFFN には工具の半径以上の値を指定してください。

注記

OFFN は、TRACYL と併用した場合と、TRACYL と併用しない場合では、動作が異なります。TRACYL と併用していない場合でも、TRC が有効なときは OFFN が含まれるため、OFFN は、TRAFOOF の後でゼロにリセットしてください。

通知
OFFN の動作は、座標変換タイプによって異なります。 TRACYL 座標変換 513 (溝壁オフセットありの TRACYL)の場合、溝幅の半分を OFFN にプログラム指令します。 TRACYL 座標変換 512 (溝壁オフセットありの TRACYL)の場合、OFFN 値は TRC の許容範囲として機能します。

工具径補正(TRC)

TRACYL 座標変換 513 の場合、TRC は溝壁ではなく、プログラム指令された溝の中心を基準にして考慮されます。工具を溝壁の左に移動させるには、命令 G41 の代わりに G42 をプログラム指令するか、OFFN の値を負の符号付きで指定してください。

工具直径

TRACYL と直径が溝幅より小さい工具を使用した場合、直径が溝幅と同じ工具を使用した場合と同じ溝壁の形状は生成されません。精度を上げるには、工具の直径が溝幅より少しだけ小さくなるよう選択することをお勧めします。

軸の使用

注記

以下の軸は、位置決め軸と揺動軸のいずれにも使用できません。

- 円筒面の外周面方向のジオメトリ軸(Y 軸)
- 溝壁補正用の追加の直線軸(Z 軸)

7.8.3 角度がプログラム指令可能な傾斜角座標変換(TRAANG)の有効化(TRAANG)

角度がプログラム指令可能な傾斜角座標変換(TRAANG)は、TRAANG 命令を使用して、パートプログラムまたはシンクロナイズドアクションで有効にします。

構文

```
TRAANG
TRAANG()
TRAANG(, <n>)
TRAANG(<α>)
TRAANG(<α>,<n>)
```

意味

TRAANG: TRAANG():	1 番目の TRAANG データセットと最後の有効な角度<α>で TRAANG を有効にします。	
TRAANG(, <n>):	<n>番目の TRAANG データセットと最後の有効な角度<α>で TRAANG を有効にします。	
TRAANG(<α>):	1 番目の TRAANG データセットと角度<α>で TRAANG を有効にします。	
TRAANG(<α>,<n>):	<n>番目の TRAANG データセットと角度<α>で TRAANG を有効にします。	
<α>:	傾斜軸の角度(任意選択)	
	値の範囲:	-90° < α < + 90°
	角度を指定しない場合は、次のマシンデータでパラメータ設定した初期状態が有効になります。 MD2xxxx $MC_TRAANG_ANGLE_<n>	
<n>:	TRAANG データセット番号(任意選択)	
	値の範囲:	1, 2

注記

チャネルで有効な傾斜角座標変換 TRAANG は、以下を行って無効にします。

- 座標変換の解除:TRAFOOF
- 別の座標変換の適用:例: TRACYL、TRANSMIT、TRAORI

例

プログラムコード	コメント
N20 TRAANG(45)	; 1 番目の TRAANG データセットと角度 45°で TRAANG を有効にします。

7.8.4 研削盤での傾斜プランジ切削(G5, G7)

G 命令の G7 と G5 は、「傾斜軸」 (TRAANG)による研削盤での傾斜プランジ切削のプログラム指令を簡素化するのに使用されます。その結果、プランジ切削時に傾斜軸だけが移動します。

プランジ切削動作の必要な終了位置だけを X と Z にプログラム指令します。G7 では、X 軸の現在位置から開始して、NC が傾斜軸のプログラム指令された終了位置と角度 α を計算し、アプローチします。

開始位置は、2 つの直線の交点から計算されます。

- X 軸の現在値から少し離れた Z 軸に平行な直線
- プログラム指令された終了位置を通る傾斜軸に平行な直線

この後の G5 で、傾斜軸はプログラム指令された終了位置に移動します。

構文

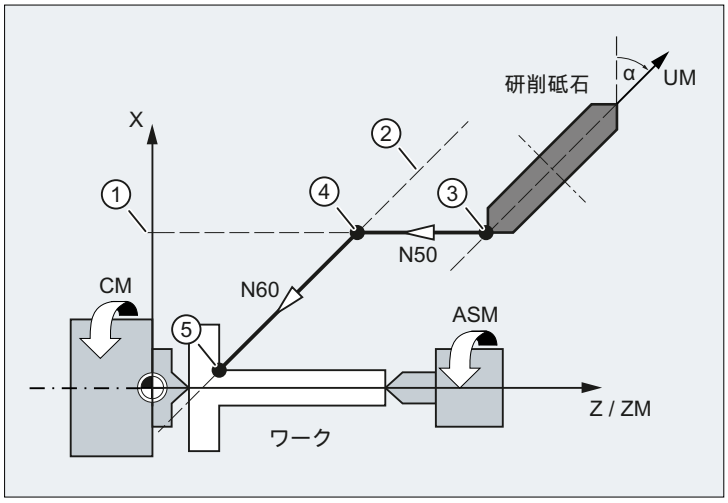
```
G7 <終了位置_X> <終了位置_Z>
G5 <終了位置_X>
```

意味

G7:	傾斜プランジ切削の開始地点を計算し、アプローチします。
G5:	傾斜軸をプログラム指令された終了位置に移動します。

<終了位置_X>:	X 軸の終了位置
<終了位置_Z>:	Z 軸の終了位置

例



① X 軸の現在値から少し離れて Z 軸に平行
② プログラム指令された終了位置を通って傾斜軸に平行
③ 開始位置
④ プランジ切削:移動
⑤ プランジ切削:終了位置
X ジオメトリ軸
Z ジオメトリ軸
ZM 機械軸
UM 機械軸

プログラムコード	コメント
N...G18	; XZ 平面を選択します
N40 TRAANG (45.0)	; TRAANG 座標変換、角度 = 45°を有効にします。
N50 G7 X40 Z70 F4000	; 開始位置を計算してアプローチします。
N60 G5 X40 F100	; 傾斜軸を終了位置に移動します。
N70 ...	

7.9 座標変換重畳(TRACON)の起動

TRAANG 座標変換は、TRAANG 命令を使用してパートプログラムまたはシンクロナイズドアクションで有効になります。

構文

```
TRACON(<座標変換番号>,<パラメータ_1>,<パラメータ_2,...)
...
TRAFOOF
```

意味

TRACON:	座標変換重畳を有効にします 別の座標変換が以前に有効であった場合、その変換は TRACON() によって自動的に解除されます。		
<座標変換番号>:	座標変換重畳の番号:		
	タイプ:	INT	
	データ範囲:	0 ... 2	
	値:	0, 1	1 番目/座標変換重畳のみ
		2	2 番目の座標変換重畳
		指定なし	0 または 1 と同じ意味
		注: 0、1、2 以外の値を指定すると、エラーメッセージが生成されます。	

<パラメータ_1>,<パラメータ_2,...:	座標変換重畳のパラメータ(例: 傾斜軸の角度) パラメータが設定されていない場合は、初期設定または最後に使用されたパラメータが有効となります。 以前のパラメータで初期設定が有効な場合は、指定したパラメータが確実に目的の順序で使用されるよう、コンマを使用してください。特に、<座標変換番号>の指定が不要な場合でも、1 つ以上のパラメータの前にはコンマが必要です。例: TRACON(, 3.7) TRANSMIT または TRAORI による TRACON (5 番目の軸の加工)では、2 番目の パラメータ Par_2 は、傾斜軸の角度として作用しません。 TRACYL による TRACON (円筒面の加工)では、Par がユニットの直径として使用されます。
TRAFOOF:	最後に有効にした座標変換(重畳)を解除します。

例

プログラムコード	コメント
...	
N230 TRACON(1,45.)	; 1 番目の座標変換重畳を有効にします。
	; 以前に有効であった座標変換は自動的に選択解除されます
	; 傾斜軸のパラメータは 45°です。
...	
N330 TRACON(2,40.)	; 2 番目の座標変換重畳を有効にします。
	; 傾斜軸のパラメータは 40°です。
...	
N380 TRAFOOF	; 2 番目の座標変換重畳を解除します。
...	

7.10 直交 PTP 移動

7.10.1 直交 PTP 移動(PTP、PTPG0、PTPWOC、CP)の有効化/無効化

直交ポイントツーポイント、つまり PTP 移動は G グループ 49 命令を使用して NC プログラムで有効化/無効化されます。

これらの命令はモーダルです。デフォルト設定は、直交軌跡移動(CP)での移動です。

CP とは対照的に、有効な PTP 移動の場合、直交ターゲット点のみが変換され、機械軸は同期制御で移動されます。

直交ターゲット点を機械軸値に一義的に変換するには、位置データと角度データに加えて、軸位置を識別するための情報も必要です。このデータは、設定可能なアドレス STAT (ページ 441)および TU (ページ 446)から取り出されます。

前提条件

座標変換 TRAORI、TRANSMIT、RCTRA、または ROBX が有効であること。

構文

```
PTP / PTPG0 / PTPWOC
...
CP
```

意味

PTP:	ポイントツーポイント移動 PTP の起動 G0 および G1 ブロック内のプログラムされた直交位置には、同期軸移動でアプローチします。
PTPG0:	ポイントツーポイント移動 PTPG0 の起動 同期軸移動でアプローチされるプログラムされた直交位置は、G0 ブロック内にのみ存在します。G1 ブロックでは、CP 軌跡移動に切り替えられます。

PTPWOC:	ポイントツーポイント移動 PTPWOC を有効化 (方向座標変換が有効な場合のみ可能) ただし、PTP と同様に、回転軸および旋回軸によって生じる補正移動はありません。
CP:	ポイントツーポイント移動の無効化と軌跡移動 CP の有効化 直交軌跡移動は CP で実行されます。

注記

PTPWOC

PTPWOC を座標変換 RCTRA または ROBX と併用することは無意味です。

例

参照:

- 例 1:ROBX 座標変換による 6 軸ロボットの PTP 移動 (ページ 449)
- 例 2:総合 5 軸座標変換の場合の PTP 移動 (ページ 450)
- 例 3:PTPG0 および TRANSMIT (ページ 451)

7.10.2 ジョイント(STAT)の位置を指定します

直交座標付きの位置データと工具オリエンテーションの指定だけでは、機械の位置を一義的に識別するには不十分です。同じ工具オリエンテーションに対して、複数のジョイント位置が考えられるからです。関連するキネマティックに応じて、8 つもの異なったジョイント位置が考えられます。これらの異なったジョイント位置は、座標変換に固有です。

不確定性を回避するために、ジョイント位置は STAT アドレスで指定されます。

注記

制御装置は PTP 移動に対してプログラムされた STAT 値のみを考慮します。有効な座標変換で移動する場合、通常は位置変更が不可能であるため、CP 移動は無視されます。有効な CP で移動する場合、ターゲット点の位置は起点から取られます。

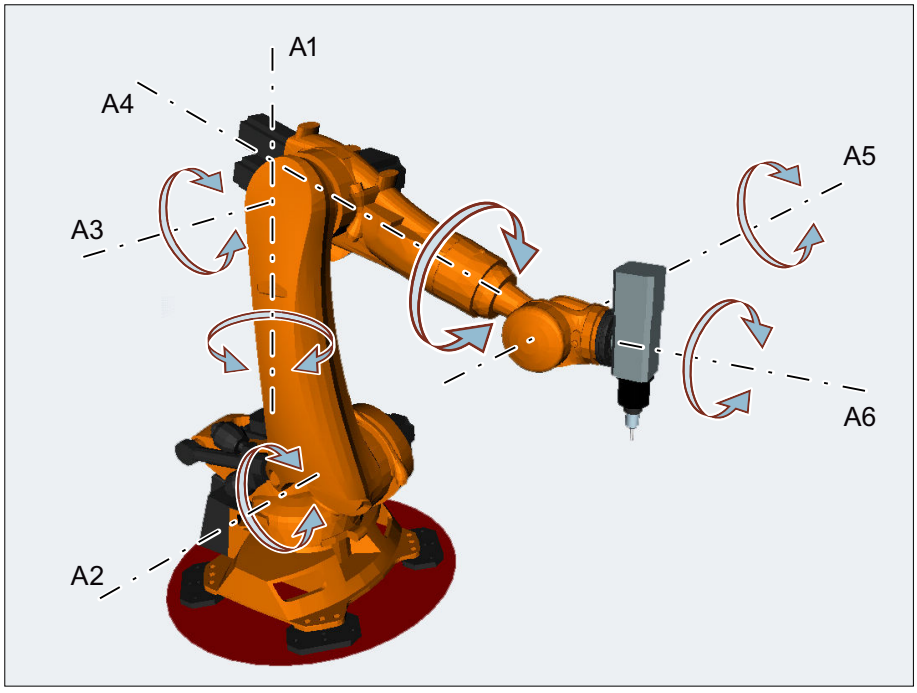
構文

STAT=<値>

意味

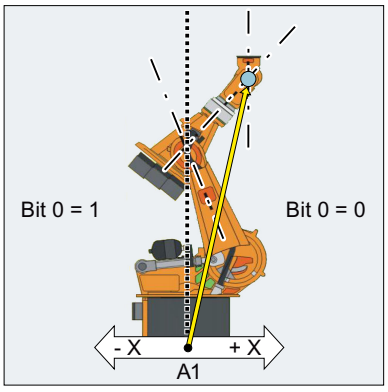
STAT:	ジョイント位置を指定するための設定可能なアドレス
<値>:	2 進数または 10 進数値 考えられる位置ごとに 1 ビットが含まれます。ビットの有効桁は特定の座標変換によって定義されます。

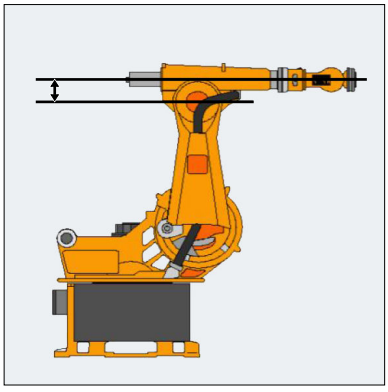
STAT の使用は、フライス主軸を備えた 6 軸の多関節ロボットで説明されます。ROBX ロボット座標変換を使用してキネティックトランスフォーメーションが実行されます(前提条件:コンパイルサイクル「RMCC/RCBX 座標変換拡張ロボティックス」がロードされ、有効であること)。



軸 A1、A2、および A3 は、多関節ロボットのメイン軸です。メイン軸とともに、ヘッド軸またはハンド/リスト軸としても指定される軸 A4、A5、および A6 が加工スペースに配置されます。ハンド/リスト軸がさらに移動した結果、特定の加工作業での必要に応じてフライス主軸を空間で旋回できます。同じ工具オリエンテーションを実現するために、さまざまな関節継手位置を使用できます。

加工に必要な関節継手位置は、調節可能な STAT アドレスのビット 0 ～ 2 のプログラミングで選択します。

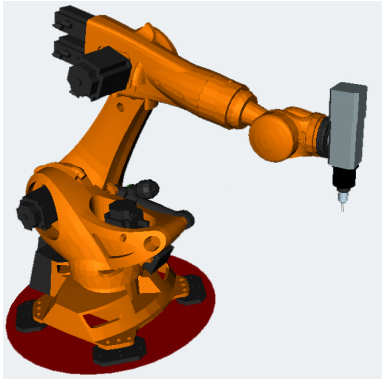
<table>
<tr><td rowspan="3">ビット 0</td><td colspan="2">手動軸の交点の位置(A4, A5, A6)</td><td rowspan="3"> 

例:ハンド/リスト軸の交点が基本範囲内にあります。</td></tr>
<tr><td>= 0</td><td>基本範囲(右ショルダー)
手動軸の交点の X 値が A1 座標系を基準にして正の場合、ロボットは基本範囲内にあります。</td></tr>
<tr><td>= 1</td><td>オーバーヘッド範囲(左ショルダー)
手動軸の交点の X 値が A1 座標系を基準にして負の場合、ロボットはオーバーヘッド範囲内にあります。</td></tr>
<tr><td rowspan="4">ビット 1</td><td colspan="3">軸 3 の位置
ビット 1 の値が変化する角度は、特定のロボットタイプによって異なります。軸 3 および 4 が交差するロボットには以下が適用されます。</td></tr>
<tr><td>= 0</td><td colspan="2">A3 < 0° (下部エルボー)</td></tr>
<tr><td>= 1</td><td colspan="2">A3 ≥ 0° (上部エルボー)</td></tr>
<tr><td colspan="2">注記:
軸 3 と 4 の間にオフセットのあるロボットの場合、ビット 1 の値が変化する角度はこのオフセットの大きさとは無関係です。</td><td> 
A3 と A4 の間のオフセット</td></tr>
<tr><td rowspan="3">ビット 2</td><td colspan="3">軸 5 の位置</td></tr>
<tr><td>= 0</td><td colspan="2">A5 ≤ 0° (ハンドフリップなし)</td></tr>
<tr><td>= 1</td><td colspan="2">A5 > 0° (ハンドフリップ)</td></tr>
</table>

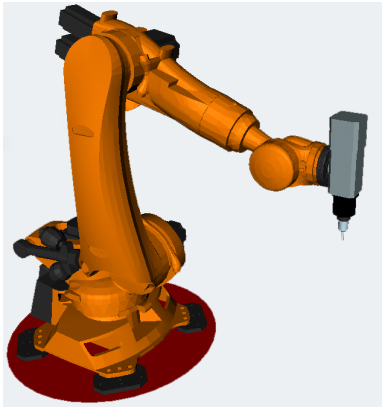
プログラム例:

プログラムコード	コメント
...	
N14 T="T8MILLD20" D1	; $TC_DP3[1,1]=132.95
N16 ORIMKS	
N17 G1 PTP X1665.67 Y0 Z1377.405 A=0 B=0 C=0 **STAT=**...F2000	; STAT 値は関節継手位置を定義します (以下を参照)
...	

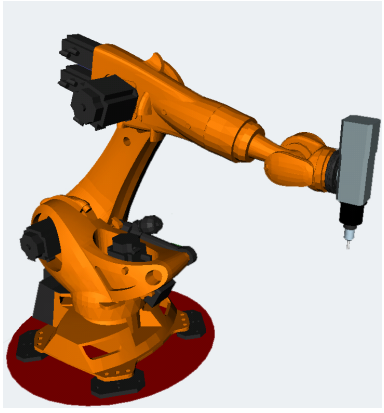
STAT=1 ('B001') → 左ショルダー
→ 下部エルボー
→ ハンドフリップなし



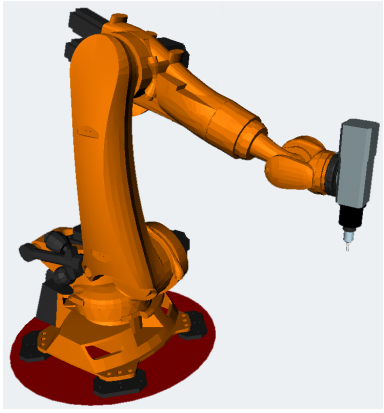
STAT=2 ('B010') → 右ショルダー
→ 上部エルボー
→ ハンドフリップなし



STAT=5 ('B101') → 左ショルダー
→ 下部エルボー
→ ハンドフリップ



STAT=6 ('B110') → 右ショルダー
→ 上部エルボー
→ ハンドフリップ



TRANSMIT

TRANSMIT の場合は、STAT アドレスを使用して、極に関する多義性を開始します。

回転軸が 180º 回転したり、CP のために回転する必要がある場合、輪郭は極を通過し、以下が適用されます。

ビット 0	$MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 1 または 2 に対してのみ関係します。	
	= 0	回転軸が+180º 移動、つまり右回りに回転します。
	= 1	回転軸が-180º 移動、つまり左回りに回転します。
ビット 1	$MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 0 に対してのみ関係します。	
	= 0	軸が極を通過します。回転軸は回転しません。
	= 1	軸が極の周囲を回転します。STAT のビット 0 が関係します。

7.10.3 軸角度の符号を指定します(TU)

回転軸が特別な移動方法(中間点など)の必要なしに+180°以上または-180°未満の軸角度にもアプローチできるようにするために、軸角度の符号を設定可能アドレス TU で指定してください。

注記

制御装置は PTP 移動に対してプログラムされた TU 値のみを考慮します。CP 移動は無視されます。

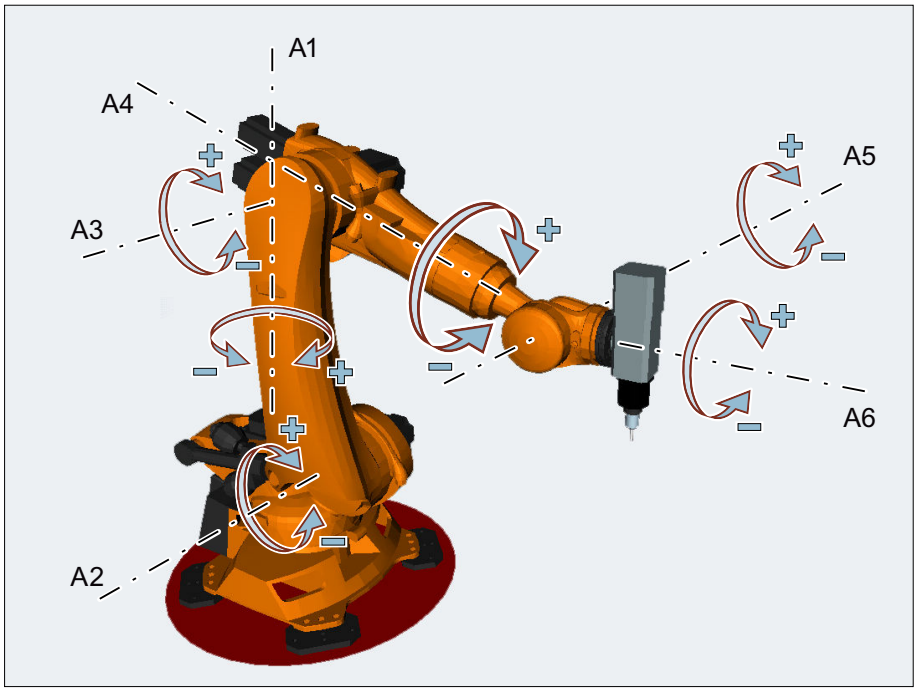
構文

TU=<値>

意味

TU:	軸角度の符号を指定するための設定可能アドレス			
<値>:	2 進数または 10 進数値 座標変換に関係する軸ごとに、軸角度(θ)の符号、すなわち移動方向を示すビットが存在します。			
	ビット	= 0	軸角度の符号: +	軸角度の範囲:0° ≤ θ < 360°
		= 1	軸角度の符号: -	軸角度の範囲: - 360° < θ < 0°

例:6 軸の多関節ロボット



ビット	意味	値	軸角度の符号	軸角度
ビット 0 [1)]	A1 の軸角度の符号	= 0	+	≥ 0°
		= 1	-	< 0°
ビット 1 [1)]	A2 の軸角度の符号	= 0	+	≥ 0°
		= 1	-	< 0°
ビット 2 [1)]	A3 の軸角度の符号	= 0	+	≥ 0°
		= 1	-	< 0°
ビット 3 [1)]	A4 の軸角度の符号	= 0	+	≥ 0°
		= 1	-	< 0°
ビット 4 [1)]	A5 の軸角度の符号	= 0	+	≥ 0°
		= 1	-	< 0°

ビット	意味	値	軸角度の符号	軸角度
ビット 5 [1)]	A6 の軸角度の符号	= 0	+	≥ 0°
		= 1	-	< 0°

1) ロボット軸のチャネル軸番号から得られた現在の TU ビット番号例では、ロボット軸(A1 ～ A6)がチャネルの最初の 6 本の軸であるため、TU ビット 0 ～ 5 が使用されます。ロボット軸の別のチャネル軸の割り当てでは、ロボット軸の TU ビット番号が、それに対応して変わります(例:ロボット軸は 3 ～ 8 番目のチャネル軸です。つまり、TU ビット 2 ～ 7 がロボット軸に使用されます)。

従って、TU=19 (TU='B010011 に対応)は以下を意味します。

ビット	値		軸角度
0	= 1	①	$\theta_{A1} < 0°$
1	= 1	①	$\theta_{A2} < 0°$
2	= 0	①	$\theta_{A3} \geq 0°$
3	= 0	①	$\theta_{A4} \geq 0°$
4	= 1	①	$\theta_{A5} < 0°$
5	= 0	①	$\theta_{A6} \geq 0°$

注記

移動範囲が±360°を超える軸の場合は、軸が常に最短の軌跡に沿って移動します。これは、軸位置を TU 情報で固有の指定をすることができないためです。

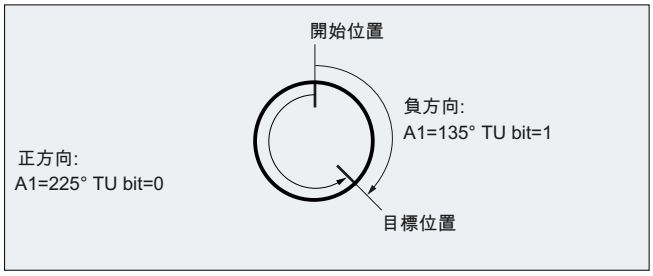
位置に対して TU がプログラム指令されていない場合は、MD30455 $MA_MISC_FUNCTION_MASK に応じて、より短い、または長い軌跡を移動します(上級機能の機能マニュアルの「PTP 移動のソフトウェア制限の考慮」の章を参照してください)。

TRANSMIT

TRANSMIT が有効の PTP 移動の場合、TU のアドレスは意味がありません!

例

次の図に示す回転軸の位置は、負または正方向へアプローチ可能です。回転位置は、アドレス A1 でプログラム指令されます。移動方向は、TU が指定されている場合にのみ絶対的にクリアです。



7.10.4 例 1:ROBX 座標変換による 6 軸ロボットの PTP 移動

以下の使用例では、直交 PTP 移動および関連する NC 命令を例として示しています。

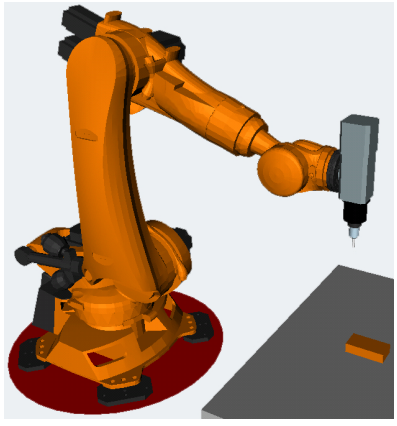


図 7-3　フライス主軸を備えた 6 軸の多関節ロボット

```
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
```

7.10.5 例 2:総合 5 軸座標変換の場合の PTP 移動

前提条件:直角の CA キネマティックがベースとして使用されていること。

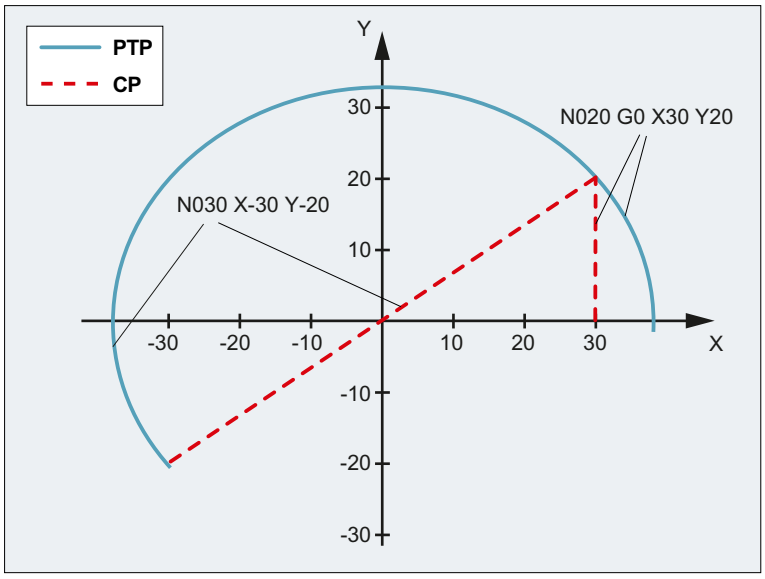
プログラムコード	コメント
TRAORI	; 座標変換 CA キネマティックスをオンにします
PTP	; PTP 移動を起動します
N10 A3=0 B3=0 C3=1	; 回転軸位置 C=0 A=0
N20 A3=1 B3=0 C3=1	; 回転軸位置 C=90 A=45
N30 A3=1 B3=0 C3=0	; 回転軸位置 C=90 A=90
N40 A3=1 B3=0 C3=1 **STAT=1**	; 回転軸位置 C=270 A=-45

回転軸位置の明確なアプローチ位置は、次のように選択します。

ブロック N40 で、回転軸は – **STAT=1** のプログラミングの結果 – 開始点(C=90, A=90)から終点(C=270, A=–45)までの距離より長い距離を移動します。一方、**STAT=0** では、回転軸は終点(C=90, A=45)までの最短距離に沿って移動します。

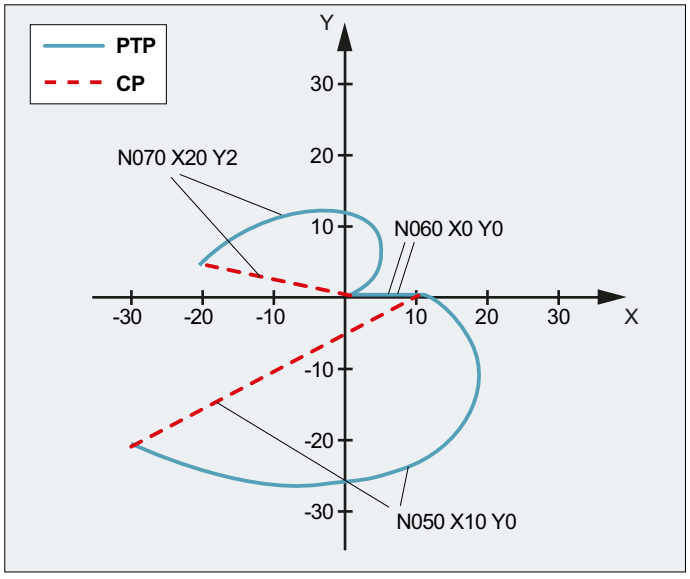
7.10.6 例 3:PTPG0 および TRANSMIT

PTPG0 および TRANSMIT での極を中心とした移動



プログラムコード	コメント
N001 G0 X30 Z0 F10000 T1 D1 G90	; 初期設定、アブソリュート指令
N002 SPOS=0	
N003 **TRANSMIT**	;TRANSMIT 座標変換
N010 **PTPG0**	; G0 ブロックごとに、自動的に PTP – その後、再び CP。
N020 G0 X30 Y20	
N030 X-30 Y-20	
N120 G1 X30 Y20	
N110 X30 Y0	
M30	

PTPG0 および TRANSMIT での極からの移動



プログラミング	コメント
N001 G0 X90 Z0 F10000 T1 D1 G90	; 初期設定
N002 SPOS=0	
N003 **TRANSMIT**	;TRANSMIT 座標変換
N010 **PTPG0**	; G0 ブロックごとに、自動的に PTP – その後、再び 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	

7.11 座標変換の選択時の制約事項

機能

座標変換は、パートプログラムまたは MDI で選択できます。注:

- 中間移動ブロックは挿入されません(面取り/半径)。
- スプラインブロック処理を完了してください。完了しない場合は、メッセージが表示されます。
- 精密工具補正の選択を解除してください(FTOCOF)。解除しない場合は、メッセージが表示されます。
- 工具径補正を解除してください(G40)。解除しない場合は、メッセージが表示されます。
- 動作中の工具長オフセットは、制御装置によって座標変換に含められます。
- 制御装置は座標変換前に、現在動作中のフレームを解除します。
- 制御装置は、座標変換に対応する軸に対して有効な動作範囲制限を解除します(WALIMOF に対応します)。
- プロテクションゾーンの監視は解除されます。
- 連続軌跡の制御と丸み付けは中断されます。
- マシンデータで指定したすべての軸を、ブロックに対して同期させてください。
- 交換した軸は、元に戻ります。戻らない場合は、メッセージが表示されます。
- 関連軸に対して、メッセージが出力されます。

工具交換

工具は、工具径補正機能が選択解除されているときにのみ交換できます。

工具長補正の変更と工具径補正の選択/解除は、同一ブロックにプログラム指令しないでください。

フレーム変更

基本座標系のみを基準とする命令はすべて使用できます(FRAME、工具径補正)。ただし、G91 (インクレメンタル指令)によるフレーム変更は、座標変換が動作していない場合とは異なり、個別に処理されません。移動するインクレメンタル値は、どのフレームが前のブロックで有効であったかにかかわらず、新しいフレームのワーク座標系で使用されます。

例外

座標変換で影響を受ける軸は、下記の目的では使用できません。

- 事前設定軸(アラーム)
- チェック点へアプローチ(アラーム)
- レファレンス点復帰動作(アラーム)

7.12 座標変換の解除(TRAFOOF)

予約 TRAFOOF 手順は、動作中の座標変換とフレームをすべて解除します。

注記

座標変換を選択解除するには、選択と同じ二次条件 (ページ 453)が適用されます。
この後に必要なフレームは、プログラミングを更新して有効にしてください。

構文

```
...
TRAFOOF
```

意味

TRAFOOF:	動作中の座標変換/フレームをすべて解除

キネマテイック結合 8

8.1 要素の削除(DELOBJ)

DELOBJ()機能は、割り当てられたシステム変数をデフォルト値にリセットすることで要素を「解除」します。

- キネマティック結合の要素
- プロテクションゾーン、プロテクションゾーン要素、干渉ペア
- 座標変換データ

構文

```
[<RetVal>=] DELOBJ(<CompType>[,,,<NoAlarm>)])
[<RetVal>=] DELOBJ(<CompType>,<Index1>[,,<NoAlarm>])
[<RetVal>=] DELOBJ(<CompType>[,<Index1>][,<Index2>][,<NoAlarm>])
```

意味

DELOBJ:	キネマティック結合、プロテクションゾーン、プロテクションゾーン要素、干渉ペア、座標変換データからの要素の削除

<CompType>:	削除する要素タイプ	
	データタイプ:	STRING
	値:"KIN_CHAIN_ELEM" **意味**:すべてのキネマティック要素のシステム変数: $NK_...	
	値:"KIN_CHAIN_SWITCH" **意味**:システム変数$NK_SWITCH[<i>]	
	値:"KIN_CHAIN_ALL" **意味**すべてのキネマティック要素とスイッチ。 「KIN_CHAIN_ELEM」および「KIN_CHAIN_SWITCH」による DELOBJ の連続呼び出しと同じです。	
	値:"PROT_AREA" **意味**:プロテクションゾーンのシステム変数: • $NP_PROT_NAME • $NP_CHAIN_NAME • $NP_CHAIN_ELEM • $NP_1ST_PROT	
	値:"PROT_AREA_ELEM" **意味**:機械プロテクションゾーンまたは自動工具プロテクションゾーンのプロテクションゾーン要素のシステム変数 • $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	
	値:"PROT_AREA_COLL_PAIRS" **意味**:干渉ペアのシステム変数: • $NP_COLL_PAIR • $NP_SAFETY_DIST	

	値:"PROT_AREA_ALL" **意味:**すべてのプロテクションゾーン、プロテクションゾーン要素、および干渉ペア(システム変数 $NP_...) 「PROT_AREA」「PROT_AREA_ELEM」、および「PROT_AREA_COLL_PAIRS」による DELOBJ の連続呼び出しと同じです。	
	値:"TRAFO_DATA" **意味:**すべての座標変換のシステム変数$NT_...	
<Index1>:	削除される 1 番目の要素のインデックス(**オプション**)	
	データタイプ:	INT
	初期値:	-1
	値の範囲:	-1 ≤ x ≤ (設定された要素の最大数 -1)
	値	**意味**
	0, 1, 2, ...	削除される要素のインデックス。
	-1	指定されたタイプのすべての要素が削除されます。**<**インデックス 2**>**は評価されません。
<Index2>:	削除する最後の要素の開始インデックス(**任意選択**) **<**Index2**>**がプログラム指令されていない場合、**<**Index1**>**で指定された要素のシステム変数だけが削除されます。	
	データタイプ:	INT
	初期値:	**<**Index1**>**で指定された要素のシステム変数だけが削除されます。
	値の範囲:	**<**Index1**> <** x ≤ (定された要素の最大数 -1)
<NoAlarm>:	アラームのマスク(**任意選択**)	
	データタイプ:	BOOL
	初期値:	FALSE
	値	**意味**
	FALSE	エラーが発生した場合(**<**RetVal**> <** 0)、プログラムの処理は停止してアラームが表示されます。.
	TRUE	エラーが発生した場合、プログラムの処理は停止せず、アラームは表示されません。 用途:戻り値に対応したユーザー固有の応答

<RetVal>:	機能の戻り値	
	データタイプ:	INT
	値の範囲:	0, -1, -2, ... -7
	値	**意味**
	0	エラーは発生していません。
	-1	パラメータなしで機能を呼び出します。最低でもパラメータ<要素タイプ>を指定してください。
	-2	<要素タイプ>は、未知の要素を示します。
	-3	<インデックス 1>は-1 より小さくなります。
	-4	<インデックス 1>は、設定された要素の数より大きくなります。
	-5	<インデックス 1>は、要素グループを削除する場合、-1 以外の値となります。
	-6	<インデックス 2>は、<インデックス 1>より小さくなります。
	-7	<インデックス 2>は、設定された要素の数より大きくなります。

8.2 名称を使用したインデックスの特定(NAMETOINT)

ユーザー固有の文字列または名称が、タイプ STRING のシステム変数配列に入力されます。システム変数の識別子と名称に基づいて、NAMETOINT()機能が名称に所属するインデックス値を決定します。インデックス値はその名前でシステム変数配列に保存されます。

構文

```
<RetVal> = NAMETOINT(<SysVar>,<Name>[,<NoAlarm>])
```

意味

NAMETOINT:	システム変数インデックスの特定	
<SysVar>:	タイプ STRING のシステム変数配列の名称	
	データタイプ:	STRING
	値の範囲:	タイプ STRING のすべての NC システム変数配列の名称
<Name>:	システム変数のインデックスの特定の対象となる文字列または名称	
	データタイプ:	STRING
<NoAlarm>:	アラームのマスク(**任意選択**)	
	データタイプ:	BOOL
	初期値:	FALSE
	値	**意味**
	TRUE	エラーが発生した場合、プログラムの処理は停止せず、アラームは表示されません。 用途:戻り値に対応したユーザー固有の応答
	FALSE	エラーが発生した場合(**<RetVal> < 0**)、プログラムの処理は停止してアラームが表示されます。

<RetVal>:	システム変数インデックスまたはエラーメッセージ	
	データタイプ:	INT
	値の範囲:	-1 ≤ x ≤ (設定された要素の最大数 -1)
	値	**意味**
	≥ 0	検索された名称が、指定されたシステム変数インデックスで見つかりました。
	-1	検索された名称が見つからなかったか、エラーが発生しました。

例

プログラムコード	コメント
DEF INT INDEX	
$NP_PROT_NAME[27]="Cover"	
...	
INDEX = NAMETOINT("$NP_PROT_NAME","Cover")	; INDEX == 27

キネマティック結合付きの衝突防止 9

注記

プロテクションゾーン

以下の章で指定されたプロテクションゾーンは、「ジオメトリック機械モデリング」機能を指しています。

参照先:
機能マニュアル、応用機能、「ジオメトリック機械モデリング」の章

9.1 干渉ペアをチェックします(COLLPAIR)

COLLPAIR()機能は、2 つのプロテクションゾーンが干渉ペアを形成するかどうかを決定します。

構文

```
[<RetVal> =] COLLPAIR(<Name_1>,<Name_2>[,<NoAlarm>)])
```

意味

COLLPAIR:	干渉ペアの要素であるかどうかをチェック		
<RetVal>:	機能の戻り値		
	データタイプ:	INT	
	値:	≥ 0	2 つのプロテクションゾーンが干渉ペアを形成しています。 戻り値 == 干渉ペアのインデックス m($NP_COLL_PAIR を参照)
		-1	2 つの文字列のどちらかが指定されていないか、2 つのうちの最低どちらかがゼロの文字列です。
		-2	1 番目のパラメータで指定されているプロテクションゾーンが見つかりません。
		-3	2 番目のパラメータで指定されているプロテクションゾーンが見つかりません。
		-4	指定された 2 つのプロテクションゾーンがどちらも見つかりません。
		-5	指定された両方のプロテクションゾーンが見つかりましたが、干渉ペアではありません。
<Name_1>:	1 番目のプロテクションゾーンの名称		
	データタイプ:	STRING	
	値の範囲:	パラメータ設定されたプロテクションゾーン名称	

<Name_2>:	2 番目のプロテクションゾーンの名称		
	データタイプ:	STRING	
	値の範囲:	パラメータ設定されたプロテクションゾーン名称	
<NoAlarm>:	アラームのマスク(任意選択)		
	データタイプ:	BOOL	
	値:	FALSE (初期設定)	エラーが発生した場合(<RetVal> < 0)、プログラムの処理は停止してアラームが表示されます。
		TRUE	エラーが発生した場合、プログラムの処理は停止せず、アラームは表示されません。 用途:戻り値に対応したユーザー固有の応答

9.2 干渉回避のマシンモデルの再計算要求(PROTA)

キネマティックチェーン$NK_...、幾何学的マシンモデリング、または干渉回避$NP_...がパートプログラムに記述されている場合、干渉回避の NC 内部機械モデルで変更が有効になるように、続いて PROTA 処理を呼び出してください。

構文

```
PROTA[(<Par>)]
```

意味

PROTA:	干渉回避のマシンモデルの再計算要求		
	● 先読み停止をおこないます。 ● ブロック内では単独にします。		
<Par>:	パラメータ(**オプション**)		
	データタイプ:	STRING	
	値:	---	パラメータなし。 マシンモデルが再計算されました。プロテクションゾーンの状態は維持されます。
		「R」	マシンモデルが再計算されました。プロテクションゾーンは、$NP_INIT_STAT に従って、それぞれの初期状態に設定されます。

必要条件

シミュレーション

PROTA 処理は、シミュレーション(simNC)と組み合わせてパートプログラムで使用しないでください。

例:シミュレーションが有効な間、PROTA 呼び出しを回避します。

プログラムコード	コメント
...	
IF $P_SIM == FALSE	; IF シミュレーションが有効ではありません
PROTA	; THEN 干渉モデルを再計算します
ENDIF	; ENDIF
...	

下記も参照

プロテクションゾーン状態の設定(PROTS) (ページ 470)

9.3 プロテクションゾーン状態の設定(PROTS)

PROTS()処理は、プロテクションゾーンの状態を、指定値に設定します。

構文

```
PROTS(<State>[,<Name_1>,...,<Name_n>])
```

意味

PROTS:	プロテクションゾーンの状態の設定		
	● ブロック内では単独にします。		
<State>:	指定されたプロテクションゾーンを設定する状態		
	データタイプ:	CHAR	
	値:	「A」または「a」	状態:有効
		「I」または「i」	状態:無効
		「P」または「p	状態:起動待ちまたは PLC 制御 [1]
		「R」または「r」	状態:初期化ステータスの NC 内部値 [2]
<Name_1> ... <Name_n>:	指定された状態に設定される複数のプロテクションゾーンの名称(**オプション**) 名称を指定しない場合、指定された状態がすべての定義済みプロテクションゾーンに対して設定されます。		
	データタイプ:	STRING	
	値の範囲:	パラメータ設定されたプロテクションゾーン名称	
	注: パラメータとして指定できるプロテクションゾーンの最大数は、プログラム行ごとの最大許容文字数によって異なります。		

[1] 起動と解除は以下によっておこなわれます。DB10.DBX234.0 - DBX241.7

[2] ステータスは、初期化ステータスの NC 内部値に設定されます。つまり、システム変数$NP_INIT_STAT の、前回の PROTA( ) (ページ 468)呼び出し時における値です。

9.4 :2 つのプロテクションゾーンの間隔の決定(PROTD)

PROTD()機能は、2 つのプロテクションゾーンの間隔を計算します。

機能の特性

- 間隔の計算は、プロテクションゾーン状態(起動済み、解除、起動待ち)とは無関係に実行されます。
- 2 つのプロテクションゾーンの間隔の計算には、$NP_USAGE = "C"または"A"とマークされたプロテクションゾーン項目のみが使用されます。プロテクションゾーンで$NP_USAGE = "V"とマークされたプロテクションゾーン項目は考慮されません。
- すべてのプロテクションゾーン項目が$NP_USAGE = "V"とマークされたプロテクションゾーンは、間隔の計算に使用できません。
- 間隔の計算は、前のブロックの終了時に有効だった位置で実行されます。
- メインラン計算に含まれる重畳(DRF オフセットや外部ゼロオフセットなど)は、機能**解析時間**での有効な値で間隔の計算に含まれます。

注記

同期制御

PROTD()機能を使用する場合、必要に応じて STOPRE 先読み停止でメインランと先読みを同期化するのはユーザーの責任で行います。

干渉

指定された 2 つのプロテクションゾーンの間に干渉がある場合、機能は 0.0 の値を返します。両方のプロテクションゾーンが互いに接触または交差している場合に、干渉が発生します。

干渉チェックの安全距離(MD10622 $MN_COLLISION_SAFETY_DIST)は、間隔の計算で考慮されません。

構文

```
[<RetVal> =] PROTD([<Name_1>],[<Name_2>],VAR <Vector>[,<System>])
```

意味

PROTD:	:2 つの指定されたプロテクションゾーンの間隔を計算します。
	● ブロック内では単独にします。

<RetVal>:	機能の戻り値:2 つのプロテクションゾーンのアブソリュート間隔値または干渉時の 0.0(上記の干渉の節を参照してください)		
	データタイプ:	REAL	
	値の範囲:	0.0 ≤ x ≤ +max.REAL 値	
<Name_1>, <Name_2>:	間隔を計算する 2 つのプロテクションゾーンの名称(**任意選択**)		
	データタイプ:	STRING	
	値の範囲:	パラメータ設定されたプロテクションゾーン名称	
	初期値:	"" (空の文字列) プロテクションゾーンを指定しない場合、この機能は干渉モデル内のすべての起動済みおよび起動待ちプロテクションゾーンから、現在の最小の間隔を計算します。	
<Vector>:	戻り値:プロテクションゾーン**<Name_2>**からプロテクションゾーン**<Name_1>**までの 3 次元の間隔ベクトル: • **<Vector>[0]**:ワールド座標系内の X 座標 • **<Vector>[1]**:ワールド座標系内の Y 座標 • **<Vector>[2]**:ワールド座標系内の Z 座標 干渉時: **<Vector> ==** ゼロベクトル		
	データタイプ:	VAR REAL [3]	
	値の範囲:	**<Vector>** [n]:0.0 ≤ x ≤ ±最大 REAL 値	
<System>:	間隔および間隔ベクトルの単位系(インチ/メトリック)(**任意選択**)		
	データタイプ:	BOOL	
	値:	FALSE (初期設定)	G グループ 13(G70、G71、G700、G710)の現在有効な G 命令に対応する単位系です。
		TRUE	設定されている基本単位系に対応する単位系 MD52806 $MN_ISO_SCALING_SYSTEM

キネマティック結合による座標変換 10

10.1 座標変換の有効化(TRAFOON)

キネティックチェーンで定義される座標変換は、予約された TRAFOON 手順によって有効化されます。呼び出しはブロック内では単独である必要があります。

注記

または、キネティックチェーンで定義される座標変換は、TRAORI や TRANSMIT などの通常の NC 命令によって有効にすることもできます。このために、ゼロに等しくない適切な値を$NT_TRAFO_INDEX のシステム変数に入力する必要があります。

$NT_TRAFO_INDEX の詳細については、「システム変数パラメータマニュアル」を参照してください。

構文

```
TRAFOON(<Trafoname>,<Diameter>,<k>)
```

意味

TRAFOON:	キネティックチェーンで定義される座標変換を有効化するための手順	
<Trafoname>:	座標変換データセットの名称	
	データタイプ:	STRING
	値の範囲:	$NK_NAME によって定義される座標変換データセットのすべての名称
	注記: 座標変換データセットの名称は一義的でなければなりません。 $NT_NAME で一度だけ発生するようにします。	
<Diameter>:	基準または作業直径(TRACYL のみ)	
	データタイプ:	REAL
	値は 1 より大きくしてください。	

<k>:	溝壁補正の使用を定義します(TRACYL のみ)。		
	データタイプ:	BOOL	
	値:	FALSE	溝壁補正を使用しない場合
		TRUE	溝壁補正を使用する場合
	TRACYL 座標変換タイプ 514 に対応します(溝壁補正はプログラム指令可能です)。**<k>**を指定していない場合、$NT_CNTRL[<n>]の 10 ビットのパラメータ設定が適用されます。		

例

プログラムコード	コメント
`TRAFOON["Trans_1"]`	`Trans_1` の名称による座標変換を有効にします。

10.2 機械計測後の方向座標変換の変更(CORRTRAFO)

キネティックチェーンによって定義された方向座標変換がある機械では、予約した CORRTRAFO 機能を使用して、機械計測後に機械のキネマティックモデルの旋回軸の補正ベクトルや方向ベクトルを変更することができます。

構文

```
<Corr_Status> = CORRTRAFO(<Corr_Vect>, <Corr_Index>, <Corr_Mode>
[, <No_Alarm>])
```

意味

CORRTRAFO:	ファンクションコール

<Corr_Status>:	機能の戻り値		
	データタイプ:	INT	
	値:	0	機能が異常なく実行されました。
		1	動作中の座標変換がありません。
		2	現在動作中の座標変換は、方向座標変換ではありません。
		3	動作中の方向座標変換は、キネティックチェーンで定義されていませんでした。
		10	<Corr_Index>コールパラメータは負です。
		11	<Corr_Mode>コールパラメータは負です。
		12	サブチェーンのセクションへの無効な参照です(<Corr_Index>の一の位)。値はサブチェーンの旋回軸の数より大きくしないでください。
		13	サブチェーンの旋回軸への無効な参照です(<Corr_Index>の一の位)。値はサブチェーンの旋回軸の数未満としてください。
		14	サブチェーンへの無効な参照です(<Corr_Index>の十の位)。値 0 と 1 のみが許容されます(パートまたは工具チェーンを参照)。この異常番号は、<Corr_Index>が参照するサブチェーンが存在しない場合に発生します。
		15	<Corr_Index>パラメータ($NT_CORR_ELEM_P または$NT_CORR_ELEM_T)で参照されたセクションに補正エレメントがありません。
		20	無効な補正モードです(<Corr_Mode>の 1 の位)。値 0 と 1 のみが許容されます。
		21	無効な補正モードです(<Corr_Mode>の 10/100 の位)。軸方向の書き込み時にゼロに等しくしてはいけないのは一の位のみです。
		30	<Corr_Mode>の 100 の位が無効です。値 0 と 1 のみが許容されます。
		31	<Corr_Mode>の 1000 の位が無効です。値 0 と 1 のみが許容されます。
		40	軸方向として扱われる方向ベクトルがゼロベクトルです。このエラーは、<Corr_Mode>の 1000 の位が

			0 に等しい場合に発生する場合があります。このパラメータの 1000 の位が 1 に等しい場合(最大補正の監視が無効化される)、ゼロベクトルも書き込むことができます。
		41	オフセットベクトルの補正では、少なくとも 1 つの座標における現在値との差分が、設定データ SD41610 $SN_CORR_TRAFO_LIN_MAX によって指定された最大値より大きくなります。 <Corr_Vect>パラメータはエラーベクトルによって上書きされます。これは、アラームによって処理が中止された場合にも適用されます(<No_Alarm>パラメータを参照)。 補正値が許容制限を超えた場合、エラーベクトルには、所定の補正値と制限値の間に、正しい符号の差分があります。 制限を超えていないコンポーネントの内容はゼロです。
		42	方向ベクトルの補正では、現在の座標と比べた角変位は、設定データ SD41611 $SN_CORR_TRAFO_DIR_MAX によって指定された最大値より大きくなります。
		43	書き込み権限がないため、システム変数を書き込もうとして拒否されました。
<Corr_Vect>:	補正ベクトル 補正ベクトルの内容は、次のパラメータ**<Corr_Index>**および<Corr_Mode>によって定義されます。 <Corr_Status> = 41 の場合、ベクトルの内容は上書きされます(前述の通り)。		
	データタイプ:	REAL	

<Corr_Index**>**:	補正エレメントを変更するセクション/ 方向ベクトルを変更する旋回軸のインデックス		
	データタイプ:	INT	
	<Corr_Index>パラメータは 10 進数コードです(1 から 10 の位):		
	1 の位:	セクションのインデックスまたはサブチェーンの旋回軸を含みます。	
	10 の位:	サブチェーンを指します。	
		0x	ワークチェーン
		1x	工具チェーン

<Corr_Mode>:	補正モード		
	データタイプ:	INT	
	<Corr_Index>パラメータは 10 進数コードです(一から千の位):		
	1 の位:	補正するエレメントを指定します。	
		xxx0	直線オフセットベクトルの補正
		xxx1	旋回軸の方向ベクトルの補正
	10 の位:	<Corr_Index>が指す補正エレメントの変更方法を指定します。	
		xx0x	補正ベクトルは、補正エレメントにすぐに書き込まれます。 このタイプは、補正エレメントをすぐに書き込むために使用できます。関連するシステムデータ($NK_OFF_DIR[<n>, ...])のインデックス<n>が既知である必要はありません。
		xx1x	0 と同様ですが、転送された補正値が世界座標で解釈される点は異なります。 0 と 1 のタイプの違いは、初期状態(すべての旋回軸の一が 0 に等しい)のキネティックチェーンに他の回転が含まれる場合に必ず発生します。
		xx2x	1 と同様ですが、補正値がセクション全体を指すという点が異なります。すなわち、補正エレメントに値が入力されると、セクション全体が補正値によって定義された長さに達するようになります。
		注記: 値 1 および 2 は、旋回軸の方向ベクトルを書き込む場合には使用できません。	
	100 の位:	<Corr_Vect>パラメータの内容の解釈方法を指定します。	
		x0xx	転送された補正ベクトル<Corr_Vect>には、補正エレメントの新しい長さ全体、または<Corr_Index>が<Corr_Mode>の 10 の位と組み合わせて参照するセクションが含まれます(絶対補正)。
		x1xx	転送された補正ベクトル<Corr_Vect>には、補正エレメントの現在の長さ、または<Corr_Index>

			が<Corr_Mode>の 10 の位と組み合わせて参照するセクションと比べた違いのみが含まれます(増分補正)。
		注記: 旋回軸の方向ベクトルの補正では、100 の位の内容は 0 でなければなりません。	
	1000 の位:	補正が次の最大値によって制限されるべきかどうかを指定します。 • SD41610 $SN_CORR_TRAFO_LIN_MAX または • SD41611 $SN_CORR_TRAFO_DIR_MAX	
		0xxx	最大補正の監視が有効です。
		1xxx	最大補正の監視は有効ではありません。
<No_Alarm>:	エラー発生時の動作(戻り値 > 0) (**オプション**)		
	データタイプ:	BOOL	
	値:	FALSE (初期設定)	エラーが発生した場合、プログラムの処理は停止し、アラーム 14103 が表示されます。
		TRUE	エラーが発生した場合、プログラムの処理は停止せず、アラームは表示されません。 用途:戻り値に対応したユーザー固有の応答

注記

機能呼び出し時にエラーが発生した場合、アラームが出力されるか、異常番号が返されますので(**<No_Alarm>**パラメータを参照してください)、エラー状態に適した方法で対応できます。問題の原因はアラームパラメータで詳細に記述されます。アラームの代わりに返された異常番号は、アラームパラメータと同じです。

CORRTRAFO の詳細情報

方向座標変換を伴う機械のキネマティック構造は、世界座標系の原点から始まる 1 つまたは 2 つのキネティックチェーン(サブチェーン)によって説明されます。2 つのチェーンの内の 1 つ、 **工具チェーン** は、工具のレファレンス点で終了し、もう 1 つのチェーンである **ワークチェーン** は、基本座標系の原点で終了します。

CORRTRAFO 機能は、機械上のレバーアームの長さと軸方向を方向座標変換によって計測し、これらの値を特別な補正エレメントに書き込みます。キネティックチェーンは、$ NK_TYPE で定義されるタイプ OFFSET のエレメントなどを使用して説明されます。

CORRTRAFO はセクションで動作する

2 つのサブチェーンは、それぞれ最大 4 つのセクションに分けることができます。

- セクション 1 はチェーンの起点で始まり、最初の旋回軸で終了します。
- セクション 2 は旋回軸 1 と旋回軸 2 の間のセクションです。
- セクション 3 は旋回軸 2 と旋回軸 3 の間のセクションです。
- セクション 4 は旋回軸 3 と工具チェーンまたはワークチェーンの終点の間のセクションです。

各セクションは、タイプ OFFSET または ROT_CONST の一定のチェーンエレメントを含む場合があります。

以下の図は、2 つの旋回軸による方向座標変換を示します。

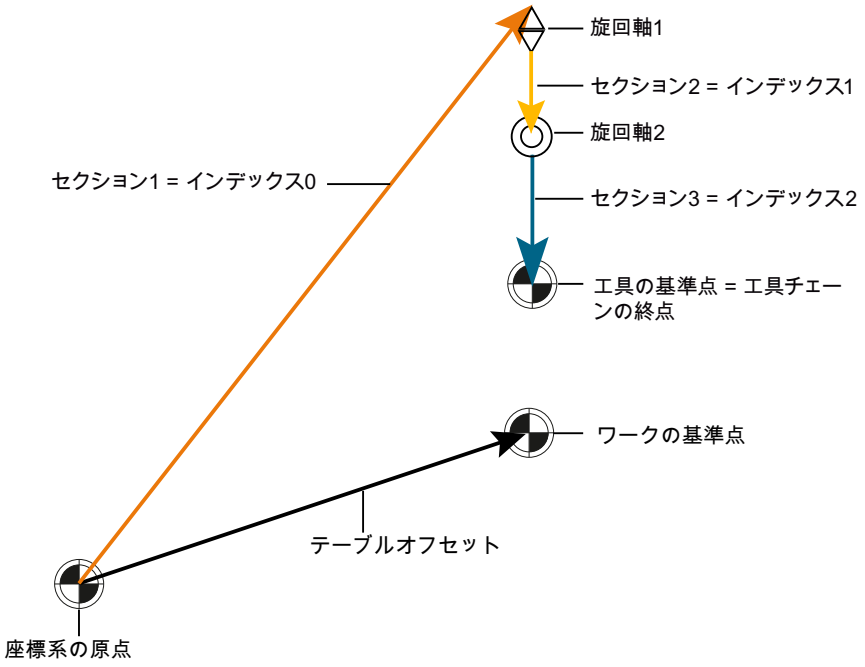


図 10-1　CORRTRAFO の例

セクションは明確に定義されます。キネマティックサブチェーンを起点から終点まで動かした場合、最初のセクションはインデックス 0 となり、次はインデックス 1、などと

なります。このとき、最後のセクションのインデックスは、常に旋回軸の数と等しくなります。

補正エレメント

$NT_CORR_ELEM_T[<n>, 0 ... 3]または$NT_CORR_ELEM_P[<n>, 0 ... 3]システム変数によって、各セクションの一定のキネティックチェーンエレメント($NK_TYPE[<n>] = "OFFSET"タイプのチェーンエレメント)を参照できます。機械計測中に特定された補正値は、CORRTRAFO 機能を使用してこれらのエレメントに書き込まれます。

座標変換インデックス = 1 の例:

- $NT_CORR_ELEM_T[1,0] = "C_AXIS_OFFSET"; セクション 1 の C 軸(旋回軸 1)のオフセットが補正エレメントとして定義されます。
- $NT_CORR_ELEM_T[1,1] = "B_AXIS_OFFSET"; セクション 2 の B 軸(旋回軸 2)のオフセットが補正エレメントとして定義されます。
- $NT_CORR_ELEM_T[1,2] = "BASE_TOOL_OFFSET"; セクション 3 の工具レファレンス点からの B 軸のオフセットが、補正エレメントとして定義されます。

$NT_CORR_ELEM_T/P[<n>, 0 ... 3]の参照順は、上述のセクションに対応する必要があります。つまり、最初の旋回軸の前にある$NT_CORR_ELEM_T/P [<n>, 0]に存在できるチェーンエレメントは 1 つのみである、などです。

CORRTRAFO 機能は、機械を計測することで特定した値を、この方法で定義した補正エレメントに書き込みます。補正値の変更は、<Corr_Mode>パラメータ経由で CORRTRAFO で定義されます。

チェーンを閉じる

$NT_CNTRL[<n>]システム変数でビット 7 またはビット 8 が設定された場合、追加の一定チェーンエレメント(チェーンの終点から機械の原点への接続を確立する)が、ワークチェーンの終点(ビット 7)または工具チェーンの起点(ビット 8)で、内部に自動的に挿入されます(「クローズチェーン」)。

これらの自動的に挿入されたエレメントは、外部に書き込むことはできず、読み込むことしかできません($AC_TRAFO_CORR_ELEM_P/T システム変数を参照)。

工具チェーンを閉じる点

$NT_CLOSE_CHAIN_T システム変数が空でない場合、工具チェーンはチェーンの終点ではなく、指定されたチェーンエレメントの終点で閉じています。この点の外側にあるその他のチェーンエレメントは、座標変換が有効になったときに、結果として対応するワークオフセットとなります。

旋回軸のインデックス

旋回軸の間の一定オフセットに加えて、旋回軸の方向ベクトルも CORRTRAFO 機能を使用して書き込むことができます。旋回軸のインデックスとは、キネマティックサブチェーンが起点から終点まで動くときに生じる、カウントがゼロで始まるインデックスです。そのため、旋回軸のインデックスは常に前のセクションのインデックスと同じになります。

旋回軸のインデックスは、$AC_TRAFO_ORIAX_LOC システム変数によって特定することもできます。

チェーンエレメントの最大許容変化率

チェーンエレメントの最大許容変化率は、旋回軸のオフセットベクトル用の SD41610 $SN_CORR_TRAFO_LIN_MAX と方向ベクトル用の SD41611 $SN_CORR_TRAFO_DIR_MAX の 2 つの設定データによって、制限することができます。SD41610 $SN_CORR_TRAFO_LIN_MAX は最大量を指定し、それによって個別のベクトルコンポーネントそれぞれを、その指令値に関して変更できます。SD41611 $SN_CORR_TRAFO_DIR_MAX は最大角を指定し、それによって軸ベクトルの方向を、その指令値に関して変更できます。指令値は常に、CORRTRAFO を呼び出したときに有効になる座標変換で有効な、対応する値です。これは、キネマティックデータの変更された内容が、座標変換の有効化後に CORRTRAFO 機能の動作方法に影響を与えないことを意味します。

工具オフセット 11

11.1 オフセットメモリ

オフセットメモリの構成

データフィールドはすべて、T と D 番号(「フラット D 番号」を除く)で呼び出すことができます。このフィールドには、工具のジオメトリデータに加えて、工具タイプなどのその他の情報も含まれます。

フラット D 番号の構成

「フラット D 番号の構成」は、工具管理機能が NC の外部で行われる場合に使用します。この場合は、対応する工具補正ブロックによって D 番号が作成されますが、工具には割り当てられません。

T は引き続き、パートプログラムでプログラム指令できます。ただし、この T は、プログラム指令 D 番号を基準としません。

ユーザー刃先データ

ユーザー刃先データは、マシンデータで設定できます。工作機械メーカの取扱説明書を参照してください。

工具パラメータ

注記

オフセットメモリの個々の値

オフセットメモリ P1 ～ P25 の個々の値は、プログラムでシステム変数を使用して読み取りと書き込みができます。その他のパラメータはすべて予約されています。

工具パラメータ$TC_DP6 ～$TC_DP8、$TC_DP10、および$TC_DP11 には、工具タイプに応じてもう 1 つの意味があります。$TC_DP15 ～$TC_DP17、$TC_DP19、および $TC_DP20 も同様です。

工具パラメータ番号(DP)	システム変数の意味	備考
$TC_DP1	工具タイプ	概要は、リストを参照してください
$TC_DP2	刃先位置	旋盤工具のみ

工具パラメータ番号(DP)	システム変数の意味	備考
ジオメトリ	**長補正**	
$TC_DP3	長さ 1	割当先
$TC_DP4	長さ 2	タイプとレベル
$TC_DP5	長さ 3	
ジオメトリ	**半径**	
$TC_DP6 [1)] $TC_DP6 [2)]	半径 1 /長さ 1 直径 d	フライス/旋削/研削工具 メタルソー
$TC_DP7 [1)] $TC_DP7 [2)]	長さ 2 /コーナ半径、先細フライス工具 切り込み幅 b のコーナ半径	フライス工具 メタルソー
$TC_DP8 [1)] $TC_DP8 [2)]	フライス工具のラウンディング半径 1 突出長 k	フライス工具 メタルソー
$TC_DP9 [1) 3)]	ラウンディング半径 2	予備
$TC_DP10 [1)]	角度 1 工具の先端	先細フライス工具
$TC_DP11 [1)]	角度 2 工具縦軸	先細フライス工具
磨耗	**長さと径補正**	
$TC_DP12	長さ 1	
$TC_DP13	長さ 2	
$TC_DP14	長さ 3	
$TC_DP15 [1)] $TC_DP15 [2)]	半径 1 /長さ 1 直径 d	フライス/旋削/研削工具 メタルソー
$TC_DP16 [1)] $TC_DP16 [3)]	長さ 2 /コーナ半径、先細フライス工具、切り込み幅 b のコーナ半径	フライス工具 メタルソー
$TC_DP17 [1)] $TC_DP17 [2)]	フライス工具のラウンディング半径 1 突出長 k	フライス加工/3 次元正面削り メタルソー
$TC_DP18 [1) 3)]	ラウンディング半径 2	予備
$TC_DP19[1)]	角度 1 工具の先端	先細フライス工具
$TC_DP20[1)]	角度 2 工具縦軸	先細フライス工具
工具ベース寸法/アダプタ	**長さオフセット**	
$TC_DP21	長さ 1	
$TC_DP22	長さ 2	
$TC_DP23	長さ 3	
テクノロジ		

工具パラメータ番号(DP)	システム変数の意味	備考
$TC_DP24	逃げ角	旋盤工具のみ
$TC_DP25		予備

1) 3 次元正面削り用フライス工具にも適用

2) メタルソー工具タイプの場合

3) 予備(SINUMERIK 840D sl では使用しません)

備考

形状変数では、複数の入力成分を使用できます(長さ 1 または半径など)。これらは合計されて 1 つの値(全長 1、全半径)となり、計算に使用されます。

不要なオフセット値には、値ゼロを割り当ててください。

輪郭工具の工具パラメータ$TC-DP1 to $TC-DP23

注記

表に含まれない$TC_DP7 などの工具パラメータは使用されません。つまり、その内容は意味がありません。

工具パラメータ番号(DP)	意味	切削 Dn		備考
$TC_DP1	工具タイプ			400 ～ 599
$TC_DP2	刃先位置			
形状	**長補正**			
$TC_DP3	長さ 1			
$TC_DP4	長さ 2			
$TC_DP5	長さ 3			
形状	**半径**			
$TC_DP6	半径			
形状	**制限角度**			
$TC_DP10	最小角度			
$TC_DP11	最大角度			
磨耗	**長さと径補正**			

工具パラメータ番号 (DP)	意味	切削 Dn		備考
$TC_DP12	磨耗長さ 1			
$TC_DP13	磨耗長さ 2			
$TC_DP14	磨耗長さ 3			
$TC_DP15	摩耗半径			
磨耗	**制限角度**			
$TC_DP19	最小磨耗角度			
$TC_DP20	最大磨耗角度			
工具ベース寸法/アダプタ	**長さオフセット**			
$TC_DP21	長さ 1			
$TC_DP22	長さ 2			
$TC_DP23	長さ 3			

基本値と摩耗値

得られる値はそれぞれ、基本値と摩耗値の合計です(半径の場合は$TC_DP6 + $TC_DP15、など)。基本計測($TC_DP21 – $TC_DP23)の値も 1 番目の刃先の工具長に加算されます。他のすべてのパラメータも、標準工具の有効工具長に影響する場合があり、また、この(アダプタ、旋回工具ホルダ、セッティングデータ)工具長にも影響します。

制限角度 1 と 2

制限角度 1 と 2 はそれぞれ、刃先中心点から刃先基準点へのベクトルに対応して、右回りにカウントされます。

11.2 追加オフセット

11.2.1 追加オフセットの選択(DL)

追加オフセットは、加工のなかでプログラム指令できる処理オフセットと見なすことができます。このオフセットは、刃先の形状データを表わしています。したがって、工具切削データの１つの成分です。

追加オフセットのデータは、DL 番号(DL:ロケーションに依存; 使用ロケーションでのオフセット)でアドレス指定し、操作画面を使用して入力します。

用途

使用するロケーションが原因で発生する寸法誤差は、追加オフセットで補正できます。

構文

```
DL=<番号>
```

意味

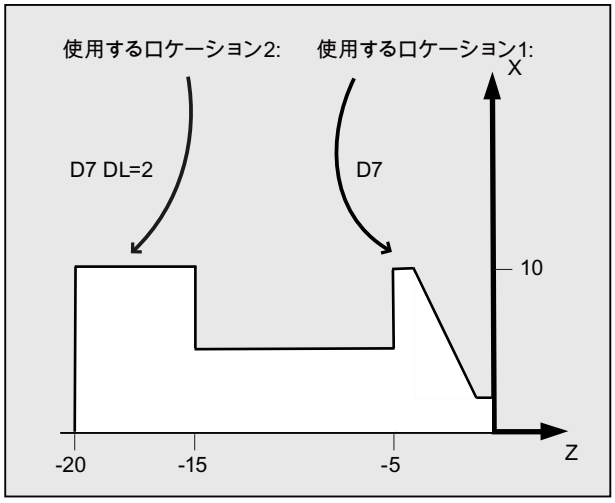
DL:	追加オフセットを起動する命令
<番号>:	有効にする追加工具オフセットデータを、<番号>パラメータで指定します。

注記

マシンデータを使用して、追加オフセットの数を定義し、さらにそれらのオフセットを起動します(→工作機械メーカのデータに従っているか、十分に注意してください)。

例

次のように、2 個の軸受座に同じ刃先を使用します。



プログラムコード	コメント
N110 T7 D7	; リボルバがロケーション 7 に位置決めされます。D7 および DL=1 が有効になり、次のブロックで移動します。
N120 G0 X10 Z1	
N130 G1 Z-6	
N140 G0 DL=2 Z-14	; D7 の他に DL=2 も有効になり、次のブロックで移動します。
N150 G1 Z-21	
N160 G0 X200 Z200	; 工具交換点へのアプローチ
...	

11.2.2 摩耗値とセットアップ値の指定($TC_SCPxy[t,d]、$TC_ECPxy[t,d])

摩耗値とセットアップ値は、システム変数を使用して読み取ったり、書き込んだりできます。この論理は、工具と刃先に対応するシステム変数の論理に基づいています。

システム変数

$TC_SCPxy[<t>,<d>]:	特定の形状パラメータに xy で割り当てた摩耗値です。ここで、x は摩耗値の番号に対応する値、y は形状パラメータの基準を決める値です。
$TC_ECPxy[<t>,<d>]:	特定の形状パラメータに xy で割り当てたセットアップ値です。ここで、x はセットアップ値の番号に対応する値、y は座標パラメータの基準を決める値です。
<t>:工具の T 番号 <d>:工具刃先の D 番号	

注記

定義した摩耗値とセットアップ値は、形状パラメータ、および他のオフセットパラメータ(D 番号)に加算されます。

例

長さ 1 の摩耗値を、工具<t>の刃先<d>に対して値 1.0 に設定します。

パラメータ: $TC_DP3 (長さ 1、旋削工具の場合)

摩耗値: $TC_SCP13 ~$TC_SCP63

セットアップ値: $TC_ECP13 ~$TC_ECP63

$TC_SCP43 [<t>,<d>] = 1.0

11.2.3 追加オフセットの削除(DELDL)

DELDL 命令は、工具の刃先の追加オフセットを削除します(メモリ空間が解放されます)。定義した摩耗値とセットアップ値が両方とも削除されます。

構文

```
DELDL[<t>,<d>]
DELDL[<t>]
DELDL
<状態>=DELDL[<t>,<d>]
```

意味

DELDL:	追加オフセットを削除する命令	
<t>:	工具の T 番号	
<d>:	工具刃先の D 番号	
DELDL[<t>,<d>]:	工具<t>の刃先<d>の追加オフセットをすべて削除します。	
DELDL[<t>]:	工具<t>のすべての刃先の追加オフセットをすべて削除します。	
DELDL:	工具オフセットユニットのすべての工具のすべての刃先の追加オフセットをすべて削除します(このチャネルでこの命令をプログラム指令した場合)。	
<状態>:	削除のステータスです	
	値:	意味
	0	削除が正常に完了しました。
	-	オフセットが削除されていません(パラメータ設定で指定した工具刃先が 1 個のみの場合)か、または完全には削除されていません(パラメータ設定で指定した工具刃先が複数の場合)。

注記

動作中の工具の摩耗値とセットアップ値は削除できません(基本的に、D または工具データの削除動作と同じです)。

11.3 工具補正の特殊処理

工具長さと摩耗の符号の評価は、セッティングデータ SD42900 ～ SD42960 で制御できます。

これは、ジオメトリ軸のミラーリング時または加工平面変更時の摩耗成分の動作、および工具方向の熱変位補正にも同様に適用されます。

磨耗値:

以下で摩耗値を参照する場合は、実際の摩耗値の合計($TC_DP12 ～$TC_DP20)、摩耗値によるサムオフセット($SCPX3 ～$SCPX11)、およびセットアップ値($ECPX3 ～ $ECPX11)として理解してください。

オフセットについて詳しくは、以下を参照してください。

参照先:

機能マニュアル 工具管理機能

セッティングデータ

SD42900 $SC_MIRROR_TOOL_LENGTH	工具長成分のミラーリング、および工具ベース寸法の成分です。
SD42910 $SC_MIRROR_TOOL_WEAR	工具長成分の摩耗値のミラーリングです。
SD42920 $SC_WEAR_SIGN_CUTPOS	摩耗成分の符号を、刃先位置に応じて評価します。
SD42930 $SC_WEAR_SIGN	摩耗寸法の符号を反転します。
SD42935 $SC_WEAR_TRANSFORM	摩耗値を座標変換します。
SD42940 $SC_TOOL_LENGTH_CONST	工具長成分をジオメトリ軸へ割り当てます。
SD42950 $SC_TOOL_LENGTH_TYPE	工具タイプに依存しない工具長成分の割り当てをおこないます。
SD42960 $SC_TOOL_TEMP_COMP	工具方向の熱変位補正値です。工具オリエンテーションをプログラム指令した場合も機能します。

参照先

機能マニュアル、基本機能; 工具オフセット(W1)

詳細情報

変更されたセッティングデータの適用

上記のセッティングデータを変更すると、次に工具刃先が選択されるまで、工具成分が再評価されません。工具がすでに有効で、この工具のデータを再評価する場合は、工具を再度選択してください。

これは、得られる工具長が、軸のミラーリングの状態が変化のために変更された場合も同様に適用されます。ミラーリング命令の後は、変更された工具長成分を起動するために、工具を再度選択してください。

旋回工具ホルダと新しいセッティングデータ

セッティングデータ SD42900 ～ SD42940 は、動作中の旋回工具ホルダの成分には影響を与えません。ただし、旋回工具ホルダの計算では常に、工具に対して、長さの合計結果は(工具長+摩耗+工具ベース寸法)になります。セッティングデータによる変更はすべて、長さの合計結果の計算に含まれます。つまり、旋回工具ホルダのベクトルは加工平面には影響されません。

注記

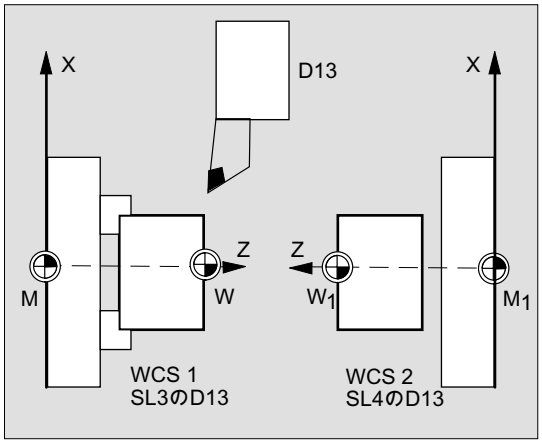
旋回工具ホルダの使用時には、多くの場合、これはミラーリングされない基本単位系のすべての工具の定義に役立ちます。これには、ミラーリングされる加工のみに使用される工具も含まれます。ミラーリングされた軸を使用して加工しているとき、工具ホルダは工具の実際の位置が正確に記述されるように回転します。すべての工具長成分は、セッティングデータを介した個々の成分評価の制御の必要なく、個々の軸のミラーリング状態に応じて、正しい方向で自動的に動作します。

その他のオプション用途

旋回工具ホルダ機能の使用は、旋削工具の物理的なオプションが機械に存在しない場合にも便利です。これは、向きの異なる工具が取り外しできないように設置されている場合も同様です。工具のサイズ決めは、基本の向きで一様におこなうことができます。この場合、加工に関連する寸法は実際の工具ホルダに応じて計算されます。

11.3.1 工具長のミラーリング

セッティングデータ SD42900 $SC_MIRROR_TOOL_LENGTH と SD42910 $SC_MIRROR_TOOL_WEAR をゼロに設定していない場合は、工具長成分と基本寸法の成分を、摩耗値とその関連軸を使用してミラーリングできます。



SD42900 $SC_MIRROR_TOOL_LENGTH

ゼロ**以外の**セッティングデータ:

工具長成分($TC_DP3、$TC_DP4、および$TC_DP5)および基本寸法成分($TC_DP21、$TC_DP22、および$TC_DP23)を、その関連軸に対してミラーリングします。また、符号の反転によるミラーリングもおこないます。

摩耗値は**ミラーリングされません**。摩耗値もミラーリングする場合は、セッティングデータ SD42910 $SC_MIRROR_TOOL_WEAR を設定してください。

SD42910 $SC_MIRROR_TOOL_WEAR

ゼロ**以外の**セッティングデータ:

工具長成分(その関連軸はミラーリングされています)の摩耗値も、符号の反転によりミラーリングされます。

11.3.2 摩耗の符号評価

セッティングデータ SD42920 $SC_WEAR_SIGN_CUTPOS と SD42930 $SC_WEAR_SIGN の設定がゼロ以外の場合は、摩耗成分の符号評価を反転できます。

SD42920 $SC_WEAR_SIGN_CUTPOS

ゼロ**以外の**セッティングデータ:

当該の刃先位置を持つ工具(旋削工具と研削工具、工具タイプは 400)の場合は、加工平面の摩耗成分の符号評価は刃先位置により異なります。このセッティングデータは、当該の刃先位置のない工具タイプに対しては無意味です。

次の表では、SD42920(ゼロ以外)で符号を反転した寸法を、X で示しています。

刃先位置	長さ 1	長さ 2
1		
2		X
3	X	X
4	X	
5		
6		
7		X
8	X	
9		

注記

SD42920 と SD42910 による符号評価は、相互には影響しません。たとえば、両方のセッティングデータを使用して寸法の符号を変更した場合は、符号はそのまま変更されません。

SD42930 $SC_WEAR_SIGN

ゼロ**以外の**セッティングデータ:

すべての摩耗寸法の符号を反転します。これは、工具の長さの他に、工具半径、ラウンディング半径などのその他の変数の両方に影響します。

正の摩耗寸法を入力した場合は、工具が「より短く」、「より細く」なります。「工具補正の特殊処理」、「変更されたセッティングデータの適用」の各章を参照してください。

11.3.3 動作中の加工運転の座標系(TOWSTD、TOWMCS、TOWWCS、TOWBCS、TOWTCS、TOWKCS)

機械のキネマティックス、または旋回工具ホルダの適用に応じて、これらの座標系のいずれかで計測した摩耗値を変換するか、または適切な座標系に座標変換します。

動作中の加工運転の座標系

次の座標系でおこなわれる工具長補正を使用すると、対応するグループ 56 の G 命令で、動作中の工具に工具長摩耗成分を取り入れることができます。

- 機械座標系(MCS)
- 基本座標系(BCS)
- ワーク座標系(WCS)
- 工具座標系(TCS)
- キネマティックトランスフォーメーションの工具座標系(KCS)

構文

```
TOWSTD
TOWMCS
TOWWCS
TOWBCS
TOWTCS
TOWKCS
```

意味

TOWSTD:	工具長摩耗値のオフセットの初期設定値
TOWMCS:	MCS の工具長のオフセット
TOWWCS:	WCS の工具長のオフセット
TOWBCS:	BCS の工具長のオフセット
TOWTCS:	工具ホルダの基準点での工具長のオフセット(旋回工具ホルダ)
TOWKCS:	工具ヘッドの工具長のオフセット(キネマティックトランスフォーメーション)

詳細情報

機能の違い

次の表に、重要な機能の違いを示しています。

G 命令	摩耗値	有効な旋回工具ホルダ
TOWSTD:	初期値、工具長	摩耗値が回転します。

TOWMCS:	MCS の摩耗値です。旋回工具ホルダが動作中でない場合、TOWMCS は、 TOWSTD と同じです。	得られる工具長のベクトルのみが回転し、摩耗は考慮されません。
TOWWCS:	WCS のなかで摩耗値を MCS に変換します。	工具ベクトルを、摩耗を考慮せずに、 TOWMCS として計算します。
TOWBCS:	BCS のなかで摩耗値を MCS に変換します。	工具ベクトルを、摩耗を考慮せずに、 TOWMCS として計算します。
TOWTCS:	WCS のなかで摩耗値を MCS に変換します。	工具ベクトルを、摩耗を考慮せずに、 TOWMCS として計算します。

TOWWCS、 TOWBCS、 TOWTCS:摩耗値が工具ベクトルに加算されます。

直線的座標変換

MCS のなかで定義した工具長が意味を持つのは、BCS からの直線座標変換によって MCS を生成した場合のみです。

非直線座標変換

たとえば、TRANSMIT による非直線座標変換が動作中の場合に、要求する座標系として MCS を指定すると、自動的に BCS を使用します。

キネマティックトランスフォーメーションが無効で、旋回工具ホルダが無効の場合

キネマティックトランスフォーメーションが無効で、旋回工具ホルダも無効の場合は、(WCS を除く)他の 4 つの座標系がすべて統合されます。この場合のみ、WCS が他の座標系と異なります。評価の必要があるのは工具長のみであるため、各座標系間の平行移動は無関係です。

参照先:
工具補正について詳しくは、以下を参照してください。
機能マニュアル 基本機能; 工具オフセット(W1)

計算に含める摩耗値

セッティングデータ **SD42935 $SC_WEAR_TRANSFORM** は、次の 3 つの摩耗成分のどれが、回転の対象となるかを定義します。

- 磨耗
- サムオフセット(仕上げ)
- 荒削りサムオフセット

この回転では、アダプタの座標変換、または以下の G 命令のいずれかが有効な場合には旋回工具ホルダを使用します。

- `TOWSTD`
  基本設定です。工具長の補正の場合。
- `TOWMCS`
  機械座標系(MCS)内の摩耗値です。
- `TOWWCS`
  ワーク座標系(WCS)内の摩耗値です。
- `TOWBCS`
  基本座標系(BCS)の摩耗値です。
- `TOWTCS`
  工具ホルダ固定部での工具座標系内の摩耗値(工具ホルダレファレンス点 T)です。
- `TOWKCS`
  キネマティックトランスフォーメーションの工具ヘッドの座標系の摩耗値です。

注記

個々の摩耗成分の評価(ジオメトリ軸への割り当て、符号評価)は、次の要素に影響を受けます。

- 有効な平面
- アダプタ座標変換
- セッティングデータ:
  - SD42910 $SC_MIRROR_TOOL_WEAR
  - SD42920 $SC_WEAR_SIGN_CUTPOS
  - SD42930 $SC_WEAR_SIGN
  - SD42940 $SC_TOOL_LENGTH_CONST
  - SD42950 $SC_TOOL_LENGTH_TYPE

11.3.4 工具長と平面変更

セッティングデータ SD42940 $SC_TOOL_LENGTH_CONST の設定がゼロ以外の場合は、平面の変更時に、長さ、摩耗、および基本寸法などの工具長成分を、旋削工具と研削工具のジオメトリ軸に割り当てることができます。

SD42940 $SC_TOOL_LENGTH_CONST

ゼロ**以外の**セッティングデータ:

工具長成分(長さ、摩耗、および工具ベース寸法)のジオメトリ軸への割り当ては、加工平面の変更時(G17 ～ G19)には変わりません。

次の表は、旋削工具と研削工具(工具タイプ 400 ～ 599)のジオメトリ軸への工具長成分の割り当てを示します。

内容	長さ 1	長さ 2	長さ 3
17	Y	X	Z
*)	X	Z	Y
19	Z	Y	X
-17	X	Y	Z
-18	Z	X	Y
-19	Y	Z	X

*) 各値が 0 以外で、6 個のリスト値のいずれにも等しくない場合は、値は 18 として使用します。

次の表は、他のすべての工具(工具タイプ< 400 または> 599)に対する工具長成分のジオメトリ軸への割り当てを示しています。

動作平面	長さ 1	長さ 2	長さ 3
*)	Z	Y	X
18	Y	X	Z
19	X	Z	Y
-17	Z	X	Y
-18	Y	Z	X
-19	X	Y	Z

*) 各値が 0 以外で、6 個のリスト値のいずれにも等しくない場合は、値は 17 として使用します。

注記

この表では、3 つまでのジオメトリ軸を X、Y、Z に指定していることが前提です。補正と軸の割り当てを特定するのは、軸識別子ではなく、軸の順序です。

11.4 オンライン工具補正

11.4.1 多項式機能(FCTDEF)の定義

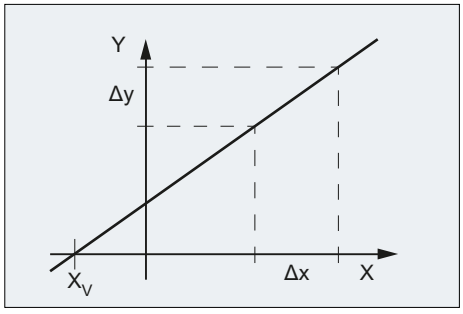
特定の目立て方法(目立てローラなど)には、目立てローラを送り込むと研削砥石の半径が連続して(直線的に)縮小するという特徴があります。この方法では、目立てローラの切り込みとそれぞれの長さの磨耗値の書き込みの間に一次関数が必要となります。直線機能は、予約手順 FCTDEF(...)を使用して三次多項式機能まで定義されます。

直線方程式

$y = f(x) = a_0 + a_1 * x_1$

a_1:直線の傾き、$_1 = \Delta x / \Delta y$

a_0:$a_0 = -a1 * X_V$ の場合の X 軸に沿った直線の移動



構文

```
FCTDEF(<Func>,<LLimit>,<ULimit>,<a0>,<a1>,<a2>,<a3>)
```

意味

FCTDEF(...):	PUTFTOCF(...)の多項式機能の定義: $y = f(x) = a_0 + a_1*x + a_2*x^2 + a_3*x^3$	
<Func>:	ファンクション番号	
	データタイプ:	INT
	値の範囲:	1, 2, 3
<LLimit>:	下限値	
	データタイプ:	REAL

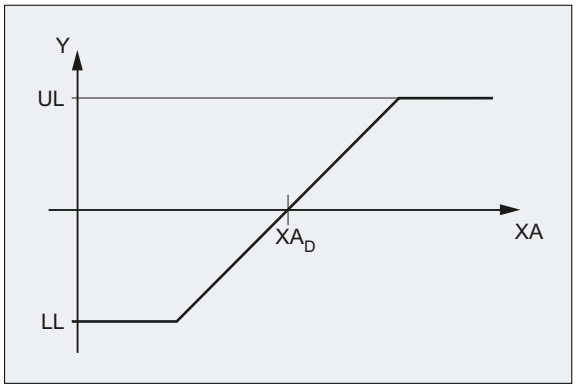
<ULimit>:	上限値	
	データタイプ:	REAL
<a0>,<a1>,<a2>,<a3>:	多項式関数の係数です。	
	データタイプ:	REAL

例

定義

- ファンクション番号:1
- 下限値および上限値: -100、100
- 特性の変化率:$a_1 = 1$
- 動作点は特性の中心に位置する必要があります。機能が NC プログラムで定義される場合の WCS の軸 XA のセットポイント位置に基づいて、特性を負の Y 方向に移動してください。$a_0 = -a_1 * XA_D = -1 * \$AA_IW$
- $a_2 = a_3 = 0$

Characteristic



UL　上限値

LL　下限値

XA_D　機能が NC プログラムで定義されている場合の軸 XA のセットポイント

プログラミング

プログラムコード	コメント
FCTDEF(1,-100,100,-$AA_IW[XA],1)	; 機能の定義

11.4.2 オンライン工具補正の連続書き込み(PUTFTOCF)

予約手順 PUTFTOCF(...)を使用して、オンライン工具補正が FCTDEF(...) (ページ 502)で以前に定義した多項式機能に基づいて実行されます。

注記

オンライン工具オフセットをシンクロナイズドアクションを使用して実現することもできます。

詳細情報については、『ファンクションマニュアル』の「シンクロナイズドアクション」を参照。

構文

```
PUTFTOCF(<Func>,<RefVal>,<ToolPar>,<Chan>,<Sp>)
```

意味

PUTFTOCF(...):	FCTDEF(...)で定義した多項式機能を使用して、オンライン工具補正をブロックごとに連続書き込み	
<Func>:	FCTDEF(...)による機能定義で定義された機能番号	
	データタイプ:	INT
	値の範囲:	1, 2, 3
<RefVal>:	オフセットの算出基準となる指令値です(軸のセットポイントなど)。	
	データタイプ:	VAR REAL
<ToolPar>:	オフセット値がインクルードされる摩耗パラメータ(長さ 1、2、または 3)の番号です。	
	データタイプ:	INT
<Chan>:	オンライン工具補正が有効となるチャネルの番号です。 **注記:** オフセットが有効なチャネルで無効な場合にのみ必要です。	
	データタイプ:	INT

<Sp>:	オンライン工具補正が有効となる主軸の番号です。 **注記:** オフセットが現在使用されている有効な工具ではなく、無効なといしに適用される場合にのみ必要です。	
	データタイプ:	INT

11.4.3 オンライン工具補正の書き込み、個別(PUTFTOC)

機能

予約手順 PUTFTOC(...)を使用して、オンライン工具補正が固定のオフセット値に基づいて実行されます。

構文

```
PUTFTOC(<CorrVal>,<ToolPar>,<Chan>,<Sp>)
```

意味

PUTFTOC(...):	オンライン工具補正の書き込み	
<CorrVal>:	摩耗パラメータに追加されるオフセット値。	
	データタイプ:	REAL
<ToolPar>:	オフセット値がインクルードされる摩耗パラメータ(長さ 1、2、または 3)の番号です。	
	データタイプ:	INT
<Chan>:	オンライン工具補正が有効となるチャネルの番号です。 **注記:** オフセットが有効なチャネルで無効な場合にのみ必要です。	
	データタイプ:	INT

<Sp>:	オンライン工具補正が有効となる主軸の番号です。 **注記:** オフセットが現在使用されている有効な工具ではなく、無効なといしに適用される場合にのみ必要です。	
	データタイプ:	INT

11.4.4 オンライン工具オフセットの起動/解除(FTOCON/FTOCOF)

オンライン工具補正は、G 命令 FTOCON および FTOCOF を使用して有効または無効にできます。

構文

```
FTOCON
...
FTOCOF
```

意味

FTOCON:	オンライン工具補正を起動します 命令は、オンライン工具補正が有効になるチャネルでプログラミングしてください。
FTOCOF:	オンライン工具長補正の解除 命令は、オンライン工具補正が無効になるチャネルでプログラミングしてください。 **注記:** FTOCOF では、軸は工具オフセットを超えて移動しません。ただし、PUTFTOC/PUTFTOCF で計算された値は切削固有オフセットデータに残ります。 オンライン工具補正を最終的に無効にするには、工具(T...)を FTOCOF 後にもう一度選択/選択解除してください。

11.5 3 次元工具径補正

11.5.1 3 次元外周加工用の 3 次元工具径補正(CUT3DC、CUT3DCD、ISD)

3 次元外周加工用の 3 次元工具径補正(3D TRC)は、限界面を考慮しない場合、モーダルで有効な G 命令、CUT3DC または CUT3DCD を使用して選択されます。

実際の有効化は、G41 または G42 を使用して実施されます。工具径補正は G40 によって無効化されます。

構文

```
G41/G42 ORIC/ORID ISD=...CUT3DC/CUT3DCD CDOF2 X...Y...Z...
...
G40 X...Y...Z...
```

意味

CUT3DC:	外周加工用の 3D TRC(5 軸座標変換が有効な場合のみ)	
CUT3DCD:	標準工具とは違う工具を基準にした外周加工用の 3D TRC (5 軸座標変換が有効な場合のみ) 半径差は工具パラメータ$TC_DP15 によって指定されます。	
G41/G42 X...Y...Z...:	工具径補正の有効化	
	G41:	輪郭の左側の工具径補正
	G42:	輪郭の右側の工具径補正
	注記: 有効化は線形ブロック(G0/G1)で実施する必要があります。	
CDOF2:	3 次元外周加工用の衝突検出の無効化	
ORIC/ORID:	外側コーナーでの向きの変更の動作が、G 命令 ORIC および ORID によって指定されます。	
	ORIC:	外側コーナーでの向きの変更を、挿入する円弧ブロックに重畳します。
	ORID:	外側コーナーでの向きの変更が、挿入する円弧ブロックの前に実行されます。

ISD=<値>:	ISD アドレスを使用すると、外周加工と**有効な** 3 次元工具径補正について、工具の切り込み深さを変更できます。	
	<値>:	切り込み深さの長さ
G40 X...Y...Z...:	工具径補正の無効化 **注記:** 無効化は線形ブロック(G0/G1)でジオメトリ軸の移動によって実施する必要があります。	

注記

3D TRC を選択するための G 命令はアプローチブロックで、つまり通常は G41 または G42 を含むブロックで使用されます。

G41 または G42 は、補正に関連するジオメトリ軸の移動動作なしで、ブロックでプログラム指令することもできます。この場合、アプローチブロックはそのようなブロックに続く最初の移動ブロックとなります。

有効な工具径補正による 3D TRC タイプの変更は、アラームなしで無視されます。

例

プログラムコード	コメント
	; 工具 D1 の定義:
$TC_DP1[1,1] = 120	; タイプ (エンドミル)
$TC_DP3[1,1] = 20	; 長さ補正ベクトル
$TC_DP6[1,1]=8	; 半径
N10 X0 Y0 Z0 T1 D1 F12000	; 工具の選択。
N20 TRAORI(1)	; 座標変換の起動。
N30 G42 ORIC ISD=10 **CUT3DC** G64 X30	; 3 次元外周加工の起動、
	; 外側コーナーでの連続的な向きの変更、
	; 切り込み深さ:10 mm
N40 ORIWKS A30 B15	; 軸位置の指定によるコーナーでの向きの変更。
N50 Y20 A3=1 C3=1	; 向きが変更される移動ブロック、
	; 方向ベクトルによる向きの指定。
N60 X50 Y30	; 一定の向きの移動ブロック、
N70 Y50 A3=0.5 B3=1 C3=5	; 向きが変更される移動ブロック。
N80 M63	; 移動情報のないブロック。
N90 X0 ISD=20	; 切り込み深さが変更される移動ブロック。
N100 G40 Y0	; 工具径補正を解除。
N110 M30	

詳細情報

経路と向き

ここでは、軌跡(主輪郭)、および対応する向きを定義して、使用する外周加工のタイプを決めます。この加工タイプでは、軌跡上の工具の形状は関係ありません。工具の接触点の半径のみによって決まります。

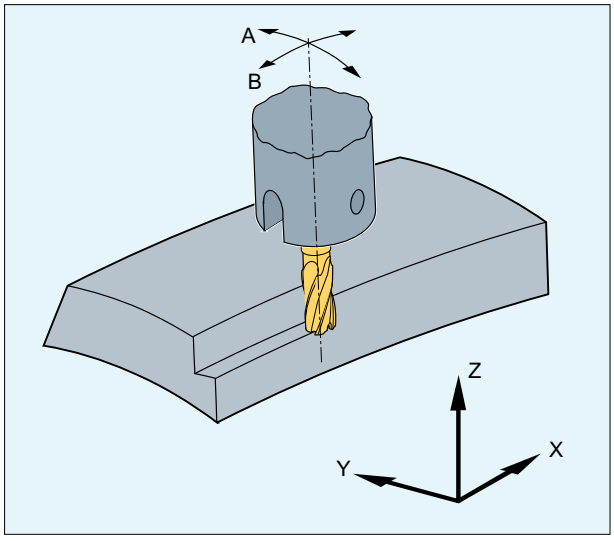


図 11-1　外周削り

アプローチ動作

アプローチ動作は常に 3 次元タイプの工具径補正の基本となります。

外側コーナーでの動作

グループ 18 の G 命令(コーナーでの動作、工具オフセット)は、3D TRC による外周加工に関して、2½D TRC の条件と同じ方法で使用されます。

- G450:挿入円(工具がワークのコーナーを、円弧軌跡を描いて回ります)
  2½D TRC のソリューションとは逆に、外側コーナーに挿入される輪郭要素は常に半径 0 の円弧であり、そこで工具径補正は他のプログラムで指令された経路と同じように動作します。円弧の代わりに円錐を挿入することはできません。この場合、DISC アドレスは重要でないため、使用されません。
- G451:等間隔の軌跡の交点(工具がワークのコーナーから後退します)
  交点は、関連する 2 ブロックのオフセット曲線を延長し、かつ工具オリエンテーションに垂直な平面上のコーナーで 2 ブロックの交点を定義して特定します。

交点の処理(G451)は、工具の向きの変更を含む少なくとも 1 つのブロックが、関連する移動ブロックの間に挿入された場合は使用されません。この場合、円弧は常にコーナーに挿入されます。

外側コーナーでの向きの変更の動作

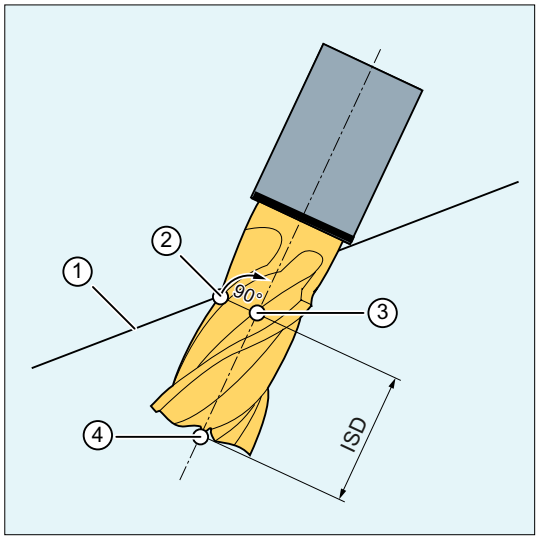
ORIC と ORID G 命令を使用して、コーナーを形成する 2 つのブロックとの間にプログラム指令した向きの変更を、挿入円弧ブロックの前に処理する(ORID)か、円弧ブロックと同時に処理する(ORIC)かを特定します。

切り込み深さ

フライス工具の切り込み深さとは、工具の先端からフライス工具のレファレンス点までの距離です。

フライス工具のレファレンス点は、プログラムされた軌跡上のフライス工具加工点を、工具の長手軸に垂直に投影した点です。

工具の円筒面の加工点位置は、切り込み深さとともに設定されます。



① プログラム指令軌跡
② フライス工具の加工点
③ フライス工具基準点
④ フライス工具の先端
ISD 切り込み深さ

図 11-2 切り込み深さ

標準工具とは違う工具を基準にした工具径補正

標準工具とは違う工具を基準にした外周加工用の 3D TRC は、CUT3DCD 命令によって選択されます。この工具径補正は、プログラムされた輪郭が標準工具の中心点軌跡を基準にしていて、標準工具とは違う工具以外の工具が加工に使用されている場合に適用します。3 次元工具径補正を計算する場合は、有効な工具の半径の摩耗値($TC_DP15)とプ

ログラムされている工具オフセット OFFN および TOFFR のみが考慮されます。動作中の工具の基本半径($TC_DP6)は **考慮されません**。

CUT3DC を使用した、外周削りの傾斜側壁のポケット加工

この 3 次元工具径補正では、加工する表面の法線方向の切り込みにより、フライス半径の差を補正します。フライス工具面が位置する平面は、挿入深さ ISD がそのまま同じ場合は、そのまま変更されません。たとえば、標準工具より半径の小さいフライス工具は、限界面でもあるポケットの底には届きません。自動工具切り込みの場合は、この限界面を制御装置が認識している必要があります。「3 次元外周加工、限界面の考慮(CUT3DCC、CUT3DCCD) (ページ 517)」の章を参照してください。

Advanced Surface / Top Surface

注記

工具径補正 CUT3DCD を「Advanced Surface」または「Top Surface」オプション(ライセンスが必要)と組み合わせて適用する場合、「Advanced Surface」/「Top Surface」の推奨設定を遵守してください。

設定データの確認用の特別なテストプログラムが SIOS ポータルに設けられています。

- Advanced Surface (https://support.industry.siemens.com/cs/ww/en/view/78956392)のテストプログラム
- Top Surface (https://support.industry.siemens.com/cs/ww/en/view/109738423)のテストプログラム

11.5.2 3 次元正面削りのための 3 次元工具径補正の選択(CUT3DF、CUT3DFS、CUT3DFF、CUT3DFD)

3 次元正面削りのための 3 次元工具径補正(3D TRC)は、モーダルで有効な G 命令 CUT3DF、CUT3DFS、CUT3DFF または CUT3DFD によって選択されます。

実際の有効化は、G41 または G42 を使用して実施されます。

3 次元正面削りでは、工具径補正を計算するため、加工面の面法線を定義する必要があります。これは G41 または G42 によって、A4、B4、C4 および A5、B5、C5 アドレスを介して行う必要があります。

工具径補正は G40 によって無効化されます。

構文

```
G41/G42 ORIC/ORID CUT3DF/CUT3DFS/CUT3DFF/CUT3DFD
X...Y...Z...A4=...B4=...C4=...A5=...B5=...C5=...
```

```
...
G40 X...Y...Z...
```

意味

CUT3DFS:	向きが一定の正面削りの 3D TRC。工具オリエンテーションは、G17 ～ G19 で特定され、フレームには影響されません。	
CUT3DFF:	向きが一定の正面削りの 3D TRC。工具オリエンテーションは、G17 ～ G19 で定義した方向で、場合によっては、フレームによって回転します。	
CUT3DF:	向きが変わる正面削りの 3 次元工具補正(動作中の 5 軸座標変換の場合のみ)。	
CUT3DFD:	標準工具とは違う工具を基準にした向きが変わる正面削りの 3 次元工具補正(動作中の 5 軸座標変換の場合のみ)。 半径差は工具パラメータ$TC_DP15 によって指定されます。 **注記**: CUT3DFD は「3 次元正面削りの面法線のスムーズ化」と組み合わせた場合のみ可能です。これは「Top Surface」機能(ライセンス要)を CYCLE832(...) (ページ 951)から呼び出すことによって有効になります。	
G41/G42 X...Y...Z...:	工具オフセットの有効化 3 次元正面削りでは G41 と G42 による動作は同じです。 **注記:** 有効化は線形ブロック(G0/G1)で実施する必要があります。	
A4/5=...B4/5=...C4/5=...:	加工面の面法線の定義	
	A4=...B4=...C4=...:	ブロック始点での定義
	A5=...B5=...C5=...:	ブロック終点での定義

ORIC/ORID:	外側コーナーでの向きの変更の動作が、G 命令 ORIC および ORID によって指定されます。	
	ORIC:	外側コーナーでの向きの変更を、挿入する円弧ブロックに重畳します。
	ORID:	円弧ブロックの前に向きの変更を実行します
G40 X...Y...Z...:	工具径補正の無効化 **注記:** 無効化は線形ブロック(G0/G1)でジオメトリ軸の移動によって実施する必要があります。	

注記

G41 または G42 は、補正に関連するジオメトリ軸の移動動作なしで、ブロックでプログラム指令することもできます。この場合、アプローチブロックはそのようなブロックに続く最初の移動ブロックとなります。
有効な工具径補正による 3D TRC タイプの変更は、アラームなしで無視されます。

例

例 1:CUT3DF による 3 次元正面削り

プログラムコード	コメント
N10	; 工具 D1 の定義:
N20 $TC_DP1[1,1]=121	; 工具タイプ (トーラス形フライス工具)
N30 $TC_DP3[1,1]= 20	; 長さ補正
N40 $TC_DP6[1,1]=5	; 半径
N50 $TC_DP7[1,1]=3	; 丸み付け半径
N60	
N70	
N80 X0 Y0 Z0 A0 B0 C0 G17 T1 D1 F12000	; 工具の選択。
N90 TRAORI(1)	; 方向座標変換を選択。
N100 B4=-1 C4=1	; 平面の定義。
N110 **G41** ORID **CUT3DF** G64 X10 Y0 Z0	; 工具オフセットの有効化。
N120 X30	
N130 Y20 A4=1 C4=1	; 外側コーナー、新しい平面定義。
N140 B3=1 C3=5	; ORID による向きの変更。
N150 B3=1 C3=1	; ORID による向きの変更。
N160 X-10 A5=1 C5=2 ORIC	
N170 A3=-2 C3=1	; ORIC による向きの変更。
N180 A3=-1 C3=1	; ORIC による向きの変更。

プログラムコード	コメント
N190 Y-10 A4=-1 C4=3	; 新しい平面定義。
N200 X-20 Y-20 Z10	; 前のブロックがある内側コーナー。
N210 X-30 Y10 A4=1 C4=1	; 内側コーナー、新しい平面定義。
N220 A3=1 B3=0.5 C3=1.7	; ORIC による向きの変更。
N230 X-20 Y30 A4=1 B4=-2 C4=3 ORID	
N240 A3 = 0.5 B3=-0.5 C3=1	; 向きの変更。
N250 X0 Y30 C4=1	; 経路の動き、新しい平面、
	; 相当するプログラミングによる向き。
N260 BSPLINE X20 Z15	; スプライン開始、向きの相当するプログラミング
N270 X30 Y25 Z18	; スプライン中も有効であり続ける。
N280 X40 Y20 Z13	
N290 X45 Y0 PW=2 Z8	
N300 Y-20	
N310 G2 ORIMKS A30 B45 I-20 X25 Y-40 Z0	; ヘリカル、軸プログラミングによる向き。
N320 G1 X0 A3=-0.123 B3=0.456 C3=2.789 B4=-1 C4=5 B5=-1 C5=2	; 経路動作、向き、一定でない平面。
N330 X-20 G40	; 工具径補正を解除。
N340 M30	

例 2 :CUT3DFD によって CAD システムから生成された NC プログラム(セクション)

プログラムコード	コメント
N01 G710	
N03 T="12"	
N06 S5305 M03	
N07 G642	
; 工具長さを考慮しながら、MCS の起点にアプローチします。	
G00 G90 X-250.62787 Y-38.37944 A=DC(253.12719) B-12.49543	
G00 G90 Z251.80052	
; MCS での位置決めの終了。	
;	
TRAORI(1)	; 方向座標変換を選択。
G500	
D1	
CYCLE832(0.01, _TOP_SURFACE_SMOOTH_ON + _ORI_FINISH, 1)	; CYCLE832 を呼び出し:
	; 輪郭許容範囲 = 0.01 mm、
	; 処理タイプ:スムージングを行う最上面、
	; 旋回の許容範囲の入力による仕上げ
	; 向きの許容範囲 = 1 度
CUT3DFD	
N08 G90 G94	
N09 G00 X-269.21195 Y128.32027 Z1.18577 A3=-.216361688 B3=.934284397 C3=-.283373051	; ブロック N09 ～ N10 は、一定の向きのワークへのアプローチ動作の高速部位です。

プログラムコード	コメント
N10 G00 X-251.90301 Y53.57752 Z23.85561	
N11 G01 X-247.57578 Y34.89183 Z29.52308 F50000.00000	; ブロック N11 ～ N21 では、アプローチ動作の低速部位が生成されます。工具はすでにワークの近くにあり、さらに、CYCLE832 からの金型加工設定 (COMPSURF など) が有効になっています (有効な G01 による)。金型加工動作のための、このいわゆる過渡段階の経路は、輪郭許容範囲の約 1000 倍としてください (この例では 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	
N22 **G41** X-248.06237 Y30.32400 Z9.36695 A5=-.060431854 B5=.973045457 C5=-.222554556 A3=-.214974689 B3=.929398552 C3=-.300007025 F1061.03295	; N22 から、面法線は初めてブロック全体で完全に定義されます (つまり、前のブロック N21 の終点に面法線が存在し、そのため、ブロック N22 の始点での面法線と、ブロック N22 の終点での面法線も存在します)。したがって、G41/G42 による工具オフセットの切り替えのための前提条件が満たされました。
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	
...	

注記

CUT3DFD による 3 次元正面削りでは、G41/G42 による工具オフセットを有効化するため、面法線の定義が必要です。面法線を定義せずに G41/G42 をプログラム指令すると、アラームが出力されます。

詳細情報

3 次元正面削り

このタイプの 3 次元加工の場合は、ワーク表面の 3 次元軌跡を行単位で記述してください。通常は CAM で実行される計算では、工具の形状と寸法が考慮されます。ポストプロセッサが、NC ブロックの他に、工具の向き(動作中の 5 軸座標変換用)、および必要な 3 次元工具補正の G 命令を、パートプログラムに書き込みます。これは、機械のオペレータが、NC 軌跡の計算に使用する工具とは違った、少し小さい工具を使用することができることを意味します。

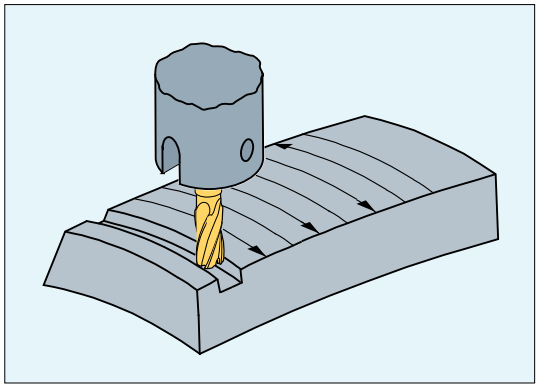


図 11-3　正面削り

アプローチ動作

アプローチ動作は常に 3 次元タイプの工具径補正の基本となります。

外側コーナーでの動作

外側コーナーは正面削りのための半径ゼロの円弧として扱われます。それにより、円弧面は最初のブロックの終端タンジェントから 2 番目のブロックの開始タンジェントまで延びます。このようにして、ブロック移行中に向きを変えることができます。そのため、円弧は常に外側コーナーに輪郭要素として挿入されます。正面削りでは交点手順はありません。

外側コーナーでの向きの変更の動作

ORIC と ORID G 命令を使用して、コーナーを形成する 2 つのブロックとの間にプログラム指令した向きの変更を、挿入円弧ブロックの前に処理する(ORID)か、円弧ブロックと同時に処理する(ORIC)かを特定します。

標準工具とは違う工具を基準にした工具径補正

標準工具とは違う工具を基準にした 3 次元工具径補正は、CUT3DFD 命令を使用して選択します。この工具径補正は、プログラムされた輪郭が標準工具の中心点軌跡を基準にしていて、標準工具とは違う工具以外の工具が加工に使用されている場合に適用します。

3 次元工具径補正を計算する場合は、有効な工具の半径の摩耗値($TC_DP15)とプログラムされている工具オフセット OFFN および TOFFR のみが考慮されます。動作中の工具の基本半径($TC_DP6)は **考慮されません**。

CUT3DFD による 3 次元正面削りは、「3 次元正面削りの面法線のスムーズ化」と組み合わせた場合のみ可能です。これは「Top Surface」機能(ライセンス要)を CYCLE832(...) (ページ 951) から呼び出すことによって有効になります。有効化は、G41/G42 によって工具オフセットを有効にする **前に** 実施する必要があります。ただし、工具の接触点の直前ではなく、その経路長さ 1 つ前に実施します。これは輪郭許容範囲の約 1000 倍に対応します(例えば 1000 x 0.01 mm = 10 mm)。無効化は逆に順番で行ってください。まず G40 によって工具オフセットをオフにし、その後輪郭許容範囲の約 1000 倍に対応する経路長さの後で、CUT2D (または類似のもの)によって無効化します。

「3 次元正面削りの面法線のスムーズ化」を使用できるようにするには、「多項式による面法線の補間」機能も有効にする必要があります。

MD28291 $MC_MM_SMOOTH_SURFACE_NORMALS = TRUE

注記

CUT3DFD による「Top Surface」と組み合わせた 3 次元正面削りでは、「Top Surface」に関連する推奨設定を遵守してください。
設定データの確認用の特別なテストプログラムが SIOS ポータルに 設けられています (https://support.industry.siemens.com/cs/ww/en/view/109738423)。

11.5.3 3 次元外周加工、限界面の考慮(CUT3DCC、CUT3DCCD)

工具オリエンテーションが連続して、または絶えず変化する 3 次元外周削りでは、多くの場合、定義した標準工具に対して工具中心点軌跡をプログラム指令します。実際には、適切な標準工具を使用できない場合が多いため、標準工具との差が小さい工具(≤ 5%)を使用できます。

CUT3DCCD は、プログラム指令した標準工具の記述とは違う実際の工具の限界面を考慮します。NC プログラムは、標準工具の中心点軌跡を記述します。

円筒工具を使用した場合の CUT3DCC では、プログラム指令した標準工具が到達する限界面が考慮されます。NC プログラムは、加工面の輪郭を記述します。

限界面の面法線ベクトルは、3 次元正面削りの A4、B4、C4 および A5、B5、C5 によって指定されます。

構文

```
G41/G42 CUT3DCCD/CUT3DCC CDOF2 X...Y...Z...A4=...B4=...C4=...A5=...B5=...C5=...
...
G40 X...Y...Z...
```

意味

CUT3DCCD:	工具中心点軌跡で標準工具とは違う工具を使用した、限界面を考慮した外周加工のための 3D TRC:限界面への切り込み	
CUT3DCC:	3 次元径補正を使用した、限界面を考慮した外周加工のための 3D TRC:加工面の輪郭	
G41/G42 X...Y...Z...:	工具径補正の有効化	
	G41:	輪郭の左側の工具径補正
	G42:	輪郭の右側の工具径補正
	注記: 有効化は線形ブロック(G0/G1)で実施する必要があります。	
CDOF2:	3 次元外周加工用の 衝突検出 の無効化	
A4/5=...B4/5=...C4/5=...:	限界面の面法線の定義	
	A4=...B4=...C4=...:	ブロック始点での定義
	A5=...B5=...C5=...:	ブロック終点での定義
G40 X...Y...Z...:	工具径補正の無効化 **注記:** 無効化は線形ブロック(G0/G1)でジオメトリ軸の移動によって実施する必要があります。	

注記

3D TRC を選択するための G 命令はアプローチブロックで、つまり通常は G41 または G42 を含むブロックで使用されます。

G41 または G42 は、補正に関連するジオメトリ軸の移動動作なしで、ブロックでプログラム指令することもできます。この場合、アプローチブロックはそのようなブロックに続く最初の移動ブロックとなります。

有効な工具径補正による 3D TRC タイプの変更は、アラームなしで無視されます。

例

プログラムコード	コメント
N10 $TC_DP1[1,1]=120	; 円筒フライス工具
N20 $TC_DP6[1,1]=10	
N30 $TC_DP15[1,1]=-3	
...	
; 円筒フライス工具と CUT3DCCD による処理	
N110 TRAORI	; 座標変換の起動。
N120 A4=0 B4=0 C4=1	; ブロック始点での限界面の面法線の定義。
N130 X0 Y0 Z0 A0 C0 T1 D1 F20000	
N140 X10 Y0 Z0 G41 **CUT3DCCD** CDOF2 G64	; 限界面を考慮して、衝突検出をオフにした 3 次元外周加工の有効化。
N150 X20	
N160 X30 A45	; 鈍角 ==> 切り込みなし
N170 X40 A-45	; 鋭角 ==> 切り込みあり
N180 X55	
N190 Y10 Z10	; 工具方向の移動。
N200 Y20	
N210 C45	; 向きの純粋な変更。
N220 Y30 C90	
N230 A5=-1 B5=0 C5=2 Y40	; 表面の変更。
N240 Y50 G40	; 工具径補正を解除。
...	

詳細情報

工具タイプ

工具タイプ(工具パラメータ$TC_DP1)を使用します。シャンクが円筒形のタイプのフライス工具(円筒またはエンドミル、トーラス形フライス工具、限られた状況においては、ボールエンドミル)のみを使用できます。これは番号 111 および 155 ～ 157 を除く工具タイプ 1 ～ 399 に対応します。

コーナ R ありの標準工具

標準工具によるコーナ R は、工具パラメータ$TC_DP7 で定義します。工具パラメータ $TC_DP16 は、標準工具との比較による実際の工具のコーナ R の誤差を表わします。

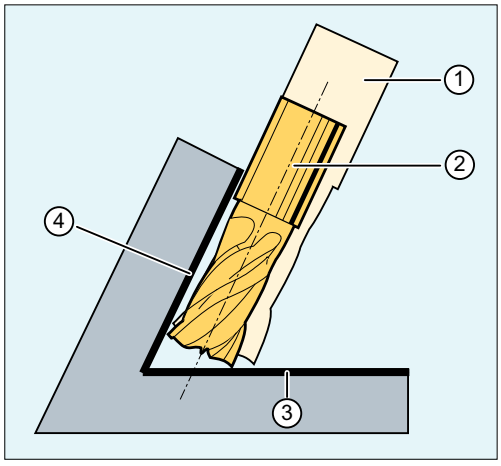
例:標準工具に比べて半径の小さいトーラス形フライス工具

工具タイプ	軸半径(R)	コーナ半径(r)
コーナ R のある標準工具	R = $TC_DP6	r = $TC_DP7
コーナの丸み付けのある実際の工具 工具タイプ 121 と 131 のトーラス形フライス工具 (コーナの丸み付けのあるエンドミル)	R' = $TC_DP6 + $TC_DP15 + OFFN	r' = $TC_DP7 + $TC_DP16

この例では、$TC_DP15 + OFFN と$TC_DP16 はいずれも負です。

CUT3DCCD による 3D TRC:限界面までの切り込みによる工具中心点軌跡

該当する標準工具より半径の小さい工具を使用した場合は、長手方向に切り込むフライス工具を使用して、ポケットの底(ベース)に達するまで加工を続行します。工具は、加工面と限界面が成すコーナから、できるだけ切り屑を取り除きます。これは、外周削りと正面削りを組み合わせた加工タイプでおこなわれます。半径が小さい工具と同じように、半径の大きな工具の場合は、逆方向に切り込みます。



① 標準工具
② 半径が小さい工具での限界面までの切り込み
③ 限界面
④ 加工面

G グループ 22 の他のすべての工具補正とは逆に、CUT3DCCD に指定した工具パラメータ$TC_DP6 は、工具半径には関係がなく、補正結果には影響しません。補正オフセットは、工具半径の摩耗値(工具パラメータ$TC_DP15)と、限界面に垂直なオフセットを計算するためにプログラム指令された工具オフセット OFFN の合計から得られます。

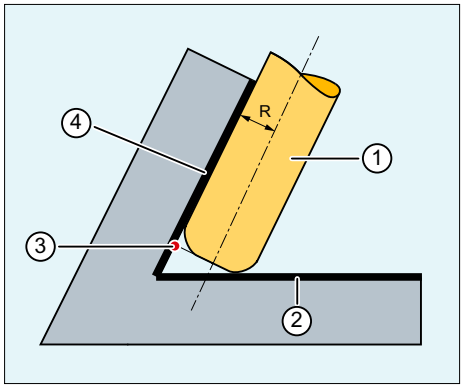
生成したパートプログラムは、加工する表面が軌跡の右側か、左側かを指定しません。したがって、半径が正の値で、オリジナル工具の摩耗値が負の値であることを前提とします。負の摩耗値は常に、より小さい直径の工具を表わします。

円筒工具の使用

円筒工具の使用時には、加工面と限界面が鋭角(90°未満)を成す場合にのみ、切り込みが必要です。トーラス形フライス工具(コーナR付きのエンドミル)を使用する場合は、鋭角と鈍角の両方の場合に、長手方向の工具の切り込みが必要です。

CUT3DCC による 3D TRC:加工面の輪郭

トーラス形フライス工具による CUT3DCC が有効な場合は、プログラム指令軌跡は、同じ直径を持つ仮想の円筒フライス工具を基準とします。得られる軌跡基準点は、次のトーラス形フライス工具の図に示されています。



① トーラス形フライス工具
② 限界面
③ 経路基準点
④ 加工面
R シャフト半径(工具半径)

加工面と限界面が成す角度は、同じブロック内で鋭角から鈍角に、またはその逆に変化する場合があります。

実際には、標準工具より大きな工具も、小さな工具も使用できます。ただし、結果としては、コーナ半径は負ではなく、工具半径の符号はそのまま保持してください。

CUT3DCC の場合は、NC パートプログラムは加工面の輪郭を基準とします。従来の工具径補正と同じように、以下で構成される工具半径の合計を使用します。

- 工具半径(工具パラメータ$TC_DP6)
- 摩耗値(工具パラメータ$TC_DP15)
- 限界面に垂直なオフセットを計算するためにプログラム指令された工具オフセット OFFN

限界面の位置は、次の差分から定義されます。

標準工具の寸法 - 工具半径(工具パラメータ$TC_DP6)

Advanced Surface / Top Surface

注記

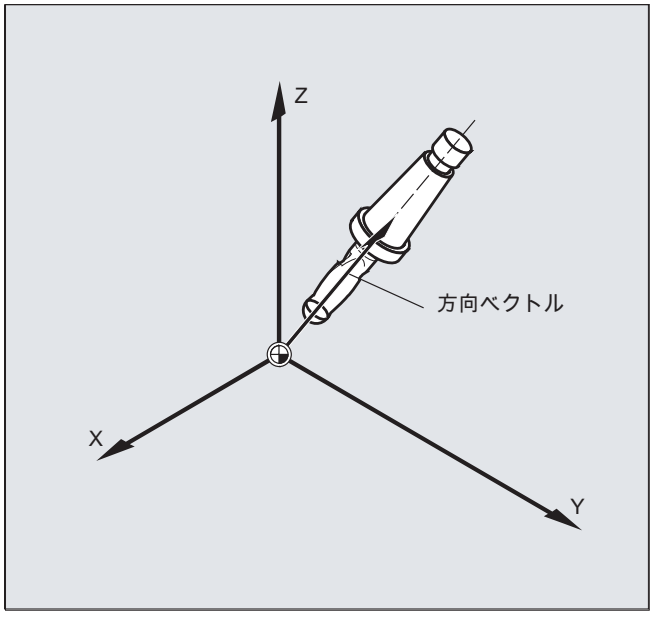
工具径補正 CUT3DCC / CUT3DCCD を「Advanced Surface」または「Top Surface」機能(ライセンスが必要)と組み合わせて適用する場合、「Advanced Surface」/「Top Surface」の推奨設定を遵守してください。

設定データの確認用の特別なテストプログラムが SIOS ポータルに設けられています。

- Advanced Surface (https://support.industry.siemens.com/cs/ww/en/view/78956392)のテストプログラム
- Top Surface (https://support.industry.siemens.com/cs/ww/en/view/109738423)のテストプログラム

11.6 工具オリエンテーション(ORIC、ORID、OSOF、OSC、OSS、OSSE、ORIS、OSD、OST)

工具オリエンテーションという用語は、空間の工具の幾何学的配置を表わします。5 軸工作機械の工具オリエンテーションは、プログラム命令を使用して設定できます。



OSD と OST で起動する向きの回転移動は、工具オリエンテーションの補間タイプに対応して、それぞれ実行されます。

ベクトル補間が有効な場合は、スムージングした旋回処理も、ベクトル補間を使用して補間されます。また、回転軸補間が有効な場合は、旋回は、回転軸移動を使用して直接スムージングされます。

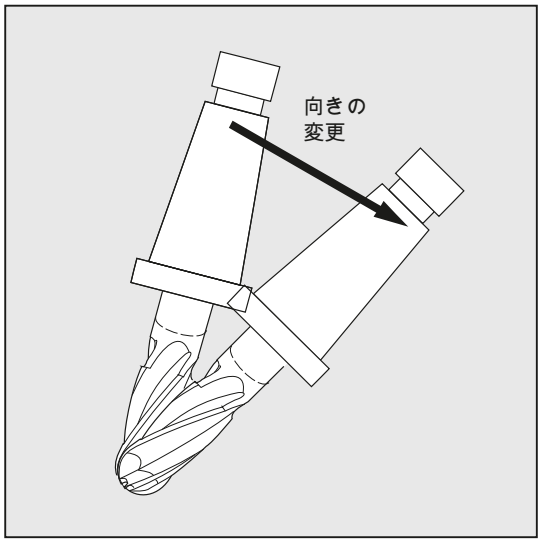
プログラミング

向きの変更のプログラミング:

工具オリエンテーションの変更は、次のようにプログラム指令できます。

- 回転軸 A、B、C の直接プログラミング(回転軸補間)
- オイラー角または RPY 角
- 方向ベクトル(A3 または B3 または C3 の指定によるベクトル補間)
- LEAD/TILT (正面削り)

基準となる座標系は、機械座標系(ORIMKS)または現在のワーク座標系(ORIWKS)です。



工具オリエンテーションのプログラミング:

ORIC:	向きの移動と軌跡移動の並行動作
ORID:	向きの移動と軌跡移動の連続動作
OSOF:	スムージングなしの旋回
OSC:	連続旋回のスムージング
OSS:	ブロックの開始時だけの旋回のスムージング
OSSE:	ブロックの開始時と終了時の旋回のスムージング
ORIS:	°/mm の単位で起動される旋回のスムージングによる向きの変更の速度(OSS と OSSE で有効)
OSD:	次のセッティングデータでスムージング距離が指定された旋回のスムージング: SD42674 $SC_ORI_SMOOTH_DIST
OST:	次のセッティングデータでベクトル補間の許容角度範囲(° 単位)が指定された旋回のスムージング: SD42676 $SC_ORI_SMOOTH_TOL 回転軸補間の場合は、指定した許容範囲が、旋回軸の最大変動範囲であることが前提となります。

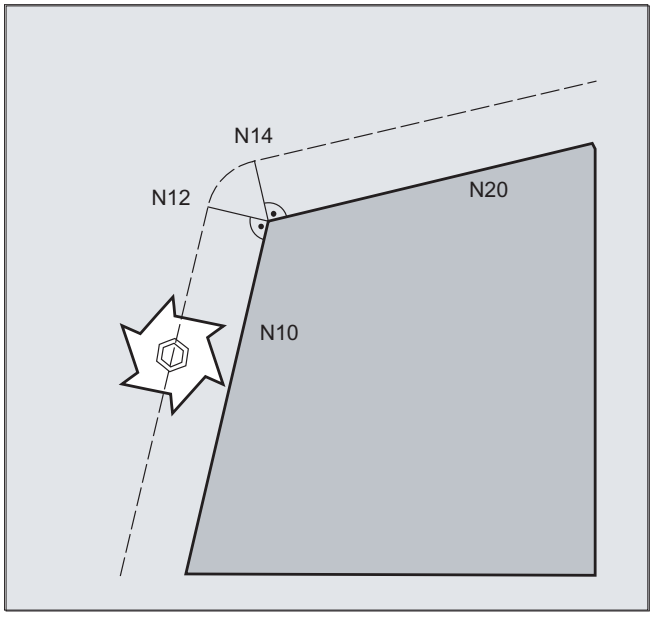
注記

工具オリエンテーションのスムージングのすべての命令(OSOF、OSC、OSS、OSSE、OSD、および OST)は、G グループ 34 にまとめられています。これらの命令はモーダルです。つまり、同時に有効となるのは、その中の 1 つだけです。

例

例 1:ORIC

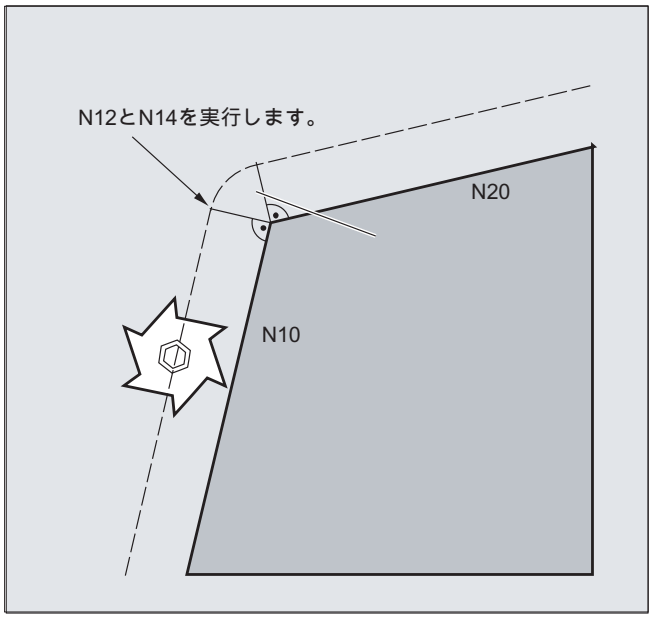
オリエンテーションの変更がある 2 つ以上のブロックが、プログラムされている移動ブロック N10 と N20 の間にプログラムされている場合(例: A2=...B2=...C2=...)、かつ ORIC が有効な場合、アブソリュート角度の変更に従って、挿入円弧ブロックがこれらの中間ブロックの間に配分されます。



プログラムコード	コメント
ORIC	
N8 A2=… B2=… C2=…	
N10 X… Y… Z…	
N12 C2=… B2=…	; 外側コーナの挿入円弧ブロックは、向きの変更に応じて、N12 と N14 の間に配分されます。円弧移動と向きの変更は、並行して実行されます。
N14 C2=… B2=…	
N20 X =…Y=… Z=… G1 F200	

例 2 :ORID

ORID が有効な場合は、2 個の移動ブロック間のすべてのブロックが、1 番目の移動ブロックの終わりに実行されます。一定の向きをもった円弧ブロックは、2 番目の移動ブロックの直前で実行されます。



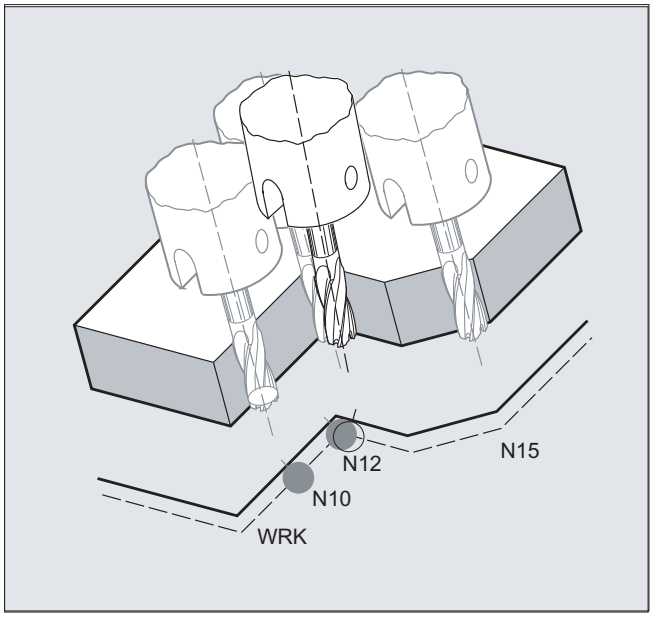
プログラムコード	コメント
ORID	
N8 A2=… B2=… C2=…	
N10 X… Y… Z…	
N12 A2=… B2=… C2=…	; N12 と N14 ブロックは、N10 の終わりで実行されます。その後、円弧ブロックが実際の向きで実行されます。
N14 M20	; 補助機能、など
N20 X… Y… Z…	

注記

外側の輪郭で向きの変更をおこなう方法は、外側コーナの最初の移動ブロックで有効なプログラム命令で特定されます。

向きが変更されない場合:ブロック境界で向きが変更されない場合は、工具の断面が円弧で、両方の輪郭に接しています。

例 3 :内側コーナでの向きの変更



プログラムコード

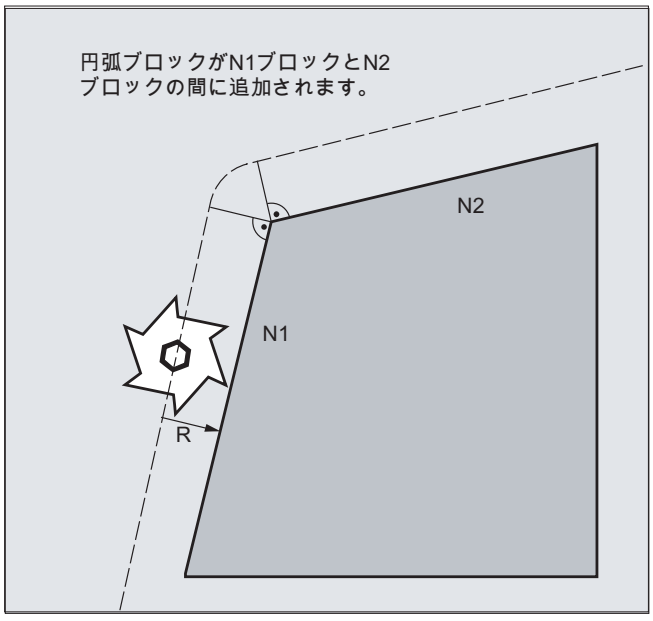
```
ORIC
N10 X …Y… Z… G1 F500
N12 X …Y… Z… A2=… B2=… C2=…
N15 X …Y… Z… A2=… B2=… C2=…
```

詳細情報

外側コーナでの動作

カッターの半径による円弧ブロックは常に、外側コーナで挿入されます。

ORIC と ORID プログラム命令を使用して、ブロック N1 と N2 の間にプログラム指令した向きの変更を、挿入円弧ブロックの前に処理するか、円弧ブロックと同時に処理するかを特定します。



向きを外側コーナで変更する必要がある場合は、補間と同時に実行するか、または補間とは別に、軌跡移動時に実行することができます。

ORID をプログラム指令すると、挿入ブロックは、最初に軌跡移動なしブロックで実行します。コーナを生成する円弧ブロックは、2 個の移動ブロックのうちの 2 番目の直前に挿入されます。

複数の旋回ブロックを外部コーナで挿入し、ORIC を選択している場合は、向きの変更の絶対値に従って、挿入した個々のブロック間に円弧移動が配分されます。

OSD または OST による旋回のスムージング

G642 によるスムージング時には、輪郭軸と旋回軸の最大変動範囲が大きく異ならないようにしてください。2 つの許容範囲のうちの小さい方により、スムージング移動のタイプと許容角度範囲のいずれか、または両方が特定され、旋回処理のスムージングが比較的強めにおこなわれるため、許容輪郭誤差がより小さくなります。

OSD と OST を有効にすると、指定したスムージング距離と許容角度範囲を使用して、旋回処理によるわずかな誤差を、大きな輪郭誤差を生じることなく、「十分に」スムージングできます。

注記

OSD と OST のいずれか、または両方で旋回を丸み付けするときは、G642 による輪郭の丸み付け処理(および旋回処理)とは異なり、別のブロックを形成することなく、丸み付け移動を直接、プログラム指令したオリジナルのブロックに追加します。

OSD と OST のいずれか、または両方を使用しても、工具オリエンテーション補間のタイプを変更する(ベクトル→回転軸、回転軸→ベクトル)場合、ブロック遷移部は丸み付けできません。これらのブロック遷移部は、必要に応じて、標準丸み付け機能 G641、G642 と G643 で丸み付けできます。

11.7 D 番号の任意割り当て、刃先番号

11.7.1 D 番号、刃先番号の任意割り当て(CE アドレス)

D 番号

D 番号は、輪郭番号として使用できます。 また、CE アドレスで刃先番号をアドレス指定することもできます。 システム変数$TC_DPCE を使用して、刃先番号を表わすことができます。

初期設定: 補正番号==工具刃先番号

マシンデータを使用して、最大 D 番号(刃先番号)、および工具毎の刃先の最大数を定義します(→ 工作機械メーカ)。 後述の命令は、指定した最大刃先番号(MD18105)が、工具毎の刃先(MD18106)の最大数より大きい場合のみ使用できます。 工作機械メーカの仕様書を参照してください。

注記

相対 D 番号割り当ての他に、T 番号(「フラット D 番号構成」機能のなかで)を基準とせずに、「フラット」または「絶対」D 番号(1 ～ 32000)として、D 番号を割り当てることができます。

参照先

総合機能説明書 基本機能; 工具オフセット(W1)

11.7.2 D 番号の任意割り当て: D 番号のチェック(CHKDNO)

CKKDNO 命令を使用すると、既存の D 番号が一義的に割り当てられているかどうかを確認できます。TO ユニット内で定義されたすべての工具の D 番号は、2 回以上出現してはなりません。予備工具は考慮されません。

構文

```
state=CHKDNO(Tno1,Tno2,Dno)
```

意味

state:	= TRUE:	D 番号は確認範囲に、一義的に割り当てられています。
	= FALSE:	D 番号が競合しているか、またはパラメータが無効です。Tno1、Tno2、および Dno は、競合をおこすパラメータを返します。これらのデータをパートプログラムで使用できます。
CHKDNO(Tno1,Tno2):	指定した範囲のすべての D 番号を確認します。	
CHKDNO(Tno1):	Tno1 のすべての D 番号を、他のすべての工具に対して確認します。	
CHKDNO:	すべての工具のすべての D 番号を、他のすべての工具に対して確認します。	

11.7.3 D 番号の任意割り当て: D 番号の名称変更(GETDNO、SETDNO)

D 番号は一義的に割り当ててください。1 つの工具の 2 個の異なる刃先が同じ D 番号にならないようにしてください。

GETDNO

この命令は、工具番号 t の工具の特定の刃先(ce)の D 番号を返します。入力したパラメータに対して D 番号が存在しない場合は、d=0 が設定されます。D 番号が無効な場合は、32000 を超える値が返されます。

SETDNO

この命令は、D 番号の値 d を工具 t の刃先(ce)に割り当てます。この命令の結果は state (TRUE または FALSE)で返されます。指定したパラメータのデータブロックが存在しない場合は、値 FALSE を返します。構文に誤りがあると、アラームが発生します。D 番号は、明示的には 0 に設定できません。

構文

```
d = GETDNO (t,ce)

state= SETDNO (t,ce,d)
```

意味

D:	工具刃先の D 番号
T :	工具の T 番号
ce:	工具の刃先番号(CE 番号)
state:	命令が実行できたかどうか(TRUE または FALSE)を表わします。

D 番号の名称変更の例

```
プログラミング                                    コメント
$TC_DP2[1.2]=120
$TC_DP3[1,2] = 5.5
$TC_DPCE[1,2] = 3                                 ; 刃先番号 CE
...
N10 def int DNoOld, DNoNew = 17
N20 DNoOld = GETDNO(1,3)
N30 SETDNO(1,3,DNoNew)
```

これで、新しい D 番号 17 が刃先 CE=3 に割り当てられます。この刃先のデータが、NC アドレス D を使用したプログラミングとシステム変数の両方で、D 番号 17 によってアドレス指定されます。

11.7.4 D 番号の任意割り当て: 指定した D 番号に対して T 番号の特定(GETACTTD)

絶対 D 番号に関連する T 番号は、GETACTTD 命令で特定します。一義的かどうかはチェックされません。工具オフセットユニット内に同じ D 番号が複数ある場合は、検索で見つかった最初の工具の T 番号を返します。この命令は、「フラット」D 番号には対応していません。フラット D 番号の場合は、返される値が常に「1」になります(データベースに T がありません)。

構文

```
status=GETACTTD(Tnr,Dnr)
```

意味

Dnr:	検索する T 番号の D 番号です。
Tnr:	見つかった T 番号です。

status:	値:	意味
	0	T 番号が見つかりました。Tno には T 番号の値が含まれます。
	-1	指定した D 番号の T 番号は存在しないため、Tno=0 です。
	-2	D 番号が絶対 D 番号ではありません。Tno は、値 Dno の D 番号を含む、最初に見つかった工具の値を受け取ります。
	-5	上記以外のなんらかの理由により、この機能を実行できません。

11.7.5 D 番号の任意割り当て: D 番号の無効化(DZERO)

DZERO 命令を使用して、段取り替えのときのサポートをおこないます。この命令でマークされた補正データセットは、CHKDNO 命令でチエックされなくなります。SETDNO を再度使用して D 番号を設定すると、このようなデータセットに再びアクセスできます。

構文

```
DZERO
```

意味

DZERO:	TO ユニットのすべての D 番号に、無効の印を付けます。

11.8 工具ホルダキネマティクス

必要条件

工具ホルダは以下の場合にのみ、空間で考えられるすべての方向に工具を旋回できます。

- 2 つの回転軸 v_1 および v_2 が存在する。
- 回転軸が互いに直交している。
- 工具縦軸が 2 番目の回転軸 v_2 に直交している。

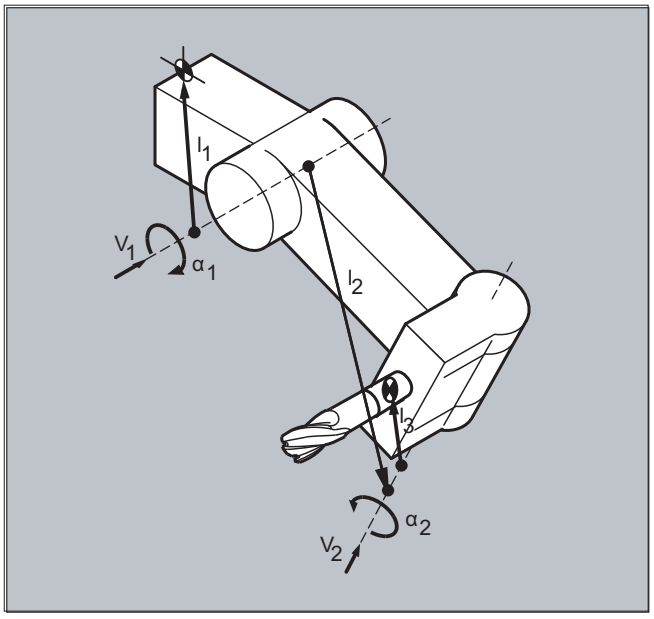
さらに、考えられるすべての旋回が設定可能な機械に対しては、以下の必要条件が適用されます。

- 工具長手軸が 1 番目の回転軸 v_1 に直交している。

機能

最大 2 つの回転軸 v_1 または v_2 による工具ホルダのキネマティックスは、17 個のシステム変数$TC_CARR1[m]～$TC_CARR17[m]を使用して定義します。工具ホルダの記述には、以下が含まれます。

- 工具ホルダの 1 番目の回転軸 I_1 からのベクトル距離、1 番目の回転軸から 2 番目の回転軸 I_2 までのベクトル距離、2 番目の回転軸から工具の基準点までのベクトル距離 I_3。
- 両方の回転軸 V_1、 V_2 の方向ベクトル。
- 2 つの軸を中心とした回転角度 α1、α2。回転角度は、回転軸ベクトルの視野方向の、右回転方向が正の角度です。



分解キネマティックス(工具と部品の両方を回転できる)を備えた機械では、システム変数が$TC_CARR18[m]～$TC_CARR23[m]で拡張されています。

パラメータ

旋回工具ホルダのシステム変数の機能			
名称	x 成分	y 成分	z 成分
I_1 オフセットベクトル	$TC_CARR1[m]	$TC_CARR2[m]	$TC_CARR3[m]
I_2 オフセットベクトル	$TC_CARR4[m]	$TC_CARR5[m]	$TC_CARR6[m]

旋回工具ホルダのシステム変数の機能			
v_1 回転軸	$TC_CARR7[m]	$TC_CARR8[m]	$TC_CARR9[m]
v_2 回転軸	$TC_CARR10[m]	$TC_CARR11[m]	$TC_CARR12[m]
α_1 回転角度 α_2 回転角度	$TC_CARR13[m] $TC_CARR14[m]		
l_3 オフセットベクトル	$TC_CARR15[m]	$TC_CARR16[m]	$TC_CARR17[m]

旋回工具ホルダのシステム変数の拡張			
名称	**x 成分**	**y 成分**	**z 成分**
l_4 オフセットベクトル	$TC_CARR18[m]	$TC_CARR19[m]	$TC_CARR20[m]
軸識別子回転軸 v_1 回転軸 v_2	回転軸 v_1 と v_2 の軸識別子(ゼロで初期化) $TC_CARR21[m] $TC_CARR22[m]		
キネマティックスタイプ **T**ool **P**art **M**ixed モード	$TC_CARR23[m]		
	キネマティックスタイプ T ->	キネマティックスタイプ P ->	キネマティックスタイプ M
	工具のみが回転可能(初期設定)	部品のみが回転可能	部品と工具が回転可能
回転軸 v_1 回転軸 v_2 の**オフセット**	回転軸 v_1 と v_2 の角度(° 単位)(初期設定 $TC_CARR24[m] $TC_CARR25[m]を前提とする)		
回転軸 v_1 回転軸 v_2 の**角度オフセット**	回転軸 v_1 と v_2 のカップリングギヤシステムのオフセット(° 単位) $TC_CARR26[m] $TC_CARR27[m]		
角度増分 v_1 回転軸 v_2 回転軸	回転軸 v_1 と v_2 のカップリングギヤシステムのオフセット(° 単位) $TC_CARR28[m] $TC_CARR29[m]		
最小位置回転軸 v_1 回転軸 v_2	回転軸 v_1 と v_2 のソフトウェア リミットの最小位置$TC_CARR30[m] $TC_CARR31[m]		

旋回工具ホルダのシステム変数の拡張			
最大位置回転軸 v_1 回転軸 v_2	回転軸 v_1 と v_2 のソフトウェアリミットの最大位置$TC_CARR32[m] $TC_CARR33[m]		
工具ホルダ名称	工具ホルダには、番号の代わりに名称を付けられます。 $TC_CARR34[m]		
ユーザー: 軸名称 1 軸名称 2 識別子 **位置**	ユーザーの計測サイクルの用途で使用$TC_CARR35[m] $TC_CARR36[m] $TC_CARR37[m]		
	$TC_CARR38[m]	$TC_CARR39[m]	$TC_CARR40[m]
仕上げオフセット	**基本パラメータ**の**値に**加算可能なパラメータ。		
l_1 オフセットベクトル	$TC_CARR41[m]	$TC_CARR42[m]	$TC_CARR43[m]
l_2 オフセットベクトル	$TC_CARR44[m]	$TC_CARR45[m]	$TC_CARR46[m]
l_3 オフセットベクトル	$TC_CARR55[m]	$TC_CARR56[m]	$TC_CARR57[m]
l_4 オフセットベクトル	$TC_CARR58[m]	$TC_CARR59[m]	$TC_CARR60[m]
v_1 回転軸	$TC_CARR64[m]		
v_2 回転軸	$TC_CARR65[m]		

注記

パラメータの意味

「m」は、プログラム指令する工具ホルダの番号を指定します。

$TC_CARR47 ～$TC_CARR54 および$TC_CARR61 ～$TC_CARR63 は定義されません。また、読み取りや書き込みをおこなった場合、アラームが発生します。

軸の距離ベクトルの始点/終点は自由に選択できます。2 つのそれぞれの軸を中心とした回転角度 α_1、 α_2 は、工具ホルダの初期状態を 0°として定義されます。この方法を使用すると、工具ホルダのキネマティックスを、任意の数だけプログラム指令できます。

回転軸が 1 つのみ、またはまったくない工具ホルダは、回転軸のいずれかまたは両方の方向ベクトルをゼロに設定すると記述できます。
回転軸を装備しない工具ホルダでは、距離ベクトルが、加工平面(G17 ～ G19)の変化に影響されない成分を持つ追加の工具オフセットとして機能します。

拡張パラメータ

回転軸のパラメータ

システム変数は、$TC_CARR24[m]～$TC_CARR33[m]項目により拡張されており、次のように記述されます。

回転軸 v_1、v_2 の **オフセット**	旋回工具ホルダの初期設定のために、回転軸 v_1 または v_2 の位置を変更。
回転軸 v_1、v_2 の **角度オフセット/角度増分**	回転軸 v_1 と v_2 のカップリングギヤシステムのオフセットまたは角度増分。プログラム指令した、または計算した角度は、phi = s + n * d (n は整数)で得られる値の整数値に切り上げられます。
回転軸 v_1、v_2 の **最小位置と最大位置**	回転軸 v1 と v2 の回転軸の制限角度(ソフトウェアリミット)の最小位置と最大位置。

ユーザー用パラメータ

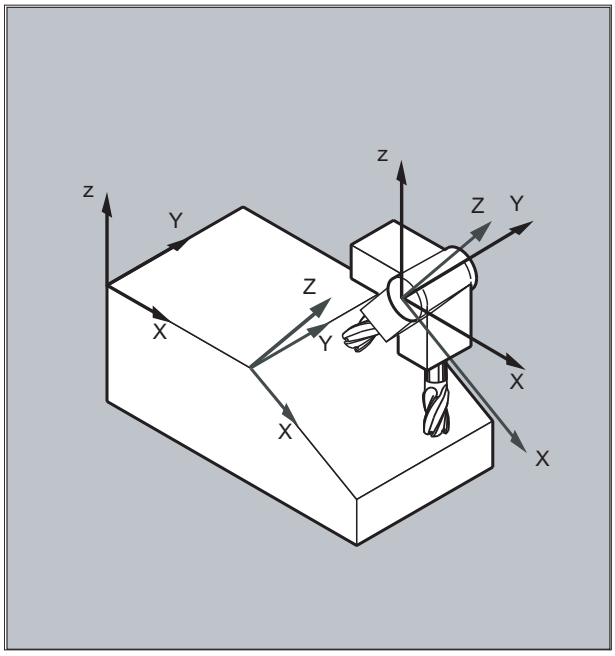
$TC_CARR34 ～$TC_CARR40 には、ユーザーが自由に使用できるパラメータが含まれます。また、これらのパラメータは、SW 6.4 以前では、標準では、NCK 内で使用できない、または意味をもちません。

仕上げオフセットパラメータ

$TC_CARR41 ～$TC_CARR65 には、基本パラメータの値に加算可能な仕上げオフセットパラメータが含まれます。基本パラメータに割り当てる仕上げオフセット値は、40 の値をパラメータ番号に加算すると得られます。

例

次の例で使用する工具ホルダは、Y 軸を中心とした回転で全体を記述できます。



プログラムコード	コメント
N10 $TC_CARR8[1]=1	; 工具ホルダ 1 の 1 番目の回転軸の Y 成分の定義です。
N20 $TC_DP1[1,1] = 120	; エンドミルの定義です。
N30 $TC_DP3[1,1]= 20	; 20 mm 長さのエンドミルの定義です。
N40 $TC_DP6[1,1]=5	; 5 mm 半径のエンドミルの定義です。
N50 ROT Y37	; Y 軸を中心とした 37°の回転によるフレーム定義です。
N60 X0 Y0 Z0 F10000	; 開始位置へアプローチします。
N70 G42 CUT2DF TCOFR TCARR=1 T1 D1 X10	回転フレームで径補正と工具長補正を設定して、工具ホルダ 1、工具 1 を選択します。
N80 X40	; 37°の回転で加工を実行します。
N90 Y40	
N100 X0	
N110 Y0	
N120 M30	

詳細情報

分解キネマティックス

分解キネマティックス(工具とワークの両方が回転する)を備えた機械では、システム変数が$TC_CARR18[m]～$TC_CARR23[m]項目で拡張されており、次のように記述されます。

以下を備えた旋回工具テーブル:

- 3 番目の回転軸で回転できる工具テーブルの基準点 I_4 に対する 2 番目の回転軸 V_2 のベクトル距離。

以下を備えた回転軸:

- 回転軸 V_1 と V_2 の参照用の 2 個のチャネル識別子(その位置は、必要に応じて、旋回工具ホルダの向きの特定に使用します)。

以下のいずれかのキネマティックスタイプ値:

- キネマティックスタイプ T:工具のみが回転可能です。
- キネマティックスタイプ P:部品のみが回転可能です。
- キネマティックスタイプ M:工具と部品が回転可能です。

工具ホルダデータの消去

すべての工具ホルダデータセットのデータは、$TC_CARR1[0]=0 を使用して削除できます。

キネマティックスタイプ$TC_CARR23[T]=T には、使用可能な 3 種類の文字(T、P、M)の大文字または小文字のいずれかを割り当ててください。このため、これは削除しないでください。

工具ホルダデータの変更

記述した値はいずれも、パートプログラムで新しい値を割り当てて変更できます。T、P、および M 以外の文字を割り当てると、旋回工具ホルダを起動しようとしたときにアラームが発生します。

工具ホルダデータの読み取り

記述した値はいずれも、パートプログラムでそれを変数に割り当てると、読み取ることができます。

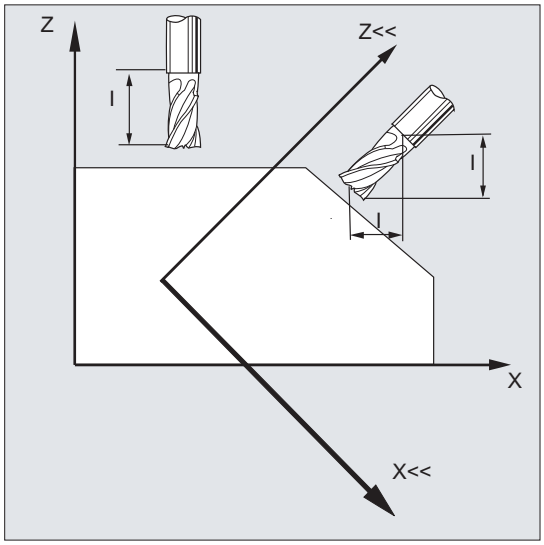
仕上げオフセット

不適切な仕上げオフセット値が検出されるのは、旋回工具ホルダが有効なときに、旋回工具ホルダがこの値を含み、同時にセッティングデータが
SD42974 $SC_TOCARR_FINE_CORRECTION = TRUE の場合のみです。

最大許容仕上げオフセットは、マシンデータの許容値に制限されます。

11.9 旋回工具ホルダの工具長補正(TCARR、TCOABS、TCOFR、TCOFRX、TCOFRY、TCOFRZ)

空間の工具の向きが変化すると、その工具長成分も変化します。



手動設定または、空間の向きが固定された工具ホルダの交換などにより、リセットされた後には、工具長成分も再度特定してください。これは、TCOABS と TCOFR の軌跡命令を使用して実行します。

動作中のフレームの旋回工具ホルダの場合は、TCOFRZ、TCOFRY、および TCOFRX で工具を選択時に、工具が向く方向を定義できます。

構文

```
TCARR=[<m>]
TCOABS
TCOFR
TCOFRZ
TCOFRY
TCOFRX
```

意味

TCARR=[<m>]:	番号「m」の工具ホルダを要求します
TCOABS:	現在の工具ホルダの向きから工具長成分を特定します。
TCOFR:	動作中のフレームの向きから工具長成分を特定します。
TCOFRZ:	動作中のフレームの旋回工具ホルダの Z 方向の工具点です。

TCOFRY:	動作中のフレームの旋回工具ホルダの Y 方向の工具点です。
TCOFRX:	動作中のフレームの旋回工具ホルダの X 方向の工具点です。

詳細情報

工具ホルダの向きによる工具長補正の特定(TCOABS)

TCOABS で、工具ホルダの現在の旋回角度から工具長補正を計算し、システム変数 $TC_CARR13 and $TC_CARR14 に保存します。

システム変数による工具ホルダのキネマティックスの定義については、「工具ホルダのキネマティックス (ページ 534)」を参照してください。

フレームの変更時には、工具長補正を新しく計算するために、工具を再度選択してください。

動作中のフレームの工具方向

旋回工具ホルダは、工具が以下の方向を向くように設定します。

- TCOFR または TCOFRZ の場合は Z 方向
- TCOFRY の場合は Y 方向
- TCOFRX の場合は X 方向

工具長補正は、TCOFR と TCOABS の間で変更があると、再計算されます。

工具ホルダ要求(TCARR)

TCARR を使用すると、工具ホルダ番号 m を、その形状データ(補正メモリ)で要求します。

m=0 の場合は、動作中の工具ホルダが解除されます。

工具ホルダの形状データは、工具の呼び出し後にのみ有効になります。選択した工具は、工具ホルダ交換をおこなった後もそのまま有効です。

工具ホルダの現在の形状データは、パートプログラムで対応するシステム変数を使用して定義することもできます。

フレーム変更時の工具長補正の再計算(TCOABS)

フレームの変更時には、工具長補正を新しく計算するために、工具を再度選択してください。

注記

工具オリエンテーションは、動作中のフレームに合わせて手動で調整してください。

工具長補正を計算するときは、工具ホルダの回転角度を中間のステップで計算します。2 つの回転軸を持つ工具ホルダの場合は通常、2 セットの回転角度があり、これらを使用すると、動作中のフレームに合わせて工具オリエンテーションを調整できます。したがって、システム変数に格納された回転角度の値は、少なくとも、機械的に設定した回転角度にほぼ対応している必要があります。

注記

工具の向き

制御装置では、フレームの向きを使用して計算した回転角度が機械に設定可能かどうかをチェックできません。

工具ホルダの回転軸が、フレームの向きを使用して計算した工具オリエンテーションに達することができない場合は、アラームが発生します。

精密工具補正と、可動式工具ホルダでの工具長補正の各機能を組み合わせることはできません。これらの両機能を同時に呼び出した場合は、エラーメッセージを出力します。

TOFRAME 機能を使用すると、選択した工具ホルダの旋回の方向に基づいて、フレームを定義できます。詳細情報については、「フレーム」の章を参照してください。

方向座標変換が有効(3 軸、4 軸、または 5 軸座標変換)な場合は、原点位置からずれている旋回工具ホルダを、アラームが発生することなく、選択できます。

標準サイクルと計測サイクルの転送パラメータ

標準サイクルと計測サイクルの転送パラメータの場合は、定義済みの以下の数値の範囲が適用されます。

角度の値の場合、数値の範囲の定義は次のとおりです。

- 1 番目のジオメトリ軸を中心とした回転: -180 ～+180°
- 2 番目のジオメトリ軸を中心とした回転: -90 ～+90°
- 3 番目のジオメトリ軸を中心とした回転: -180 ～+180°

フレームの章の「プログラマブル回転(ROT、AROT、RPL)」を参照してください。

注記

角度の値を標準サイクルまたは計測サイクルに渡すときは、以下のこと注意しておこなってください。

NC の計算最小単位より小さい値は切り捨てて、ゼロにしてください。

回転位置の NC の計算最小単位は、以下のマシンデータで定義してください。

MD10210 $MN_INT_INCR_PER_DEG

11.10 オンライン工具長補正(TOFFON、TOFFOF)

システム変数$AA_TOFF[<n>]を使用して、有効な工具長を 3 次元の 3 つの工具方向に応じて、リアルタイムで加算します。

3 個のジオメトリ軸識別子をインデックス<n>として使用します。これにより、有効な方向オフセットの数が、そのときに有効なジオメトリ軸で特定されます。

すべての補正を同時に有効にすることができます。

オンライン工具長補正機能を使用して、以下を実行できます。:

- 方向座標変換 TRAORI
- 旋回工具ホルダ TCARR

注記

オンライン工具長補正は**オプションです。**事前に有効にしておいてください。この機能は、動作中の方向座標変換または動作中の旋回工具ホルダと組み合わせた場合にのみ、有効です。

構文

```
TRAORI
TOFFON(<補正方向>[,<オフセット値>])
WHEN TRUE DO $AA_TOFF[<補正方向>]                    ; シンクロナイズドアクションの場合です。
...
TOFFOF(<補正方向>)
```

シンクロナイズドアクションのオンライン工具長補正のプログラミングについて詳しくは、「シンクロナイズドアクション (ページ 729)」を参照してください。

意味

TOFFON:	オンライン工具長補正の**起動**	
	<補正方向>:	オンライン工具長補正が有効になる工具方向 (X、Y、Z)
	<オフセット値>:	適用時に、当該の補正方向に対してオフセット値を指定でき、この値が直ちに有効になります。
TOFFOF:	オンライン工具長補正の**リセット** 指定した補正方向への補正値がリセットされ、先読み停止が開始されます。	

例

例 1:工具長補正の選択

プログラムコード	コメント
MD21190 $MC_TOFF_MODE = 1	; 絶対値へアプローチします。
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)	; 座標変換オン
N20 TOFFON(Z)	; Z の工具方向のオンライン工具長補正を有効にします。
N30 WHEN TRUE DO $AA_TOFF[Z]=10 G4 F5	; 10 の工具長補正を Z 工具方向に補間します。
...	
N100 XOFFSET=$AA_TOFF_VAL[X]	; 実際の補正を X 方向に割り当てます。
N120 TOFFON(X,-XOFFSET) G4 F5	; X の工具方向で、工具長補正が 0 に戻ります。

例 2 :工具長補正の解除

プログラムコード	コメント
N10 TRAORI(1)	; 座標変換オン
N20 TOFFON(X)	; X の工具方向のオンライン工具長補正を有効にします。
N30 WHEN TRUE DO $AA_TOFF[X] = 10 G4 F5	; 10 の工具長補正を X 工具方向に補間します。
...	

プログラムコード	コメント
N80 TOFFOF(X)	; X の工具方向の位置オフセットを ...$AA_TOFF[X]=0 で削除します。 軸は移動しません。 位置オフセットが、実際の向きに対応するワークの実位置に追加 されます。

詳細情報

ブロック解析

先読みのブロック解析のとき、メインランで有効な現在の工具長補正も考慮されます。最大許容軸速度を広範囲に使用できるようにするには、工具補正の設定中に STOPRE 先読み停止でブロック解析を中止してください。

プログラムスタート後に工具長補正を変更しない場合、または、工具長補正の変更後に、先読みとメインランの間に IPO バッファが受け入れ可能な数より多いブロックを処理していた場合は、開始時に常に工具補正は決定されています。

変数$AA_TOFF_PREP_DIFF

補間器で現在有効な補正と、ブロック解析時に有効であった補正との寸法差は、変数 $AA_TOFF_PREP_DIFF[<n>]でチェックできます。

マシンデータとセッティングデータの調整

オンライン工具長補正では、以下のシステムデータを使用できます。

- MD20610 $MC_ADD_MOVE_ACCEL_RESERVE (重畳移動の加速マージン)
- MD21190 $MC_TOFF_MODE
  システム変数の内容$AA_TOFF[<n>]が、絶対値として移動するか、または統合されます。
- MD21194 $MC_TOFF_VELO (オンライン工具長補正の速度)
- MD21196 $MC_TOFF_ACCEL (オンライン工具長補正の加速度)
- 事前設定の制限値のセッティングデータ
  :
  SD42970 $SC_TOFF_LIMIT (工具長補正の上限値)

参照先:
機能マニュアル、応用機能; F2:多軸座標変換

11.11 旋回工具のオフセットデータ変更

11.11.1 オリエンテーションの計算(ORISOLH)

事前定義された ORISOLH 機能は、ワークを基準に定義されたキネマティックとは無関係な位置に旋削工具を移動するために、ユーザーが機械の回転軸位置を設定するうえで役立ちます。前提条件は、キネマティック結合によりパラメータ設定された 6 軸変換が有効であることです。

次の 2 つの基本機能が使用可能です:

- 工具調整
  β 角と γ 角を指定します。この機能は、このために必要な 3 つの旋回軸の角度を計算します。
- 直接工具調整
  2 番目および 3 番目の旋回軸の角度を指定します。この機能は、関連する角度 β および γ と、欠如する第 1 の旋回軸を計算します。

注記

旋回軸の順序

ワークから工具への機械の構造を記述するキネマティック結合を実行する場合、6 軸変換の 3 旋回軸の順序で以下の指定が適用されます。

- **ワーク**に最も近い旋回軸が**第 1** の旋回軸です。
- **工具**に最も近い旋回軸が**第 3** の旋回軸です。

一般に、第 1 の旋回軸は主軸であり、したがってこの場合には対応する回転は回転フレームを通じて実装されます。

構文

```
<RetVal> = ORISOLH(<Cntrl>,<W1>,<W2>)
```

意味

ORISOLH:	ファンクションコール

<RetVal>:	機能の戻り値		
	データタイプ:	INT	
	値の範囲:	0, -2, -3, ..., -17	
	規格値:	0	機能は異常なく終了しました。
		-2	有効な変換(6 軸オリエンテーション変換)がありません。
		-3	第 1 のパラメータ(<Cntrl>)は負です。
		-4	第 1 のパラメータ(<Cntrl>)のユニットの位は無効です。 値 0 および 1 のみが指定可能です。
		-5	第 1 のパラメータ(<Cntrl>)の 10 の位が無効です。 値 0 ～ 3 のみが指定可能です。
		-6	第 1 のパラメータ(<Cntrl>)の 100 の位が無効です。 値 0 および 1 のみが指定可能です。
		-7	第 1 のパラメータ(<Cntrl>)の 1000 の位が無効です。 値 0 ～ 3 のみが指定可能です。
		-8	"直接工具調整"機能に対して角度 γ が大きすぎます。
		-9	指定した軸位置の少なくとも 1 つが、"直接工具調整"機能の軸の制限に違反しています。
		-10	工具が有効になっていません。
		-11	要求されたオリエンテーションが設定されていません。
		-12	ハースジョイントの自由な軸角度の調整は、第 1 の解または唯一の解では使用可能ではありません。
		-13	ハースジョイントの自由な軸角度の調整は、第 2 の解では使用可能ではありません。
		-14	ハースジョイントの自由な軸角度の調整は、2 つの解のいずれかで使用可能ではありません。
		-15	第 1 の旋回軸はハース軸としてパラメータ設定されています。
		-16	第 2 の回転軸と第 3 の回転軸がハース軸としてパラメータ設定されています。2 軸のうちのいずれかのみをハース軸にできます。

		-17	指定した軸位置の少なくとも 1 つが、"直接旋回"機能について、関連するハースジョイントとの互換性がありません。

<Cntrl>:	この機能の動作を制御します。		
	データタイプ:	INT	
	<Cntrl>パラメータは、10 進コードです(1000 の位に対するユニット):		
	ユニットの位:	ユニットの位は、異常に対する応答を制御します。	
		xxx0	異常発生時は(戻り値<0)、アラーム 14106 が出力され、プログラムの処理が中止します。 注記: また、<Cntrl>パラメータが負の場合、ユニットの位の値に関係なくアラームが出力されます。
		xxx1	異常発生時は(戻り値<0)、アラームは出力されません。ユーザーは、プログラムで適切に応答できます。
	10 の位:	ハースジョイントによる旋回軸が存在する場合に、動作を制御します。 注記: このパラメータは、"工具調整"機能について(つまり、100 の位が値"0"である場合に)のみ評価されます。	
		xx0x	軸の位置は、最も近い位置に丸み付けされます。
		xx1x	プログラムされた値に対する β 角度の相違が最小限になるように、軸位置が丸み付けされます。
		xx2x	β 角度が、プログラムされた値より小さい最大許容値に一致するように、軸の位置が丸み付けされます(β が切り下げられます)。
		xx3x	β 角度が、プログラムされた値より大きい最小許容値に一致するように、軸の位置が丸み付けされます(β が切り下げられます)。
	100 の位:	実行する機能、または以下の 2 つのパラメータ<W1>および<W2>の内容を指定します。	
		x0xx	"工具調整"機能 パラメータ<W1>および<W2>には以下の意味があります: • <W1> = β • <W2> = γ 旋回軸の関連する角度が計算されます。
		x1xx	"直接工具調整"機能

			<W1>は第 2 の旋回軸の位置指定、<W2>は 6 軸変換の 3 番目の旋回軸の位置指定です。2 つの位置指定と互換性のある第 1 の旋回軸の位置および β と γ の角度を定義します。 異常が発生しなかった場合、2 つの解は常に、$P_ORI_POS[<n>, <m>]システム変数に出力します。第 1 のインデックス(0 または 1)は解を参照し、第 2 のインデックス(0 ～ 2)は旋回軸を参照します: • $P_ORI_POS[0/1, 0]:第 1 の旋回軸の位置 • $P_ORI_POS[0/1, 1]:角度 β • $P_ORI_POS[0/1, 2]:角度 γ 位置指定<W1>および<W2>が任意のハースジョイントまたは有効なソフトウェア制限に対して互換性があるかどうかがチェックされます。互換性がない場合、対応する異常番号が返されます(<RetVal>パラメータを参照)。 角度<W1>および<W2>を任意に選択した場合、工具の刃先は一般に加工平面にありません。刃先が加工平面から回転する角度 γ は、セッティングデータ SD42999 $SC_ORISOLH_INCLINE_TOL によって定義される制限値以下であることが必要です。
	1000 の位:	100 の位が値"0"である場合、つまり"工具調整"機能では、変更可能な解の位置を指定します。	
		0xxx	計算される軸の位置は、現在の機械軸の位置に可能なかぎり近いことが必要です。
		1xxx	計算されるモジュロ軸の軸位置は、モジュロ範囲の中央に可能なかぎり近いことが必要です。また他の軸では、可能なかぎり 0 に近いことが必要です。非モジュロ軸では、これは軸位置が範囲-180 ～+180°に削減されることを示します。
		2xxx	計算される軸位置は、軸タイプに関係なく、範囲-180 ～+180°に削減されることが必要です。
<W1>:	第 1 の角度 <Cntrl>パラメータの 100 の位から結果が得られます。		
	データタイプ:	REAL	

<W1>:	第 2 の角度 <Cntrl>パラメータの 100 の位から結果が得られます。	
	データタイプ:	REAL

注記

プログラムされていないパラメータの初期値は"0"です。

詳細情報

検出された解の数と、ORISOLH 機能の実行時の詳細な状態情報は、以下のシステム変数で読み取ることができます:

システム変数	意味	
$P_ORI_POS [<n>, <m>]	オリエンテーションプログラミングから得られる旋回軸の角度を返します。	
	<n>:	解のインデックス
		値の範囲: 0, 1
	<m>:	旋回軸のインデックス
		値の範囲: 0 ... 2
		旋回軸(1 ～ 3)の順序は、$NT_ROT_AX_NAME の軸の定義に対応します。
	ORISOLH 機能の呼び出し時に、"直接工具調整"モード、$P_ORI_POS[0/1, 1]、および P_ORI_POS[0/1, 2]変数には、2 つの解に属する 2 の角度 β および γ の値が含まれます。 $P_ORI_POS[<n>, <m>]に入力された第 1 の解(つまりインデックス <n>=0)は常に、要求されたオリエンテーションに直接近づくときに制御によって選択される解です。第 2 のインデックス<m>は、旋回軸、つまり$NT_ROT_AX_NAME を参照します。 $P_ORI_POS[<n>, <m>]に入力された軸位置では、$NK_OFF および $NK_OFF_FINE に入力されたオフセットが考慮されます。つまり、これらの軸角度は、それ以上変更の必要のないオリエンテーションを設定するために以下のブロックで使用できます。 回転軸がハース軸である場合、解の位置はハースジョイントの最も近い休止位置に丸み付けされます。ハースジョイント回転軸では、正確な解の軸位置と、$P_ORI_DIFF システム変数のハース増分値に調整した解の軸位置間の差異を読み取ることができます。	

システム変数	意味		
$P_ORI_DIFF [<n>, <m>]	旋回軸の正確な位置と、オリエンテーションプログラミングから得られる$P_ORI_POS の位置の差異を返します。		
	<n>:	解のインデックス	
		値の範囲:	0, 1
	<m>:	旋回軸のインデックス	
		値の範囲:	0 ... 2
			旋回軸(1 ～ 3)の順序は、$NT_ROT_AX_NAME の軸の定義に対応します。
	位置がインクリメントされる場合(ハースジョイント)、つまり、関連する軸のシステムデータ$NT_HIRTH_INCR が 0 ではなく、この軸が手動の回転軸である場合にのみ、内容を 0 以外にできます。		

システム変数	意味		
$P_ORI_SOL	複数の旋回軸を含むオリエンテーション変換では、軸角度の計算で指定のオリエンテーションが得られるはずです。一般に、解は複数存在します。$P_ORI_SOL システム変数には、追加の状態情報が含まれる有効な解の番号が格納されます。 $P_ORI_SOL の内容は、以下のようにコード化されます:		
	値 < 0	一般的な異常状態	
		-1	有効な変換に対して、解がまだ計算されていません(ORISOLH が呼び出されていません)。
		-2	変換が有効ではないか、または有効な変換が指定のオリエンテーションプログラミングに対して位置を提供できるオリエンテーション変換(6 軸変換)ではありません。
		-4	適切なオリエンテーションは、現在発生しているキネマティックで設定できません。
		-5	"直接工具調整"モードで ORISOLH 機能が呼び出されたときに、解が見つかりませんでした。
		-6	"直接工具調整"モードでの ORISOLH 機能の呼び出しで、角度 γ が大きすぎます。
		-7	ハースジョイントのために設定できない"直接工具調整"モードで ORISOLH 機能が呼び出されたときに、角度が指定されました。
		-8	第 1 の旋回軸(フレーム軸)は、ハース軸としてパラメータ設定しないでください。
		-9	第 2 の回転軸と第 3 の回転軸がハース軸としてパラメータ設定されています。2 軸のうちのいずれかのみをハース軸にできます。
		-10	ハースジョイントの解の補正が見つかりませんでした。
	値 > 0 ユニットの位	軸の制限および異常条件を考慮せずに数学的に可能な解の数。	
		0	解はありません、つまり、要求されたオリエンテーションは設定できません。 この場合、3 つの異なる原因が考えられます: ● 原則として、要求されたオリエンテーションは、旋回軸の自由な移動範囲であっても、機械キネマ

システム変数	意味		
			ティック(直角に配置されていない旋回軸)のために実現できません。この場合、$P_ORI_SOL の 10 の位および 100 の位はともに 0 であり、旋回軸に割り当てられた$P_ORI_STAT 状態変数の値は"-4"です。 ● 軸の制限に違反するので、計算された解は実現できません。軸の制限なしで得られる旋回軸の位置は、$P_ORI_POS で読み取ることができます。 ● 位置が計算される第 1 の旋回軸に平行に位置合わせされる工具のオリエンテーションベクトルまたはオリエンテーション法線ベクトルのいずれかが得られる"直接工具調整"モードで ORISOLH 機能が呼び出されたときに、軸位置が指定されました。この場合、この軸の位置は定義されません。
		1	解が存在します。 この場合、3 つの異なる原因が考えられます: ● 指定のオリエンテーションおよび機械のキネマティックに基づいて、軸の制限を考慮しない場合でも 1 つの解のみが存在します(数学的な観点からは 2 つの一致する解)。この状況は、直角ではないキネマティックのオリエンテーション範囲の刃先で発生します。$P_ORI_POS には、両方の(同じ)解が含まれます。 ● 第 2 の解は軸の制限違反により無効であるために、1 つの解のみが存在します。有効な解は常に、$P_ORI_POS の第 1 の解です。軸の制限を考慮しない場合に得られる第 2 の解は、$P_ORI_POS で読み取ることができます。 ● ORISOLH 機能が"直接工具調整"モードで呼び出される場合、これは通常の状況です。2 つの旋回軸の指定された軸位置について、欠如する旋回軸に対して計算すべき 1 つの有効な位置のみが一般に存在します。
		2	2 つの解があります。
		8	無限数の解があります。つまり、旋回軸(極軸)の位置は任意です。ただし、他の軸の 2 つの可能な位置から、軸の制限違反のために 1 つは除外されます。
		9	無制限数の解があります。つまり、旋回軸(極軸)の位置は無限です。無制限の軸は、数百の位置または $P_ORI_STAT システム変数から決定できます。

システム変数	意味		
	値 > 0 10 の位:	違反する軸の制限のビット指定による表示。異常の正確な原因は、$P_ORI_STAT システム変数から特定することができます。	
		ビット 0 (値 10):	少なくとも 1 つの解について、第 1 の旋回軸の少なくとも 1 つの軸の制限に違反します。
		ビット 1 (値 20):	少なくとも 1 つの解について、第 2 の旋回軸の少なくとも 1 つの軸の制限に違反します。
		Bit 2 (値 40):	少なくとも 1 つの解について、第 3 の旋回軸の少なくとも 1 つの軸の制限に違反します。
	値 > 0 100 の位	未定義の軸位置のビット指定による表示(無限数の解がある場合、つまりユニット位置が"9"と同じである場合にのみ発生することがあります)。	
		ビット 0 (値 100):	第 1 の旋回軸の位置が定義されていません。
		ビット 1 (値 200):	第 2 の旋回軸の位置が定義されていません。
		ビット 2 (値 400):	第 3 の旋回軸の位置が定義されていません。
	第 1 旋回軸、第 2 旋回軸、第 3 旋回軸の名称は、$NT_ROT_AX_NAME の軸の定義を参照します。		

システム変数	意味		
$P_ORI_STAT [<n>]	ORISOLH の呼び出し後、最大 3 旋回軸のそれぞれの状態を返します。		
	<n>:	旋回軸のインデックス ($NT_ROT_AX_NAME の関連する旋回軸のインデックスに対応します)	
		値の範囲:	0 ... 2
			旋回軸(1 ～ 3)の順序は、$NT_ROT_AX_NAME の軸の定義に対応します。
	$P_ORI_STAT の内容は、以下のようにコード化されます:		
	値 < 0	一般的な異常状態	
		-1	状態が定義されていません(ORISOLH が呼び出されていません)。
		-2	変換が有効ではないか、または有効な変換が指定のオリエンテーションプログラミングに対して位置を提供できるオリエンテーション変換(6 軸変換)ではありません。
		-3	軸が有効な変換に含まれていません。
		-4	任意に想定される軸の移動範囲を持つ場合でも、要求されたオリエンテーションが現在のキネマティックにより実現できないために、軸の位置は計算できません。
		-5	位置が計算される第 1 の旋回軸に平行に位置合わせされる工具のオリエンテーションベクトルまたはオリエンテーション法線ベクトルのいずれかが得られる"直接工具調整"モードで ORISOLH 機能が呼び出されたときに、軸位置が指定されました。この場合、この軸の位置は定義されません。
		-6	"直接工具調整"モードでの ORISOLH 機能の呼び出しで、角度 γ が大きすぎます。
		-7	ハースジョイントのために設定できない"直接工具調整"モードで ORISOLH 機能が呼び出されたときに、角度が指定されました。
		-8	第 1 の旋回軸(フレーム軸)は、ハース軸としてパラメータ設定しないでください。

システム変数	意味		
		-9	第 2 の回転軸と第 3 の回転軸がハース軸としてパラメータ設定されています。2 軸のうちのいずれかのみをハース軸にできます。
		-10	ハースジョイントの解の補正が見つかりませんでした。
	値 > 0 ユニットの位	第 1 の解の軸の制限違反に対応するビット指定の表示。	
		ビット 0 (値 1):	第 1 の解が軸の下限に違反します。
		ビット 1 (値 2):	第 1 の解が軸の上限に違反します。
	値 > 0 10 の位:	第 2 の解の軸の制限違反に対応するビット指定の表示。	
		ビット 0 (値 10):	第 2 の解が軸の下限に違反します。
		ビット 1 (値 20):	第 2 の解が軸の上限に違反します。
	値 > 0 100 の位	未定義の軸位置の表示。	
		ビット 0 (値 100):	旋回軸の位置が定義されていません。つまり、要求されたオリエンテーションは、回転軸(極位置)の任意の各設定により実現します。この情報は、$P_ORI_SOL システム変数に含まれています。
	軸制限の違反を示す異常番号のうち、いくつかが同時発生する場合があります。軸の制限に違反する場合、360°の倍数を加算または減算することにより、許容可能な軸制限内の位置への到達が試みられます。これが可能でない場合、軸の下限または上限の制限に違反するかどうかは明らかには定義されません。 要求されたオリエンテーションに対応する解がない場合 ($P_ORI_SOL = 0)、変換時の旋回軸の状態は"0"です。		

注記

$NT_ROT_AX_NAME

このシステム変数は、オリエンテーションの設定のために最大 3 軸を参照します。キネマティックトランスフォーメーションから得られるオリエンテーション移動を実行する必要がある機械軸(回転軸)を定義する結合エレメント($NK_NAME)の名前が含まれます。これはキネマティック結合の構造から導出されるので、最大 3 つの回転軸がこのシステム変数に含まれる順序は、機械キネマティックには関連しません。ただし、これによって他の変数が回転軸にアクセスする順序が定義されるので、$NT_ROT_AX_NAME の旋回軸の順序はキネマティックの記述に一致する必要があります。

注記

ステータス情報

たとえば、オリエンテーションが実現できないか、関連する軸の制限に違反する場合にのみ実現できることを示す状態情報では、NC アラームはトリガされません。ユーザーが自身の責任で指定された条件に適切に応答する必要があります。

11.11.2 旋回工具のオフセットデータの変更の起動(CUTMOD、CUTMODK)

旋回工具のオフセットデータの変更は、CUTMOD(旋回工具ホルダと組み合わせた場合)または CUTMODK 言語命令(キネマティック結合によって定義されたオリエンテーション変換の場合)を通じて NC プログラムで起動されます。

注記

キネマティック結合によって定義された旋回工具ホルダおよびオリエンテーション変換は、同時には有効にできず、2 タイプ間で競合は発生しません。

構文

```
CUTMOD = <値>
```

または

```
CUTMODK = <命令>
```

意味

CUTMOD:	旋回工具ホルダと組合わせたファンクションコール

<値>:	割り当て値		
	データタイプ:	INT	
	値:	0	機能は解除されます。 システム変数$P_AD...で提供される値は、対応する工具パラメータと同じです。
		> 0	指定した番号の旋回工具ホルダが有効な場合、この機能が起動します。つまり、この起動は、指定した旋回工具ホルダにリンクされます。 システム変数$P_AD...で提供される値は、有効な回転に応じて、対応する工具パラメータに対して修正することができます。 指定の旋回工具ホルダを解除すると、一時的にこの機能が解除されます。別の旋回工具ホルダを起動すると、永続的にこの機能が解除されます。このことが、第 1 の状況で同じ旋回工具ホルダを再度選択したときに機能が再起動する理由です。第 2 の状況では、新たな選択が必要です。以降は、指定の番号の旋回工具ホルダが再起動されます。 この機能は、リセットの影響を受けません。
		-1	旋回工具ホルダが有効な場合、この機能は常に起動されます。 工具ホルダを変更する場合、または選択解除後に新たに選択する場合、再度 CUTMOD を設定する必要はありません。
		-2	現在有効な旋回工具ホルダと同じ番号の旋回工具ホルダが有効な場合、この機能は常に有効になります。 旋回工具ホルダが有効でない場合は、CUTMOD=0 と同じ意味になります。 旋回工具ホルダが有効な場合は、現在の工具ホルダ番号を直接指定する場合と同じ意味になります。
		< -2	2 未満の値は無視されます。すなわちこの場合は、CUTMOD がプログラム指令されなかったかのように処理されます。

			注: この数値の範囲は、この後の拡張の可能性のために予約されているので使用しないでください。
CUTMODK:	キネマティック結合によって定義されたオリエンテーション変換と組み合わせたファンクションコール		
<命令>:	割り当てられた命令		
	データタイプ:	STRING	
	値:	"NEW"	"オフセットデータの変更"に関連するキネマティック結合により定義された有効な変換の状態、変換の名前、および現在の輪郭フレームが保存されます。 **注:** この命令は、適切な変換(TRAORI_DYN、TRAORI_STAT、または TRAANG_K)が有効な場合にのみ許容されます。
		「OFF」	有効な"オフセットデータの変更"をオフに切り替えます。以前に"NEW"により格納されたデータが保持されます。 **注:** またこの命令は、CUTMODK が有効でない場合にのみ許容されます。そうでない場合は起動しません。"オフセットデータの変更"に対してすでに設定されているすべてのデータセットが保持されます。
		"ON"	この命令により、以前に"NEW"命令で保存されたデータセットで"オフセットデータの変更"が再起動されます。 この命令の実行時に、保存されているデータセット名の変換が有効である場合、"オフセットデータの変更"が直ちに有効になります。それ以外の場合は、有効な変換が起動されるまで、起動は遅延します。
		"CLEAR"	"OFF"命令の場合と同様に、"オフセットデータの変更"をオフに切り替え、保存されたデータセットも削除します。 **注:** またこの命令は、CUTMODK が有効でない場合にのみ許容されます。

注記

SD42984 $SC_CUTDIRMOD

CUTMOD/CUTMODK 命令では、セッティングデータ SD42984 $SC_CUTDIRMOD を使用して起動できる機能を置換します。ただし、このセッティングデータで起動した機能は、変更されずそのまま使用可能です。この両機能を並行して使用することは無意味なため、CUTMOD がゼロに等しい場合にのみ、セッティングデータで有効にできます。

詳細情報

変更されたオフセットデータの読み取り

変更されたオフセットデータは、以下のシステム変数および OPI 変数で提供されます:

意味	システム変数	OPI 変数
刃先位置	$P_AD[2]	cuttEdgeParam2
ホルダ角度	$P_AD[10]	cuttEdgeParam10
切削方向	$P_AD[11]	cuttEdgeParam11
逃げ角	$P_AD[24]	cuttEdgeParam24

"旋回工具のオフセットデータの変更"機能が CUTMOD または CUTMODK 命令により起動され、工具が旋回工具ホルダまたは適切なオリエンテーション変換によって回転したときに、データは常に、対応する工具パラメータ($TC_DP2[..., ...] など)に対して変更されます。

その他の機能に関連するシステム変数

システム変数	意味
$P_CUTMOD_ANG / $AC_CUTMOD_ANG	有効な加工平面、および CUTMOD 機能と CUTMODK 機能で使用可能な変更された刃先データの工具の回転角度を返します。
$P_CUTMOD / $AC_CUTMOD	CUTMOD 命令(オフセットデータの変更を起動する工具ホルダの番号)で最後にプログラムされた現在有効な値を読み取ります。 最後にプログラムされた値が CUTMOD = -2 (現在有効な旋回工具ホルダによる起動)である場合、システム変数に値"-2"は返されず、プログラミング時に有効な旋回工具ホルダの番号が返されます。

システム変数	意味
$P_CUTMODK / $AC_CUTMODK	"オフセットデータの変更"に対して現在有効なデータセットが作成された変換の名前を読み取ります。
$P_CUT_INV / $AC_CUT_INV	工具の回転によって主軸回転方向を反転しなければならない場合、値 TRUE が提供されます。これをおこなうには、読み取り操作の対象となるブロックで、以下の 4 つの条件を満たしてください。 1. 旋削工具または研削工具が有効です (工具タイプ 400 ～ 599 や SD42950 $SC_TOOL_LENGTH_TYPE = 2)。 2. オフセットデータの変更は、CUTMOD 命令または CUTMODK 命令で起動されました。 3. CUTMOD 命令または CUTMODK 命令で選択され、キネマティック結合により定義された旋回工具ホルダまたはオリエンテーション変換が有効です。 4. 工具刃先の得られる法線が 90°を超えて(一般的には 180°)初期位置を基準に回転するように、工具は旋回工具ホルダまたはキネマティックオリエンテーション変換によって回転します。 指定した 4 つの条件のうち少なくとも 1 つが満たされていない場合、この変数は値 FALSE を返します。刃先位置が定義されていない工具の場合、変数の値は常に FALSE です。

システム変数	意味	
$P_CUTMOD_ERR	CUTMOD 機能の前回の呼び出し後に異常状態 CUTMOD 機能も、工具交換のために自動的に呼び出すことができます。リセット時には、この変数はゼロにリセットされます。工具交換のたびにリセットされ、必要な場合には再度書き込まれます。 この変数はビット指定です。このビットの意味は以下のとおりです:	
	ビット 0:	有効な工具に対して利用できる切削方向が定義されていません。
	ビット 1:	有効な工具の刃先角度(逃げ角とホルダ角度)が、両方ともゼロです。
	ビット 2:	有効な工具の逃げ角が許容されない値(0°未満または 180°超)です。
	ビット 3:	有効な工具のホルダ角度が許容されない値(0°未満または 90°超)です。
	ビット 4:	有効な工具のプレート角度が許容されない値(0°未満または 90°超)です。
	ビット 5:	有効な工具の刃先位置 - ホルダ角度の組合わせが許容されません(刃先位置 1 ～ 4 のホルダ角度は ≤ 90 度、刃先位置 5 ～ 8 のホルダ角度 ≥ 90 度でなければならない)。
	ビット 6:	有効な工具の不正な回転。 工具は有効な加工平面から± 90°回転しました(許容値は約 1°)。したがって、刃先位置はもう加工平面では定義されません。
	ビット 7:	切削プレートが加工平面内になく、切削プレートと加工平面の角度がセッティングデータ SD42998 $SC_CUTMOD_PLANE_TOL によって指定された上限を超過しています。
	ビット 8:	切削プレートが加工平面内にありません。角度 α が 1°を超えています。角度 α は、角度 β の回転軸および角度 γ の回転軸(G18 の X 軸)と直交する座標軸を中心とした回転角です。

$P_ ...:先読み変数

$AC_...:メインラン変数

すべてのメインラン変数はシンクロナイズドアクションで読み取ることができます。先読みから読み取りアクセス操作をおこなうと、先読み停止が発生します。

平面変更

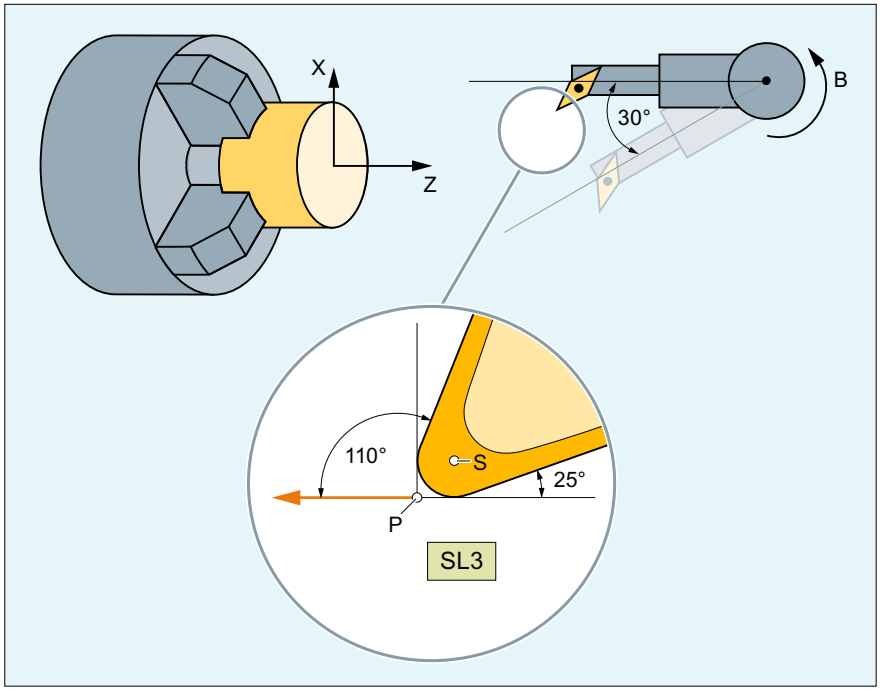
修正された刃先位置、切削方向および工具ホルダまたは逃げ角を求めるために、有効な平面(G17 ～ G19)の刃先の評価が重要です。

ただし、セッティングデータ SD42940 $SC_TOOL_LENGTH_CONST (平面選択時の工具長成分の変更)がゼロ以外の有効値(正または負の 17、18、または 19)の場合は、その内容で、当該の数量を使用する平面を定義します。

G コードに対するセッティングデータのこの優先規則は、マシンデータ $MC_TOOL_PARAMETER_DEF_MASK のビット 18 を設定することにより解除できます。これは、このビットが設定された場合に、グループ 6 の G 命令で定義された平面が依然として有効であることを示します。

修正された切削データの効果

修正された刃先位置と修正された刃先基準点は、すでに有効な工具に対しても、プログラミング時に直ちに有効になります。このために、工具を再選択する必要はありません。



S: 刃先の中心点

P: 刃先基準点

SL: 刃先位置

図 11-4 刃先位置 3 の工具と、B 軸を中心として工具を旋回できる旋回工具ホルダ。

プログラムコード	コメント
N10 $TC_DP1[1,1]=500	
N20 $TC_DP2[1,1]=3	;刃先位置
N30 $TC_DP3[1,1]=12	
N40 $TC_DP4[1,1]=1	
N50 $TC_DP6[1,1]=6	
N60 $TC_DP10[1,1]=110	; ホルダ角度
N70 $TC_DP11[1,1]=3	; 切削方向
N80 $TC_DP24[1,1]=25	; 逃げ角
N90 $TC_CARR7[2]=0 $TC_CARR8[2]=1 $TC_CARR9[2]=0	; B 軸
N100 $TC_CARR10[2]=0 $TC_CARR11[2]=0 $TC_CARR12[2]=1	; C 軸
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	

コメントの数値は、機械座標(MCS)でのブロック終点位置を、X → Y → Z の順で示しています。

説明

ブロック N180 では、最初に CUTMOD=0 および回転しない旋回工具ホルダの工具を選択します。旋回工具ホルダのすべてのオフセットベクトルが 0 であるため、$TC_DP3[1,1] と$TC_DP4[1,1]で指定した工具の長さに対応する位置へアプローチします。

B 軸を中心に 30°回転する旋回工具ホルダがブロック N200 で有効です。CUTMOD=0 により刃先位置は変化しないため、以前と同様に既存の刃先基準点で決まります。このため、ブロック N210 では、位置へのアプローチにより、以前の刃先基準点がゼロのまま保持されます(つまり、ベクトル(1, 12)が Z/X 平面で 30°回転します)。

ブロック N260 では、ブロック N200 とは逆に、CUTMOD=2 が有効です。旋回工具ホルダの回転の結果、刃先位置は 8 に変更されます。またこれによって、軸位置が逸脱します。

工具径補正(TRC)は、ブロック N220 と N270 の両方またはいずれかで有効です。両方のプログラム区間で刃先位置が異なっていても、TRC が有効なブロックの終了位置には影響しません。したがって、対応する位置は同じです。異なった刃先位置は、解除ブロック N260 と N300 の両方またはいずれかでのみ、再度有効になります。

11.12 工具環境の使用

機能の概要

- 工具環境の保存(TOOLENV) (ページ 571)
- 工具環境の削除(DELTOOLENV) (ページ 574)
- T 番号、D 番号、および DL 番号の読み出し (GETTENV) (ページ 575)
- 工具長、工具長成分、またはその両方の読み取り(GETTCOR) (ページ 577)
- 工具成分の変更(SETTCOR) (ページ 585)

システム変数の概要

- 保存した工具環境に関する情報の読み取り($P_TOOLENVN、$P_TOOLENV) (ページ 577)

11.12.1 工具環境の保存(TOOLENV)

TOOLENV 機能を使用すれば、メモリに格納されている工具データの評価に必要な現在の状態を保存できます。

個々のデータは、次のとおりです。

- 次のグループで有効な G 命令:
  - 6 (G17, G18, G19)
  - 56 (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)
- 有効な径方向軸
- マシンデータ:
  - MD18112 $MN_MM_KIND_OF_SUMCORR (TO 領域内のサムオフセットのプロパティ)
  - MD20360 $MC_TOOL_PARAMETER_DEF_MASK (工具パラメータの定義)

- セッティングデータ:
  - SD42900 $SC_MIRROR_TOOL_LENGTH (ミラーリング時の工具長符号変更)
  - SD42910 $SC_MIRROR_TOOL_WEAR (ミラーリング時の工具摩耗の符号変更)
  - SD42920 $SC_WEAR_SIGN_CUTPOS (刃先位置のある工具の摩耗の符号)
  - SD42930 $SC_WEAR_SIGN (摩耗の符号)
  - SD42935 $SC_WEAR_TRANSFORM (工具成分の座標変換)
  - SD42940 $SC_LENGTH_CONST (平面の変更による工具長成分の変更)
  - SD42942 $SC_TOOL_LENGTH_CONST_T (平面の変更による工具長成分の変更)
  - SD42950 $SC_TOOL_LENGTH_TYPE (工具タイプと無関係な工具長成分の割り当て)
  - SD42954 $SC_TOOL_ORI_CONST_M (平面の変更によるフライス工具の工具方向成分の変更)
  - SD42956 $SC_TOOL_ORI_CONST_T (平面の変更による旋削工具の工具方向成分の変更)
- 現在のすべてのフレームの向きコンポーネント(回転とミラーリング、作業オフセットまたはスケーリングなし)
- 向きコンポーネント、および有効な旋回工具ホルダの結果として生じた長さ
- 向きコンポーネント、および有効な座標変換の結果として生じた長さ

工具の環境を記述するデータに加えて、有効な工具の T 番号、D 番号、および DL 番号も格納されます。この結果、工具を再度指定しなくても、TOOLENV 呼び出しと同じ環境で、後ほどこの工具にアクセスできます。

構文

```
<状態> = TOOLENV(<名前>)
```

意味

TOOLENV(...) :	工具環境を保存するための定義済み機能	
	単独ブロック指令:	あり

<状態>:	機能の戻り値。負の値は異常状態を示します。		
	データタイプ:		INT
	値:	0	動作 OK
		-1	工具環境に予約されたメモリがありません。 MD18116 $MN_MM_NUM_TOOL_ENV = 0 これは、「工具環境」機能が使用できないことを示します。
		-2	工具環境に対応した空きメモリ場所がこれ以上ありません。
		-3	工具環境名として空文字列は不適切です。
		-4	パラメータ(<名前>)が指定されていません。
パラメータ			
1 <名称>:	現在のデータセットを保存するときに使用する名前。 同じ名前のデータセットがすでにある場合、このデータセットは上書きされます。この場合、Status は 0 になります。		
	データタイプ:		STRING

他の情報

基本寸法/アダプタ寸法の工具長補正

工具マガジン管理が有効な場合(「工具管理」オプションのみで使用可能)、次のマシンデータの値で、アダプタ長さまたは工具ベース寸法(刃先別パラメータ$TC_DP21、$TC_DP22、および$TC_DP23)を工具長の計算に組み込むかどうかを定義します。

MD18104 $MN_MM_NUM_TOOL_ADAPTER (TO 領域内の工具アダプタ)

このマシンデータに加えた変更が効果を現すのはコントロールシステムの電源オン後に限るため、この変更は工具環境に保存されません。

旋回工具ホルダと座標変換の結果として生じる長さ:

注記

方向機能および座標変換を含む両工具ホルダでは、工具の長さの追加成分として、実行される回転に部分的にまたは完全に適用できるシステム変数またはマシンデータを使用できます。得られた工具長さの追加成分も、工具を使用する環境の一部を表すために、TOOLENV の呼び出し時に保存する必要があります。

アダプタ座標変換

アダプタ座標変換は、工具アダプタのプロパティであるため、すべての工具のプロパティでもあります。このため、アダプタ座標変換は、工具環境の一部ではなく、別の工具に適用できます。

工具の全長を特定するのに必要な全データを保存すれば、工具が無効になっても、または環境の条件(たとえば、G コードまたはセッティングデータ)が変更されていても、これより後の時点で工具の有効長さを計算できます。同様に、状態を保存した工具と同じ条件下で使用すると仮定して、別の工具の有効長さを計算できます。

工具環境のデータセットの最大数

マシンデータ MD18116 $MN_MM_NUM_TOOL_ENV は、工具環境を記述するために保存できるデータセットの最大数を定義します。このデータは、TOA 領域に格納されます。また、コントロールシステムをオフにしても、データは維持されます。

データをバックアップすることはできません。つまり、異なるコントローラシステム間でこのデータを伝送することはできません。

11.12.2 工具環境の削除(DELTOOLENV)

DELTOOLENV 機能は、工具環境の記述に使用されるデータセットを削除します。削除とは、特定の名前で格納されているデータセットにアクセスできなくなる(アクセスしようとするとアラームが発生する)ことを指します。

注記

データセットを削除できるのは、DELTOOLENV 機能を使用する場合、INITIAL.INI ダウンロードによる場合、またはコールドスタートの(初期設定マシンデータを指定して NC が起動する)場合に限ります。自動削除操作が追加実行されることはありません。

構文

```
<状態> = DELTOOLENV(<名前>)
<状態> = DELTOOLENV()
```

意味

DELTOOLENV(...):	工具環境を削除するための定義済み機能	
	単独ブロック指令:	あり

<状態>:	機能の戻り値。負の値は異常状態を示します。		
	データタイプ:		INT
	値:	0	動作 OK
		-1	工具環境に予約されたメモリがありません。 MD18116 $MN_MM_NUM_TOOL_ENV = 0 これは、「工具環境」機能が使用できないことを示します。
		-2	指定名の工具環境が存在しません。
パラメータ			
1 <名称>:	削除するデータセットの名前		
	データタイプ:		STRING
DELTOOLENV():	DELTOOLENV()は、名前を指定せずに工具環境を記述するデータセットを削除します。		

11.12.3 T 番号、D 番号、および DL 番号の読み出し (GETTENV)

GETTENV 機能を使用すれば、工具環境内に格納された T、D、および DL の各番号を読み取ることができます。

構文

```
<状態> = GETTENV(<名前>, <TDDL>)
```

意味

GETTENV(...):	工具環境を記述するデータセット内の T 番号、D 番号、および DL 番号を読み取る事前定義された機能	
	単独ブロック指令:	あり

<状態>:	機能の戻り値。負の値は異常状態を示します。		
	データタイプ:		INT
	値:	0	動作 OK
		-1	工具環境に予約されたメモリがありません。 MD18116 $MN_MM_NUM_TOOL_ENV = 0 これは、「工具環境」機能が使用できないことを示します。
		-2	指定名の工具環境が存在しません。
パラメータ			
1 <名称>:	T、D、および DL の各番号を読み取るデータセットの名前		
	データタイプ:		STRING
2 <TDDL>:	この結果パラメータのフィールドには、工具の T 番号、D 番号、および DL 番号が含まれ、これらの工具環境は次のように指定のデータセットに保存されます。 • <TDDL> [0]:T 番号 • <TDDL> [1]:D 番号 • <TDDL> [2]:DL 番号		
	データタイプ:		INT[3]
GETTENV(,<TDDL>), GETTENV("",<TDDL>):	GETTENV 機能を呼び出す場合、1 番目のパラメータを省略するか、1 番目のパラメータとして空文字列を伝送することができます。この 2 つの特殊な場合のどちらでも、<TDDL> で、有効な工具の T、D、および DL 番号が戻されます。		

11.12.4 保存した工具環境に関する情報の読み取り($P_TOOLENVN、$P_TOOLENV)

保存された工具環境に関する情報は、次のシステム変数を使用して読み取ることができます。

$P_TOOLENVN:	工具環境を記述するために、TOOLENV を使用して定義された(まだ削除されていない)データセットの数を指定します			
	構文:	<n> = $P_TOOLENVN		
	意味	<n>:	定義されたデータセットの数	
			データタイプ:	INT
			値の範囲:	0 ... MD18116 $MN_MM_NUM_TOOL_ENV
	工具環境が使用できない場合(MD18116 = 0)も、このシステム変数にアクセスできます。この場合、戻り値は「0」になります。			
$P_TOOLENV:	工具環境を記述する**<i>**番目のデータセットの名前を指定します			
	構文:	<名前> = $P_TOOLENV[<i>]		
	意味	<名称>:	番号**<i>**で指定されたデータセットの名前	
			データタイプ:	STRING
		<i>:	データセットの番号	
			データタイプ:	INT
			値の範囲:	1 ... $P_TOOLENVN
	データセットへの番号の割り当ては、固定されていないため、データセットを削除や作成すると変更されます。データセットは内部的に番号が付けられます。 **<i>**が定義されていないデータセットを参照する場合、空文字列が返されます。 インデックス**<i>**が有効でない場合、つまり**<i>**が 1 より小さいか、または工具環境(MD18116 $MN_MM_NUM_TOOLENV)のデータセットの最大数より大きい場合、次のアラームが出力されます。 アラーム 17020「配列インデックス 1 が許可されません」			

11.12.5 工具長、工具長成分、またはその両方の読み取り(GETTCOR)

GETTCOR 機能を使用して、工具長または工具長コンポーネントを読み出すことができます。

各パラメータを使用して、対象とするコンポーネントと、工具を使用するときの条件を指定できます。

構文

```
<Status> = GETTCOR(<Len>[, <Comp>, <Stat>, <T>, <D>, <DL>])
```

意味

GETTCOR(...):	工具の長さを読み取るか、または工具の長さのコンポーネントを読み取るために事前定義された機能		
	単独ブロック指令:		あり
<状態>:	機能の戻り値。負の値は異常状態を示します。		
	データタイプ:		INT
	値:	0	動作 OK
		-1	工具環境に予約されたメモリがありません。 MD18116 $MN_MM_NUM_TOOL_ENV = 0 これは、「工具環境」機能が使用できないことを示します。
		-2	**<**状態**>**で指定された名前の工具環境が存在しません。
		-3	パラメータ**<Comp>**の文字列が無効です。 この異常の発生原因には、文字列が無効な場合または同じ文字列が 2 回プログラム指令されている場合があります。
		-4	T 番号が無効です。
		-5	D 番号が無効です。
		-6	DL 番号が無効です。
		-7	アクセスしようとしたメモリモジュールは存在しません。
		-8	アクセスしようとしたオプションは存在しません(プログラム可能な工具向き、工具管理機能)。
		-9	**<Comp>**文字列には、コロン(座標系の指定用識別子)が含まれていますが、その後に、座標系を示す有効な文字が指定されていません。
パラメータ			

1	<Len>:	得られるベクトル
		データタイプ: REAL[11]
		ベクトルコンポーネントは、次の順序で配列されます。 • <Len> [0]:工具タイプ • <Len> [1]:刃先位置 • <Len> [2]:横座標 • <Len> [3]:縦座標 • <Len> [4]:垂直座標 • <Len> [5]:工具半径 <Comp>および<Stat>で定義された座標系は、長さコンポーネントの基準座標系として使用されます。<Comp>に座標系が定義されていない場合、工具長は機械座標系で表示されます。 ジオメトリ軸への横軸、縦軸、および垂直軸の割り当ては、工具環境内で有効な平面によって違ってきます。つまり G17 では、縦軸は X に平行であり G18 では、Z 軸に平行になります。 <Len>[6]～<Len>[10]の各コンポーネントには、追加のパラメータが含まれており、これらを使って工具のジオメトリの記述を指定できます(たとえば、ジオメトリの場合は$TC_DP7 ～$TC_DP11、摩耗またはサムオフセットおよびセットアップオフセットの場合は対応するコンポーネント)。 これらの 5 つの追加要素と工具半径が定義されるのは、E、G、S、および W の各コンポーネントの場合に限ります。この評価は、<Stat>に左右されません。このため、<Len>[6]～<Len>[10]の対応する値をゼロ以外の値にできるのは、指定された 4 つのコンポーネントのうち少なくとも 1 つが工具長計算に関係している場合に限ります。残りのコンポーネントは、結果に影響を及ぼしません。各寸法は、制御の基本単位系(インチまたはメートル)を参照します。

2	<Comp>:	工具長コンポーネント(オプション)			
		データタイプ:	STRING		
		この文字列は、2 つの 2 次文字列で構成され、各 2 次文字列はコロンで区切られます。 一般形式: "<SubStr_1> [: <SubStr_2]"			
		<SubStr_1>:	**1 番目の 2 次文字列**は、工具長の計算時に考慮する工具長コンポーネントを示します。 2 次文字列内の文字の順序、およびその表記法(大文字か小文字か)は自由です。ブランクまたはタブは、文字の間に自由に挿入できます。 **注:** 2 次文字列内で文字を 2 回プログラムすることは**できません**。		
			文字:	-	マイナス符号(1 番目の文字としてのみ使用できます) すべての工具長コンポーネントから、マイナスに続く文字列のコンポーネントを差し引いて、計算されます。
				C	アダプタまたは工具ベース寸法(使用中の工具に対して有効な、2 つの選択可能な成分のどちらでも)
				E	セットアップオフセット
				G	ジオメトリ
				K	キネマティックトランスフォーメーション(総合 3、4、および 5 軸座標変換の場合に限り評価されます)
				S	サムオフセット
				T	旋回工具ホルダ
				W	摩耗
			1 番目の 2 次文字列が空の場合(ただし、タブは除きます)、すべてのコンポーネントを考慮して、すべての工具長が計算されます。**<Comp>**パラメータが指定されていなくても、この操作は適用されます。		
		<Substr_2>:	オプションの **2 番目の 2 次文字列**は、工具長を出力する座標系を示します。		

			2 番目の 2 次文字列は、1 つの当該文字のみから構成されます。		
			文字:	A	設定可能座標系(ACS)
				B	基本座標系(BCS)
				K	キネマティックトランスフォーメーションの工具座標系(KCS)
				M	機械座標系(MCS)
				T	工具座標系(TCS)
				W	ワーク座標系(WCS)
			座標系が指定されていない場合、評価は MCS (機械座標系)で実行されます。回転が考慮される場合、回転は、**<Stat>**で定義されている工具環境で指定されます。		
3	<_Stat>:	工具環境を記述するデータセットの名前(オプション)			
		データタイプ:	STRING		
		このパラメータの値が空文字列("")になっているか、指定されていない場合、現在の状態が使用されます。工具が指定されない場合、現在の工具が使用されます。			
4	<T>:	工具の内部 T 番号(オプション)			
		データタイプ:	INT		
		このパラメータが指定されていないか、その値が「0」の場合、**<Stat>**に設定されている工具が使用されます。 このパラメータの値が「-1」の場合、有効な工具の T 番号が使用されます。有効な工具の番号を明確に指定することもできます。 **注:** **<Stat>**が指定されていない場合、工具環境として現在の状態が使用されます。**<T> = 0** が、工具環境に保存されている T 番号を参照するため、その環境では、有効な工具が使用されます。つまり、この特殊な場合では、パラメータ**<T> = 0** と**<T> = -1** には同じ意味があります。		

5	<D>:	工具の刃先(オプション)
		データタイプ: INT
		このパラメータを指定しないか、その値が「0」になっている場合、使用される D 番号は、T 番号に基づいて決まります。工具環境からの T 番号を使用する場合、工具環境の D 番号も読み取られます。使用しない場合は、現在有効になっている工具の D 番号が読み取られます。
6	<DL>:	ロケーションに依存するオフセットの番号(オプション)
		データタイプ: INT
		このパラメータを指定しない場合、使用される DL 番号は、T 番号に基づいて決まります。工具環境からの T 番号を使用する場合、工具環境の D 番号も読み取られます。使用しない場合は、現在有効になっている工具の D 番号が読み取られます。

例

GETTCOR(_LEN)	すべてのコンポーネントを考慮して、機械座標系内で現在有効になっている工具の工具長を計算します。
GETTCOR(_LEN,"CGW:W")	アダプタまたは工具ベース寸法、ジオメトリ、および摩耗で構成される有効な工具の工具長を計算します。旋回工具ホルダまたはキネマティックトランスフォーメーションなどの他のコンポーネントは考慮されません。ワーク座標系で出力します。
GETTCOR (_LEN,"-K:B")	有効なキネマティックトランスフォーメーションの長さコンポーネントを考慮せずに、有効な工具の工具長全体を計算します。基本座標系で出力します。
GETTCOR (_LEN,":M","Testenv1",,3)	「Testenv1」と命名された工具環境に格納された工具に対して機械座標系で工具長全体を計算します。ただし、格納されている刃先番号に関係なく、刃先番号 D3 に計算を実行します。

他の情報

アダプタ座標変換/旋回工具ホルダ/キネマティックトランスフォーメーション

アダプタ座標変換、旋回工具ホルダ、およびキネマティックトランスフォーメーションにより行われる回転とコンポーネントの入れ替えは、工具環境の一部です。このため、対応する長さコンポーネントが含まれていないと考えられる場合でも必ず実行されます。この動作が望ましくない場合、対応する座標変換が無効である工具環境を定義してください。多くの場合(つまり、機械で、座標変換、または旋回工具ホルダを使用しない場合はいつでも)、工具環境用に格納されているデータセットは自動的にこれらの条件を満たすので、ユーザーは、特殊な用意をする必要がありません。

旋削工具および研削工具 MD20360 $MC_TOOL_PARAMETER_DEF_MASK に依存する工具長の計算

次のマシンデータは、旋削工具と研削工具で直径軸が使用される場合に、摩耗および工具長を評価する方法を定義します。

MD20360 $MC_TOOL_PARAMETER_DEF_MASK (工具パラメータの定義)

ビット	値	
0	旋削工具と研削工具では、経方向軸の**摩耗パラメータ**は直径値として考慮されます。	
	= 0 (初期設定)	なし
	= 1	あり
1	旋削工具と研削工具では、経方向軸の**工具長コンポーネント**は直径値として考慮されます。	
	= 0 (初期設定)	なし
	= 1	あり

関連するこの 2 つのビットが設定されている場合、対応する入力値は、係数 0.5 で重み付けされます。この重み付けは、GETTCOR で返される工具長に反映されます。

例:

MD20360 $MC_TOOL_PARAMETER_DEF_MASK = 3

MD20100 $MC_DIAMETER_AX_DEF (径方向軸機能によるジオメトリ軸) = "X"

X は径方向軸です(標準的な旋盤構成)

プログラムコード	コメント
N30 $TC_DP1[1,1]=500	
N40 $TC_DP2[1,1]=2	
N50 $TC_DP3[1,1]=3.0	; ジオメトリ L1
N60 $TC_DP4[1,1]=4.0	
N70 $TC_DP5[1,1]=5.0	
N80 $TC_DP12[1,1]=12.0	; 磨耗 L1
N90 $TC_DP13[1,1]=13.0	
N100 $TC_DP14[1,1]=14.0	
N110 T1 D1 G18	
N120 R1=GETTCOR(_LEN,"GW")	
N130 R3=_LEN[2]	; 17.0 (= 4.0 + 13.0)
N140 R4=_LEN[3]	; 7.5 (= 0.5 * 3.0 + 0.5 * 12.0)
N150 R5=_LEN[4]	; 19.0 (= 5.0 + 14.0)
N160 M30	

キネマティックトランスフォーメーションおよび旋回工具ホルダの長さコンポーネント

工具長の計算中、**旋回工具ホルダ**を考慮する場合、この計算には、次の各ベクトルが含まれています。

タイプ	ベクトル
M	l1 および l2
T	l1、l2、および l3
P	工具長は、旋回工具ホルダの影響を受けません。

総合 **5 軸座標変換**では、変換タイプ 24 と 56 の工具長計算に、次のマシンデータが使用されます。

座標変換タイプ	マシンデータ
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

座標変換タイプ 56(移動工具および移動ワーク)は、旋回工具ホルダのタイプ M に対応します。

以前のソフトウェアリリースの場合、この 5 軸座標変換では、ベクトル MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2 (チャネルの 1 番目/2 番目の 5 軸座標変換のキネマティックオフセットのベクトル)は、タイプ M の旋回工具ホルダの 2 つのベクトル l_1 および l_3 の合計に対応します。

どちらの場合も、合計のみが座標変換に関連します。2 つの個別のコンポーネントを構成する方法は意味がありません。ただし、工具長を計算する場合、どちらのコンポーネントを工具に割り当て、どちらのコンポーネントを工具台に割り当てるかは重要です。これは、マシンデータ MD24558/24658 $MC_TRAFO5_JOINT_OFFSET_PART_1/2 (表のベクトルキネマティックオフセット)が導入された理由です。これはベクトル l3 に対応します。マシンデータ: MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2 は、l1 と l3 の合計に対応しなくなり、ベクトル l1 にのみ対応します。マシンデータ MD24558/24658 $MC_TRAFO5_JOINT_OFFSET_PART_1/2 がゼロである場合、動作は以前と同じです。

互換性

GETTCOR 機能を、TOOLENV および SETTCOR 機能と一緒に使用すれば、計測サイクルで以前外部で実行されていた機能の一部を置換できます。

有効な工具長を実際に特定するパラメータの一部のみが、計測サイクルで実行されていました。上記各機能を使用して、工具長の計算に関連する計測サイクルの挙動を再現できます。

11.12.6 工具成分の変更(SETTCOR)

SETTCOR 機能を使用して、個々の成分を評価するときに、関連するすべての一般条件を考慮して、工具成分を変更することができます。

注記

用語について:以下において、工具長と組み合わせた工具成分が関係する場合、成分はすべての工具長を形成する、ベクトルの観点から考慮される成分、たとえば、ジオメトリや磨耗などを意味します。そのような成分は、3 つの個々の値(L1、L2、L3)で構成され、この値を以後、座標値と呼びます。

したがって、工具成分「ジオメトリ」は、3 つの座標値$TC_DP3 ～$TC_DP5 で構成されます。

構文

```
<Status> = SETTCOR(<CorVal>, <Comp>, [<CorComp>, <CorMode>,
<GeoAx>, <Stat>, <T>, <D>, <DL>])
```

意味

SETTCOR(...):	工具成分を変更するための事前定義された機能		
	単独ブロック指令:		あり
<状態>:	機能の戻り値。負の値は異常状態を示します。		
	データタイプ:		INT
	値:	0	動作 OK
		-1	工具環境に予約されたメモリがありません。 MD18116 $MN_MM_NUM_TOOL_ENV = 0 これは、「工具環境」機能が使用できないことを示します。
		-2	**<**状態**>**で指定された名前の工具環境が存在しません。
		-3	パラメータ**<Comp>**の文字列が無効です。 この異常の発生原因には、文字列が無効な場合または同じ文字列が 2 回プログラム指令されている場合があります。
		-4	T 番号が無効です。
		-5	D 番号が無効です。
		-6	DL 番号が無効です。
		-7	アクセスしようとしたメモリモジュールは存在しません。
		-8	アクセスしようとしたオプションは存在しません(プログラム可能な工具向き、工具管理機能)。
		-9	パラメータ**<CorComp>**の数値が不正です。
		-10	パラメータ**<CorMode>**の数値が不正です。
		-11	パラメータ**<Comp>**および**<CorComp>**の内容が矛盾しています。
		-12	パラメータ**<Comp>**および**<CorMode>**の内容が矛盾しています。
		-13	**<GeoAx>**パラメータの内容がジオメトリ軸を指定していません。
		-14	書き込もうとしたセットアップオフセットは存在しません。

パラメータ			
1	<CorVal>:	補正ベクトル **<Stat>**によって定義されたワーク座標系(WCS)では、次の割り当てが適用されます。 • **<CorVal>** [0]:横座標 • **<CorVal>** [1]:縦座標 • **<CorVal>** [2]:垂直座標 1 つの工具成分だけを補正する場合(つまり、ベクトルの補正は行われません。パラメータ**<CorMode>**を参照してください)、補正値は常に、その成分が作用する軸には関わりなく**<CorVal>**[0]にあります。残りの 2 つの成分の内容は、評価されません。 **<CorVal>**、または**<CorVal>**の成分が径方向軸を参照する場合、データは**半径寸法**として評価されます。つまり、たとえば、工具が、指定された寸法だけ「長く」なります。これに応じて、ワークの直径が 2 倍に変更されます。 各寸法は、コントロールシステムの**基本単位系** (インチまたはメトリック)を基準にしています。	
		データタイプ:	REAL[3]

2	<Comp>:	工具成分			
		データタイプ:	STRING		
		この文字列は、2 つの 2 次文字列で構成され、各 2 次文字列はコロンで区切られます。 一般形式: "<SubStr_1> [: <SubStr_2]"			
		<SubStr_1>:	**1 番目の 2 次文字列**は常に必要であり、1 文字または2 文字を含めることができます。第 1 の構成部分(Val_1)の単一の文字、および第 2 の構成部分(Val_2)の第 2 の文字。後続のパラメータ**<CorComp>**および**<CorMode>**に従って処理されます。		
			文字:	C	アダプタまたは工具ベース寸法(使用中の工具に対して有効な、2 つの選択可能な成分のどちらでも)
				E	セットアップオフセット
				G	ジオメトリ
				S	サムオフセット
				W	摩耗
		<Substr_2>:	**2 番目の 2 次文字列**はオプションです。また、(個別の)文字「W」または「T」を構成できます。		
			文字:	W	2 番目の 2 次文字列が空白であるか、文字「W」を含む場合、あたかも**ワーク**座標系(WCS)で計測されたかのように、オフセット値が考慮されます。
				T	2 番目の 2 次文字列に文字「**T**」が含まれる場合、あたかも**工具**座標系(Tool Coordinate System、TCS)で計測されたかのように、オフセット値が考慮されます。
		文字列内の文字の表記法(大文字か小文字か)は自由です。空白またはタブ(余白)をいくつでも挿入できます。			

3	<CorComp>:	記述するツールデータセットの成分を指定します(オプション)。		
		データタイプ:		INT
		値:	0	オフセット値<CorVal>[0]は、ワーク座標系または工具座標系で、パラメータ<GeoAx>に伝送されたジオメトリ軸を示します(パラメータ<Comp>の説明も参照)。つまり、オフセット値は、工具長の計算に影響を及ぼすすべてのパラメータを考慮して、指定された工具成分で計算する必要があります。工具長全体が指定された軸方向に指定された値だけ変更されます。 この変更は、<Comp>で指定された成分と<CorMode>で指定されたシンボリックアルゴリズムの修正によって実現してください(以下のパラメータを参照してください)。そのため、結果として得られた修正は、3 つの軸成分のすべてに影響を及ぼします。
			1	0 の場合と同じ。ただし、ベクトル。ベクトル<CorVal>の内容は、ワーク座標系または工具座標系で横軸、縦軸、および垂直軸を示します(パラメータ<Comp>の説明を参照)。 後続のパラメータ<GeoAx>は評価されません。
			2	ベクトルオフセット、つまり、L1、L2、および L3 は同時に変更することができます。 「0」と「1」のタイプとは違い、<CorVal>に含まれるオフセット値は工具の Val_1 成分の座標(後述のパラメータ<CorMode>を参照してください)を基準にします。 ワーク座標系と比較した、既存の工具の考えられるすべての傾きはオフセットには影響しません。
			3 - 5	工具長 L1 ～ L3 ($TC_DP3 ～$TC_DP5)または対応する磨耗の値の補正、設定オフセットまたは追加オフセット. オフセット値は<CorVal>[0]に含まれています。オフセット値は、工具の Val_1 成分の座標で計測されます(後述のパラメータ<CorMode>を参照してください)。ワーク座標系と比較した、既存の工具の考えられるすべての傾きはオフセットには影響しません。
			6	工具半径($TC_DP6)または対応する磨耗値の補正、設定オフセット、または追加オフセットマシンデータ MD20360 $MC_TOOL_PARAMETER_DEF_MASK で

				ビット 10 および 11(半径または直径として指定された直径/直径摩耗データの評価)が考慮されます。
			7 – 11	$TC_DP7 ～$TC_DP11 または対応する磨耗値の補正、設定オフセット、または追加オフセットこれらのパラメータは、工具半径とまったく同じように処理されます。
		このパラメータを指定しない場合、値は「0」になります。		

<table>
<tr><td>4</td><td><CorMode>:</td><td colspan="3">実行する書き込み動作のタイプを指定します(オプション)。</td></tr>
<tr><td></td><td></td><td colspan="2">データタイプ:</td><td>INT</td></tr>
<tr><td></td><td></td><td rowspan="4">値:</td><td>0</td><td>Val_{1new} = <CorVal></td></tr>
<tr><td></td><td></td><td>1</td><td>Val_{1new} = Val_{1old} + <CorVal></td></tr>
<tr><td></td><td></td><td>2</td><td>Val_{1new} = <CorVal>
Val_{2new} = 0</td></tr>
<tr><td></td><td></td><td>3</td><td>Val_{1new} = Val_{1old} + Val_{2old} + <CorVal>
Val_{2new} = 0</td></tr>
<tr><td></td><td></td><td colspan="3">表記 Val_{1old} + Val_{2old} はシンボル表記です。2 つの成分が(<_Stat>の状態のために)別々の方法で評価される場合、つまり、2 つの成分間で回転が有効である場合、Val_{2old} は追加の前に座標変換されるため、Val_{2new} の削除後と<CorVal>の追加前に得られる工具長は変わりません。

<CorVal>は常に、Val_1 を参照します。<CorVal>は、パラメータ<Comp>の 2 番目の部分に依存し、ワーク座標系(WCS)または工具座標系(TCS)で計測される値です。そのため、計算がおこなわれる工具成分に対して既に座標変換済みです。したがって、保存されている値と一緒に直接計算することはできず、Val_1 または Val_2 に追加する前に座標変換し直す必要があります。.つまり、このオフセットは、<CorComp>で定義された軸とは異なった軸に作用します – または、複数の軸に作用します。

<CorComp> = 0 の場合、つまり、<CorVal>にベクトルが含まれておらず、個々の値だけが含まれている場合、記述された動作は、<CorVal>が計測された座標(WCS/TCS)で実行されます。特に、これは、タイプ 2 および 3 で Val_{2new} を 0 に設定する場合にも適用されます。そして、その結果は、工具の座標に再変換されます。つまり、0 に設定される座標値(L1、L2、L3)のいずれもが 0 にならないか、以前に 0 であった座標値が 0 と等しくなくなります。ただし、対応する動作が 3 つのジオメトリ軸すべてに対して正常に実行された場合、削除される成分の 3 つの座標値はすべて常に 0 になります。また、工具がワーク座標系を基準にして回転しない場合、または、すべての工具成分が座標軸と平行を保つように回転する場合(軸交換動作)、1 つの工具座標だけが確実に変更されます。

<CorComp> = 0 を使用して、同じ動作(<CorMode>)を 3 つの座標軸すべてに対して任意の順序で連続して実行するのは、同じ動作を <CorComp> = 1 で単独で実行するのと同じです。</td></tr>
</table>

		パラメータ値「0」および「1」ではパラメータ<Comp>に 1 文字を含める必要があり、パラメータ値「2」および「3」では 2 文字を含める必要があります。 例: <Comp>には文字列「ES」、<CorMode>には値「2」を含めます ⇒ Setup offset$_{new}$ = <CorVal>, summed offset$_{new}$ = 0 パラメータ<CorMode>を指定しない場合、値は「0」になります。	
5	<GeoAx>:	オフセット値[0]を読み取るジオメトリ軸のインデックスを指定します(オプション)。	
		データタイプ:	INT
		値の範囲:	0 ... 2
		インデックス 0 ～ 2 は、現在の工具環境で有効な平面(G17/G18/G19)の横軸、縦軸、および垂直軸を示します。 パラメータ<CorComp>の値が「0」である場合、このパラメータの内容のみが評価されます。	
6	<Stat>:	工具環境を記述するデータセットの名前(オプション)	
		データタイプ:	STRING
		このパラメータの値が空文字列("")になっているか、指定されていない場合、現在の状態が使用されます。工具が指定されない場合、現在の工具が使用されます。	
7	<T>:	工具の内部 T 番号(オプション)	
		データタイプ:	INT
		このパラメータが指定されていないか、その値が「0」の場合、<Stat>に設定されている工具が使用されます。 このパラメータの値が「-1」の場合、有効な工具の T 番号が使用されます。有効な工具の番号を明確に指定することもできます。 **注:** <Stat>が指定されていない場合、工具環境として現在の状態が使用されます。<T> = 0 が、工具環境に保存されている T 番号を参照するため、その環境では、有効な工具が使用されます。つまり、この特殊な場合では、パラメータ<T> = 0 と<T> = -1 には同じ意味があります。	

8	<D>:	工具の刃先(オプション)
		データタイプ: INT
		このパラメータを指定しないか、その値が「0」になっている場合、使用される D 番号は、T 番号に基づいて決まります。工具環境からの T 番号を使用する場合、工具環境の D 番号も読み取られます。使用しない場合は、現在有効になっている工具の D 番号が読み取られます。
9	<TL>:	ロケーションに依存するオフセットの番号(オプション)
		データタイプ: INT
		このパラメータを指定しない場合、使用される DL 番号は、T 番号に基づいて決まります。工具環境からの T 番号を使用する場合、工具環境の D 番号も読み取られます。使用しない場合は、現在有効になっている工具の D 番号が読み取られます。T、D、および DL が、ロケーションに依存するオフセットなしで工具を指定している場合、サムオフセットまたは設定オフセットをパラメータ**<Comp>**で指定することはできません(**<Status>**のエラーコード)。

注記

3 つのパラメータ**<Comp>**、**<CorComp>**、および**<CorMode>**すべての組合わせに意味があるわけではありません。たとえば、**<CorComp>**のアルゴリズム 3 では、**<Comp>**で 2 文字を指定する必要があります。.無効なパラメータの組み合わせが指定されている場合、対応するエラーコードが**<Status>**で返されます。

例

例 1

プログラムコード	コメント
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1] = 120	; フライス工具
N30 $TC_DP3[1,1] = 10.0	; ジオメトリ L1
N40 $TC_DP12[1,1]=1.0	; 磨耗 L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=**SETTCOR(_CORVAL,"G",0,0,2)**	
N80 T1 D1 X0 Y0 Z0	; ==> MCS 位置 X0.000 Y0.000 **Z1.333**
N90 M30	

<CorComp>は「0」であるため、Z 方向で作用するジオメトリ軸の座標値をオフセット値 0.333 と置き換えてください。

その結果、合計工具長は次のようになります。L1 = 0.333 + 1.000 = 1.333

例 2

プログラムコード	コメント
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1] = 120	; フライス工具
N30 $TC_DP3[1,1] = 10.0	; ジオメトリ L1
N40 $TC_DP12[1,1]=1.0	; 磨耗 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	;==> MCS 位置 X0.000 Y0.000 Z11.333
N90 M30	

<CorComp>は「1」です。これはつまり、Z 軸に作用する 0.333 のオフセット値が摩耗値 1.0 に追加されることを意味します。

その結果、合計工具長は次のようになります。L1 = 10.0 + 1.333 = 11.333

例 3

プログラムコード	コメント
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1] = 120	; フライス工具
N30 $TC_DP3[1,1] = 10.0	; ジオメトリ L1
N40 $TC_DP12[1,1]=1.0	; 磨耗 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	; ==> MCS 位置 X0.000 Y0.000 Z0.333
N90 M30	

<CorComp>は「2」であるため、Z 軸で有効なオフセットがジオメトリコンポーネントに入力され(前の値は上書きされます)、磨耗値が削除されます。

その結果、合計工具長は次のようになります。L1 = 0.333 + 0.0 = 0.333

例 4

プログラムコード	コメント
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1] = 120	; フライス工具
N30 $TC_DP3[1,1] = 10.0	; ジオメトリ L1
N40 $TC_DP12[1,1]=1.0	; 磨耗 L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	

プログラムコード	コメント
N70 R1=**SETTCOR(_CORVAL,"GW",0,3,2)**	
N80 T1 D1 X0 Y0 Z0	;==> MCS 位置 X0.000 Y0.000 Z11.333
N90 M30	

<CorComp>は「3」であるため、磨耗値と補正値がジオメトリコンポーネントに加算され、磨耗コンポーネントが削除されます。

その結果、合計工具長は次のようになります。L1 = 11.333 + 0.0 = 11.333

例 5

プログラムコード	コメント
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1] = 120	; フライス工具
N30 $TC_DP3[1,1] = 10.0	; ジオメトリ L1
N40 $TC_DP12[1,1]=1.0	; 磨耗 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	;==> MCS 位置 X0.333 Y0.000 Z11.000
N90 M30	

前述の例では、<CorComp>は「3」ですが、インデックス「0」(X 軸)のジオメトリ軸に対して現在補正が有効であり、これはフライス工具で、G17 により工具成分 L3 に割り当てられます。その結果として、SETTCOR の呼び出し時に、工具パラメータ$TC_DP3 および$TC_DP12 は影響を受けません。その代わりに、$TC_DP5 に補正値が入力されます。

例 6

プログラムコード	コメント
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; 旋削工具
N30 $TC_DP3[1,1] = 10.0	; ジオメトリ L1
N40 $TC_DP4[1,1]=15.0	; ジオメトリ L2
N50 $TC_DP12[1,1]=10.0	; 磨耗 L1
N60 $TC_DP13[1,1]=0.0	; 磨耗 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	; ==> MCS 位置 X24.330 Y0.000 Z17.500
N120 M30	

この工具は旋削工具です。N80 でフレーム回転が起動されるため、基本座標系(BCS)がワーク座標系(WCS)に対して回転します。WCS では、G18 が有効なため、インデックス 1 のジオメトリ軸、つまり X 軸に補正値(N70)が影響を及ぼします。<CorMode> = 3 であるため、N100 が実行された後、WCS の X 軸方向での工具磨耗を 0 でなくてはなりません。

プログラム終了時の関連工具パラメータの内容は、次のとおりです。

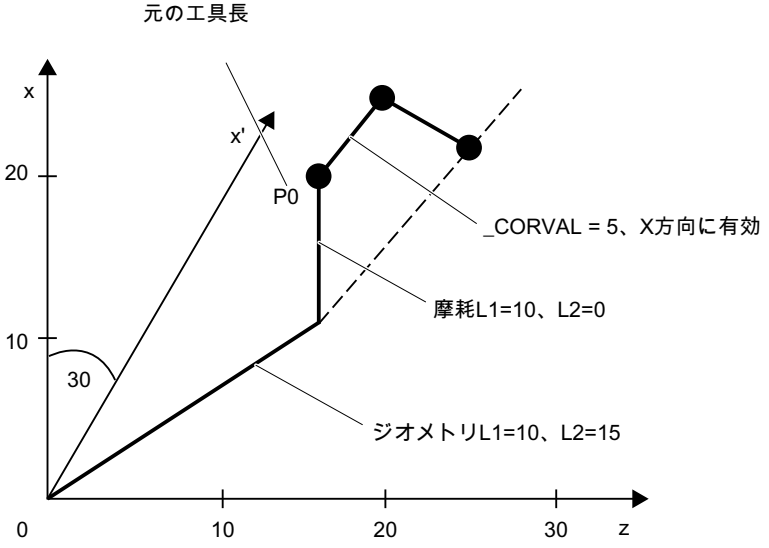
$TC_DP3[1,1]:21.830 ; ジオメトリ L1

$TC_DP4[1,1] :21.830 ; ジオメトリ L2

$TC_DP12[1,1] :2.500 ; 磨耗 L1

$TC_DP13[1,1] : -4.330 ; 磨耗 L2

次の図に、ジオメトリの関係を示します。_CORVAL を含む磨耗合計が、WCS の X 方向に割り当てられます。これによって点 P2 が生成されます。この点の座標(X/Y 座標で測定)が、工具のジオメトリコンポーネントに入力されます。差分ベクトル P_2 - P_1 が引き続き磨耗に含まれます。したがって、摩耗には_CORVAL 方向のコンポーネントがなくなります。



このプログラムの例で、N110 の後、次の命令が続く場合、補正が Z 軸で有効になるため(パラメータ<GeoAx> = 0)、残りの磨耗は完全にジオメトリに含まれます。

```
N120 _CORVAL[0]=0.0
N130 R1=SETTCOR(_CORVAL,"GW",0,3,0)
N140 T1 D1 X0 Y0 Z0 ; ==> MCS 位置 X24.330 Y0.000 Z17.500
```

新しい補正値は「0」であるため、合計工具長と、それに応じて N140 でアプローチする位置は、変更されない場合があります。N120 で_CORVAL が「0」に等しくない場合、新しい合計工具長と、それに応じて N140 でアプローチする新しい位置が生成されますが、工具長の磨耗コンポーネントは常に 0 になります。つまり、合計工具長はその後常に工具のジオメトリコンポーネントに含まれます。

パラメータ<CorComp> = 0 を指定して SETTCOR 機能を 2 回呼び出した場合と同じ結果が、<CorComp> = 1 (ベクトル補正)を 1 回だけ呼び出しても得られます。

プログラムコード	コメント
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; 旋削工具
N30 $TC_DP3[1,1] = 10.0	; ジオメトリ L1
N40 $TC_DP4[1,1]=15.0	; ジオメトリ L2
N50 $TC_DP12[1,1]=10.0	; 磨耗 L1
N60 $TC_DP13[1,1]=0.0	; 磨耗 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	; ==> MCS 位置 X24.330 Y0.000 Z17.500
N120 M30	

この場合、工具のすべての磨耗コンポーネントが、N100 における SETTCOR の最初の呼び出しの直後に 0 に設定されます。

例 7

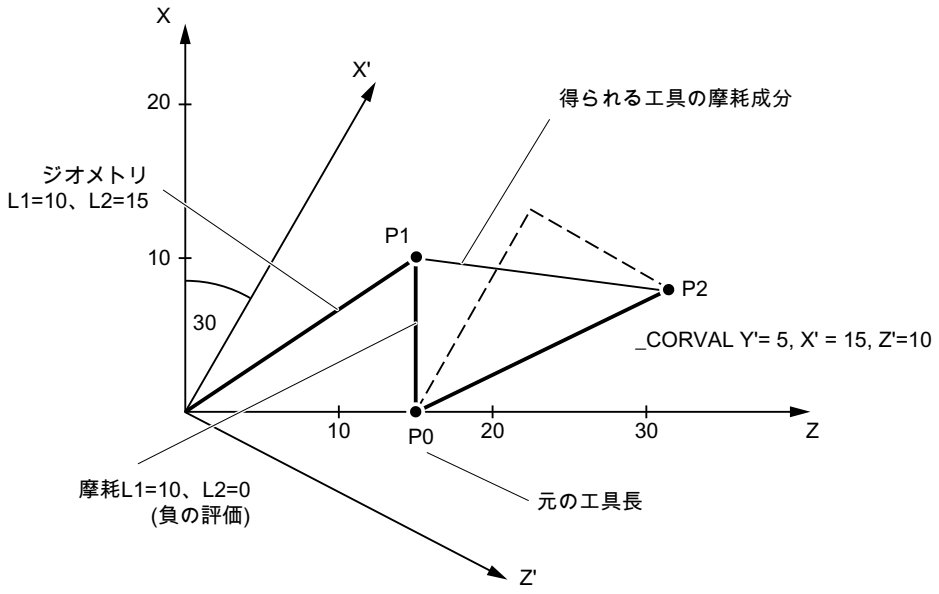
プログラムコード	コメント
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; 旋削工具
N30 $TC_DP3[1,1] = 10.0	; ジオメトリ L1
N40 $TC_DP4[1,1]=15.0	; ジオメトリ L2
N50 $TC_DP12[1,1]=10.0	; 磨耗 L1
N60 $TC_DP13[1,1]=0.0	; 磨耗 L2
N70 _CORVAL[0]=5.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=**SETTCOR(_CORVAL,"GW",3,3)**	
N110 T1 D1 X0 Y0 Z0	; ==> MCS 位置 X25.000 Y0.000 Z15.000
N120 M30	

例 6 とは対照的に、パラメータ<CorComp> = 3 であり、したがって<GeoAx>パラメータを省略できます。_CORVAL[0]に含まれる値が工具長コンポーネント L1 に即座に影響を及ぼすようになり、N80 での回転は結果に影響せず、$TC_DP12 の磨耗コンポーネントは_CORVAL[0]とともにジオメトリコンポーネントに含まれ、その結果 N100 における SETTCOR の最初の呼び出し後に$TC_DP13 により合計工具長がジオメトリコンポーネントに格納されます。

例 8

プログラムコード	コメント
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; 旋削工具
N30 $TC_DP3[1,1] = 10.0	; ジオメトリ L1
N40 $TC_DP4[1,1]=15.0	; ジオメトリ L2
N50 $TC_DP5[1,1]=20.0	; ジオメトリ L3
N60 $TC_DP12[1,1]=10.0	; 磨耗 L1
N70 $TC_DP13[1,1]=0.0	; 磨耗 L2
N80 $TC_DP14[1,1]=0.0	; 磨耗 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	; ==> MCS 位置 X7.990 Y25.000 Z31.160
N170 M30	

N90 でセッティングデータ:SD42930 $SC_WEAR_SIGN が有効です。つまり、摩耗は負の符号で評価する必要があります。この補正はベクトル補正(<CorComp> = 1)です。補正ベクトルを磨耗(<CorMode> = 1)に加算してください。Z/X 平面でのジオメトリの関係を次の図に示します。



工具のジオメトリ成分は、<CorMode> = 1 であるために変更されません。図 3 の合計工具長の基準が点 P_2 になるように、WCS で定義された補正ベクトル(Y 軸を中心とする回転)を磨耗成分に含めてください。したがって、得られる工具の摩耗成分は、2 点 P_1 および P_2 の距離によって示されます。
ただし、セッティングデータ SD42930 $SC_WEAR_SIGN により、摩耗は負で評価されるので、この方法で決定された補正を負符号とともに補正メモリに入力する必要があります。プログラム終了時の関連工具パラメータの内容は、次のとおりです。

$TC_DP3[1,1]:10.000 ; ジオメトリ L1 (不変)

$TC_DP4[1,1] :15.000 ; ジオメトリ L2 (不変)

$TC_DP5[1,1]:10.000 ; ジオメトリ L3 (不変)

$TC_DP12[1,1] :2.010 ; 摩耗 L1 (= 10 - 15 * cos(30) + 10 * sin(30))

$TC_DP13[1,1] : -16.160 ; 摩耗 L2 (= -15 * sin(30) - 10 * cos(30))

$TC_DP14[1,1] : -5.000 ; 磨耗 L3

Y 方向の L3 コンポーネントに対するセッティングデータ SD42930 $SC_WEAR_SIGN の影響は、フレーム回転によってさらに複雑化されることなく認識できます。

他の情報

旋削/研削工具:MD20360 $MC_TOOL_PARAMETER_DEF_MASK に依存する工具長の計算

次のマシンデータは、旋削/研削工具で直径軸が使用される場合に、摩耗および工具長を評価する方法を定義します。

MD20360 $MC_TOOL_PARAMETER_DEF_MASK.<ビット> = <値>

<ビット>	<値>	意味
0	0	旋削/研削工具では、経方向軸の**摩耗パラメータ**が**半径値**で考慮されます:
	1	旋削/研削工具では、経方向軸の**摩耗パラメータ**が**直径値**として考慮されます:
1	0	旋削/研削工具では、経方向軸の**工具長成分**が**半径値**として考慮されます:
	1	旋削/研削工具では、経方向軸の**工具長成分**が**直径値**として考慮されます:

関連するこの 2 つのビットが設定されている場合、対応する入力値は、係数 0.5 で重み付けされます。SETTCOR を使用した補正が、有効な工具長全体の変更が<CorVal>で転送される値と等しくなるよう実行されます。長さを計算する場合、マシンデータ MD20360 $MC_TOOL_PARAMETER_DEF_MASK の結果として、係数 0.5 で長さを評価し、次に伝送された値の 2 倍でこの成分の補正を実行する必要があります。

例

MD20360 $MC_TOOL_PARAMETER_DEF_MASK = 2 (工具長は、係数 0.5 を使用して、直径軸で評価する必要があります)

軸 X が直径軸です

プログラムコード	コメント
N10 DEF REAL _LEN[11]	
N20 DEF REAL _CORVAL[3]	
N30 $TC_DP1[1,1]=500	; 工具タイプ
N40 $TC_DP2[1,1]=2	; 刃先位置
N50 $TC_DP3[1,1]=3.	; ジオメトリ - 長さ 1
N60 $TC_DP4[1,1]=4.	; ジオメトリ - 長さ 2
N70 $TC_DP5[1,1]=5.	; ジオメトリ - 長さ 3
N80 _CORVAL[0]=1.	
N90 _CORVAL[1]=1.	
N100 _CORVAL[2]=1.	
N110 T1 D1 G18 G0 X0 Y0 Z0	; ==> MCS 位置 X1.5 Y5 Z4
N120 R1=SETTCOR(_CORVAL,"G",1,1)	
N130 T1 D1 X0 Y0 Z0	; ==> MCS 位置 X2.5 Y6 Z5
N140 R3=$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)

プログラムコード	コメント
N170 M30	

各軸では、工具長の補正は 1 mm (N80 ～ N100)であることが必要です。したがって、長さ L2 および L3 の元の長さに 1 mm が加算されます。要求どおり合計長を 1 mm 変更するために、L1 の元の長さに 2 倍の補正値(2 mm)が加算されます。ブロック N110 および N130 でアプローチする位置を比較すると、各軸位置が 1 mm 変更されたことがわかります。

11.13 座標軸への工具長 L1、L2、L3 の割り当ての読み取り(LENTOAX)

「LENTOAX」機能は、横軸、縦軸、および垂直軸への**有効な**工具の工具長 L1、L2、および L3 の割り当てに関する情報を提供します。ジオメトリ軸への横軸、縦軸、および垂直軸の割り当ては、フレームと有効な平面(G17 ～ G19)の影響を受けます。

工具のジオメトリ成分($TC_DP3[<t>,<d>]～$TC_DP5[<t>,<d>])のみが考慮されます。つまり、他の成分(たとえば、摩耗)に別の軸割り当てを実行しても、結果に影響しません。

構文

```
<Status> = LENTOAX(<AxInd>, <Matrix>[, <Coord>])
```

原理

LENTOAX(...):	座標軸に対する有効な工具の工具長 L1、L2、および L3 の割り当てを読み取るために事前定義された機能		
	単独ブロック指令:		あり
<状態>:	機能の戻り値。負の値は異常状態を示します。		
	データタイプ:		INT
	値:	0	動作 OK **<AxInd>**の情報は、記述として十分です(工具長成分はすべて、ジオメトリ軸と平行です)。
		1	機能 OK。ただし、適切な記述かどうか、**<Matrix>**の内容を評価してください(工具長成分は、ジオメトリ軸に平行ではありません)。
		-1	パラメータ**<Coord>**の文字列が無効です。
		-2	工具が有効になっていません。
パラメータ			

<table>
<tr><td rowspan="5">1</td><td rowspan="5"><AxInd>:</td><td colspan="3">工具長成分がジオメトリ軸と平行な場合、長さ成分 L1 ～ L3 に割り当てられた軸インデックスは、<AxInd>配列で返されます。
● <AxInd> [0]:横座標
● <AxInd> [1]:縦座標
● <AxInd> [2]:垂直座標</td></tr>
<tr><td colspan="2">データタイプ:</td><td>INT[3]</td></tr>
<tr><td rowspan="2">値:</td><td>0</td><td>既存の割り当てなし(軸はありません)</td></tr>
<tr><td>1 ... 3
または
-1 ... -3</td><td>対応する座標軸で有効な長さの番号。
工具長成分が、負の座標方向を向いている場合には、符号が負になります。</td></tr>
<tr><td colspan="3">一部の長さ成分が、ジオメトリ軸に平行または逆平行になっていない場合、工具長成分の最も大きな部分が含まれている軸のインデックスが<AxInd>で返されます。この場合(別の理由で機能が異常を返さない場合)、戻り値は<Status> = 1 です。ジオメトリ軸 1 ～ 3 に対する工具長成分 L1 ～ L3 のマッピングは、2 番目のパラメータ <Matrix>の内容によって完全に記述されます。</td></tr>
<tr><td rowspan="3">2</td><td rowspan="3"><Matrix>:</td><td colspan="3">座標軸(横軸、縦軸、垂直軸)のベクトルに対する工具長のベクトル(L1=1、L2=1、L3=1)を表すマトリクスです。つまり工具長成分は、L1, L2, L3 の順で列に割り当てられ、各軸は、横軸、縦軸、垂直軸の順に行に割り当てられます。</td></tr>
<tr><td colspan="2">データタイプ:</td><td>REAL</td></tr>
<tr><td colspan="3">座標軸に所属するジオメトリ軸を使用できなくても、つまり、<AxInd> 内の対応する入力値が 0 でも、マトリクス内の要素はすべて常に有効です。</td></tr>
</table>

3	<Coord>:	割り当てに使用する座標系(オプション)		
		データタイプ:		STRING
		文字:	MCS M	工具長が、機械座標系で表されます。
			BCS B	工具長が、基本座標系で表されます。
			WCS W	工具長が、ワーク座標系で表されます(初期設定)。
			KCS K	工具長が、キネマティック座標変換の工具座標系で表されます。
			TCS T	工具長が、工具座標系で表されます。
		文字列内の文字の表記法(大文字か小文字か)は自由です。 パラメータ**<Coord>**を指定しない場合、WCS が使用されます(初期設定)。		

注記

TCS では、工具長成分はすべて常に軸と平行か逆平行です。
成分を逆平行にできるのは、ミラーリングが有効になっており、次のセッティングデータが有効になっている場合に限ります。
SD42900 $SC_MIRROR_TOOL_LENGTH (ミラーリング時の工具長符号変更)

例

標準アプリケーション、G17 のフライス工具。

L1 は Z 軸(垂直軸)に適用され、L2 は Y 軸(縦軸)に適用され、L3 は X 軸(横軸)に適用されます。

次の形式で機能呼び出しを実行します。

```
<Status>=LENTOAX(<AxInd>,<Matrix>,"WCS")
```

結果パラメータ**<AxInd>**には次の値が含まれます。

<AxInd>[0] = 3

<AxInd>[1] = 2

<AxInd>[2] = 1

または、省略形では次のようになります。(3, 2, 1)

この場合、当該のマトリクス(<Matrix>)は次のとおりです。

$$\text{<Matrix>} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

G17 から G18 または G19 に変更しても、結果は変わりません。というのも、ジオメトリ軸への長さ成分の割り当ては、横軸、縦軸、垂直軸の割り当てと同じように変化するからです。

ここで、60°の Z 軸フレーム回転が、G17 が有効な状態でプログラム指令されます。たとえば
`ROT Z60` です

垂直軸方向(Z 方向)は変更されません。この時点で、L2 の主要成分は、新しい X 軸方向を向いており、L1 の主要成分は、負の Y 軸方向を向いています。その結果として、戻り値(<Status>)は「1」であり、<AxInd>には値(2、-3、1)が含まれます。

この場合、当該のマトリクス(<Matrix>)は次のとおりです。

$$\text{<Matrix>} = \begin{pmatrix} 0 & \sin 60° & \cos 60° \\ 0 & \cos 60° & -\sin 60° \\ 1 & 0 & 0 \end{pmatrix}$$

軌跡の移動動作 12

12.1 法線方向制御

12.1.1 連結の定義(TANG)

予約手順 TANG(...)を使用して、回転軸との間の法線方向連結がスレーブ軸として定義され、2 本のジオメトリ軸がマスタ軸として定義されます。スレーブ軸はマスタ軸の軌跡タンジェントと連続的に調整されます。

注記

連結係数

1 という連結係数は、明示的にプログラムする必要はありません。
接線軸の方向が、連結係数-1 を使用して回転されます。

構文

```
TANG(<スレーブ軸>, <マスタ軸_1>, <マスタ軸_2>, <連結係数>, <座標系>, <調整>)
```

意味

TANG(...):	法線方向連結の定義	
<スレーブ軸>:	スレーブ軸(回転軸)の軸名称	
	データタイプ:	AXIS
	値の範囲:	チャンネル軸名称
<マスタ軸_1> <マスタ軸_2>:	マスタ軸(ジオメトリ軸)の軸名称 1)	
	データタイプ:	AXIS
	値の範囲:	チャネルのジオメトリ軸の名称

<連結係数>:	マスタ軸の軌跡タンジェントの変更のための、スレーブ軸の角度変化の係数 n: 角度変化$_{スレーブ軸}$ = 角度変化$_{軌跡タンジェント}$ * n		
	データタイプ:	REAL	
	初期値:	1.0	
<座標系>:	有効な座標系 [2)]		
	データタイプ:	CHAR	
	値:	"B" :	基本座標系(デフォルト値)
		"W":	ワーク座標系(使用不可)
<調整>:	調整タイプ		
	データタイプ:	CHAR	
	値:	"S":	Standard (デフォルト値) 回転軸のダイナミック応答はマスタ軸に影響を及ぼしません。回転軸のダイナミック応答が実速度追従に必要とされるよりも大きい場合、この方法は十分に正確です。回転軸のダイナミック応答が軌跡タンジェントの変化に従うのに十分な大きさがない場合、回転軸の向きは未定義の丸み付け隙間に沿ったターゲット方向からはずれます。
		"P":	回転軸のダイナミック応答はマスタ軸の軌跡演算で考慮されます。 このために、TANGON()による法線方向連結の起動時に、2 つの追加パラメータを指定してください。 ● 丸み付き隙間 ● 角度許容範囲 セクション 「連結を起動 (TANGON) (ページ 610)」 を参照 **注記** キネマティックトランスフォーメーションの場合は、調整方法「P」の使用をお奨めします。

注 デフォルト値は明示的にプログラムする必要はありません。
[1] **注記** 法線方向連結用のマスタ軸として、機械の初期位置を基準にした機械座標系(MCS)内をプログラムされた軌跡に沿って移動するジオメトリ軸を使用してください。たとえば、旋回ヘッドを持つフライス盤で旋回サイクル CYCLE800 を使用する場合、サイクルがどのように設定されているかに従って、補間が WCS で、たとえばジオメトリ軸 X および Y によって行われます。ただし、法線方向連結は、プログラムされた軌跡に沿って MCS 内で移動するマスタ軸としてのジオメトリ軸で定義してください。このために、機械の**非旋回**条件のジオメトリ軸をマスタ軸として使用してください。
[2] **注記** 基本座標系(BCS)を MCS を基準にして回転させないでください。たとえば、BCS を ROT 命令または旋回サイクル CYCLE800 で回転させると、法線方向制御が正しくなくなります。

12.1.2 挿入ブロックの生成の有効化(TLIFT)

プログラムされた軌跡上の任意の位置でのスレーブ軸のタンジェント変化が、マシンデータ MD37400 $MA_EPS_TLIFT_TANG_STEP でパラメータ設定されている制限値を超えると、それ以降の軌跡演算はコーナでの設定した動作によって決まります。予約手順 TLIFT(...)を使用しない場合、軌跡は TANG(...) (ページ 607)および TANGON(...) (ページ 610)に関連してプログラムされた丸め動作に従って移動されます。

挿入ブロックの生成の有効化

TLIFT(...)が TANG(...)の後にプログラムされている場合、制御装置によって自動的に生成された挿入ブロックが、先読み中にコーナが検出されたときに当該ポイントに挿入されます。

プログラムが実行されると、マスタ軸は挿入ブロックに達したときに停止されます。挿入ブロックで、スレーブ軸は後続のブロックの軌跡タンジェント方向に向かって最大の軸応答性で回転されます。その後、マスタ軸はプログラムされた軌跡上をさらに移動します。

挿入ブロックの生成の無効化

挿入ブロックの生成を無効化するには、TANG(...)を使用して(ただし、その後で TLIFT(...)を使用して挿入ブロックの生成を有効にせずに)法線方向連結をもう一度定義してください。

構文

```
TLIFT(<スレーブ軸>)
```

意味

TLIFT(...):	挿入ブロックの計算によるコーナ検出の有効化	
<スレーブ軸>:	スレーブ軸(回転軸)の軸名称	
	データタイプ:	AXIS
	値の範囲:	チャンネル軸名称

スレーブ軸の回転速度

軌跡軸

法線方向連結が有効になる前にスレーブ軸が軌跡軸としてすでに移動している場合、回転動作は挿入ブロックで軌跡軸として実行されます。

FGREF[<axis>]=0.001 を使用して基準半径を指定する場合、回転動作はパラメータ設定された最大軸速度で実行されます。

MD32000 $MA_MAX_AX_VELO[<スレーブ軸>]

軸の位置決め

法線方向連結が有効になる前にスレーブ軸が軌跡軸としてまだ移動していない場合、回転は挿入ブロックで位置決め軸として実行されます。

回転動作はパラメータ設定された位置決め軸速度で実行されます。

MD32060 $MA_POS_AX_VELO[<スレーブ軸>]

12.1.3 連結を起動(TANGON)

予約手順 TANGON(...)を使用して、TANG(...) (ページ 607)で以前に定義した法線方向連結を有効化します。これで、スレーブ軸はその後の移動中に、軌跡タンジェントと連続的に調整されます。

スレーブ軸の角度

軌跡タンジェントを基準にしたスレーブ軸の角度は、TANG(...)で指定された座標変換比、マシンデータ MD37402 $MA_TANG_OFFSET でパラメータ設定されたオフセット角度、追加で適用される TANGON(...)に対して指定されたオフセット角度によって決まります。

調整「P」

値「P」が法線方向連結の定義(TANG(...))で調整パラメータとして指定されている場合、パラメータ「丸み付け隙間」とオプションでパラメータ「角度許容範囲」を連結の起動時に設定してください。

角度許容範囲に値 0 が指定されている場合、パラメータ「丸み付け隙間」のみが有効になります。

角度許容範囲に 0 よりも大きい値が指定されている場合、有効な丸み付け隙間は、パラメータ設定された丸み付け隙間とパラメータ設定された角度許容範囲に基づいた丸み付け隙間の最小の値になります。

スレーブ軸のダイナミック応答がパラメータ設定された条件に従うのに十分でない場合は、マスタ軸の軌跡速度がそれに応じて小さくなります。

構文

TANGON(<スレーブ軸>, <オフセット角度>, <丸み付け隙間>, <角度許容範囲>)

意味

TANGON(...):	法線方向連結の有効化	
<スレーブ軸>:	スレーブ軸(回転軸)の軸名称	
	データタイプ:	AXIS
	値の範囲:	チャンネル軸名称
<オフセット角度>:	軌跡タンジェントを基準にしたスレーブ軸のオフセット角度 レファレンス点は回転軸の原点になります。	
	データタイプ:	REAL
<丸み付け隙間>:	最大許容丸み付け隙間 丸み付け隙間がダイナミック条件によって大きくなる場合は、マスタ軸の軌跡速度が小さくなります。	
	データタイプ:	REAL
<角度許容範囲>:	スレーブ軸のゼロ設定と軌跡タンジェントの間の指定された角度を基準にした最大許容範囲	
	データタイプ:	REAL

12.1.4 連結の無効化(TANGOF)

予約手順 TANGOF(...)を使用して、TANG(...) (ページ 607)で定義され、TANGON(...) (ページ 610)で有効になった法線方向連結を無効にします。これで、スレーブ軸はマスタ軸の軌跡タンジェントと連続的に調整されなくなります。ただし、スレーブ軸のマスタ軸への連結は無効化後も保持され、これによって、たとえば以下の機能が防止されます。

- 平面変更
- ジオメトリ軸の切り替え
- スレーブ軸の新しい法線方向連結の定義

スレーブ軸のマスタ軸への連結の最終的な取り消しは、連結が TANGDEL(...) (ページ 612)で解除されるまで完了しません。

プログラミング

```
TANGOF(<スレーブ軸>)
```

意味

TANGOF(...):	法線方向連結の無効化	
<スレーブ軸>:	スレーブ軸(回転軸)の軸名称	
	データタイプ:	AXIS
	値の範囲:	チャンネル軸名称

12.1.5 連結の解除(TANGDEL)

TANG(...) (ページ 607)で定義された法線方向連結は、TANGOF(...) (ページ 612)による法線方向連結の無効化後も保持されます。このため、既存の法線方向連結は引き続き、たとえば以下の機能を防止します。

- 平面変更
- ジオメトリ軸の切り替え
- スレーブ軸の新しい法線方向連結の定義

予約手順 TANGDEL(...)を使用して、法線方向連結が TANGOF(...)で無効化された後で、既存の法線方向連結を解除します。

構文

TANGDEL(<スレーブ軸>)

意味

TANGDEL(...):	TANG()で定義された法線結合連結の解除	
	効果:	ノンモーダル
<スレーブ軸>:	法線方向連結を解除するスレーブ軸の名称	
	データタイプ:	AXIS
	値の範囲:	チャンネル軸名称

例

マスタ軸の変更

スレーブ軸に対して新しい法線方向連結を別のマスタ軸で定義するには、まず既存の法線方向連結を解除してください。

プログラムコード	コメント
N10 **TANG**(A, X, Y,1)	; スレーブ軸 A の法線方向連結の定義 A と X および Y 間
N20 TANGON(A)	; スレーブ軸 A の法線方向連結の有効化
N30 X10 Y20	
...	
N80 TANGOF(A)	; スレーブ軸 A の法線方向連結の無効化
N90 **TANGDEL**(A)	; スレーブ軸 A の法線方向連結の解除
...	
N120 **TANG**(A, X, Z)	; スレーブ軸 A の新しい法線方向連結の定義
N130 TANGON(A)	; スレーブ軸 A の新しい法線方向連結の有効化
...	

ジオメトリ軸の切り替え

既存の連結に対してジオメトリ軸の切り替えを実行するには、まず連結を解除してください。

プログラムコード	コメント
N10 GEOAX(2,Y1)	; 2 番目のジオメトリ軸 = 機械軸 Y1
N20 TANG(A, X, Y)	; スレーブ軸 A の法線方向連結の定義
N30 TANGON(A, 90)	; スレーブ軸 A の法線方向連結の有効化
N40 G2 F8000 X0 Y0 I0 J50	; 回転のブロック
N50 TANGOF(A)	; スレーブ軸 A の法線方向連結の無効化
N60 **TANGDEL**(A)	; スレーブ軸 A の法線方向連結の解除

プログラムコード	コメント
N70 **GEOAX**(2, Y2)	; 2 番目のジオメトリ軸 = 機械軸 Y2
N80 TANG(A, X, Y)	; スレーブ軸 A の新しい法線方向連結の定義
N90 TANGON(A, 90)	; スレーブ軸 A の新しい法線方向連結の有効化
...	

12.2 送り速度特性(FNORM、FLIN、FCUB、FPO)

送り速度特性を柔軟に定義できるようにするため、DIN 66025 に従った送り速度指令は、一次特性と 3 次特性により拡張されます。

3 次特性は、直接、または補間スプラインとしてプログラム指令できます。これらの付加特性により、加工するワークの曲率に応じて、連続した滑らかな速度特性をプログラム指令できます。

これらの付加特性により、加工するワークの曲率に応じて、連続した滑らかな速度特性をプログラム指令できます。

構文

```
F… FNORM
F… FLIN
F… FCUB
F=FPO(…,…,…)
```

意味

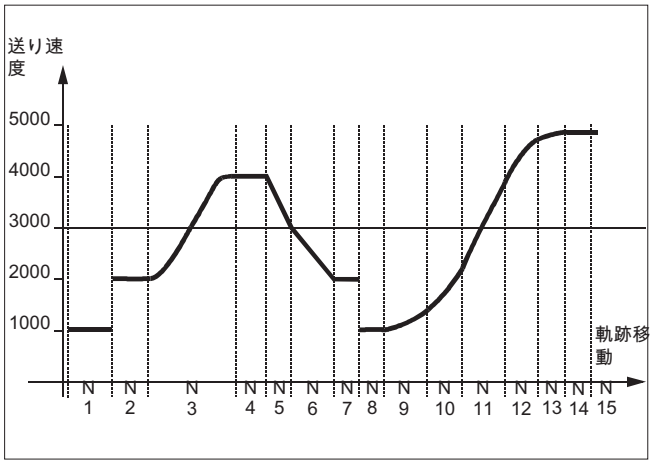
FNORM:	初期設定です。この送り速度値はブロックの移動軌跡に応じて指定され、その後、モーダル値として有効になります。
FLIN:	軌跡速度パターン **1 次**: 送り速度値は、ブロック始点の現在値からブロック終点へ、移動軌跡を通って直線的にアプローチして、その後、モーダル値として有効になります。この動作は G93 および G94 との組み合わせが可能です。
FCUB:	軌跡速度パターン **3 次:** ブロック毎にプログラム指令した F 値 (ブロック終点に対する)をスプラインで接続します。スプラインは、その前後に定義した送り速度で接線方向に開始されて終了し、G93 と G94 で有効になります。 F アドレスがブロックにない場合は、最後にプログラム指令した F 値を使用します。
F=FPO…:	**多項式**の軌跡速度パターン: F アドレスは、現在の値からブロック終了値まで、多項式で送り速度特性を定義します。終了値はそれ以降、モーダル値として有効となります。

曲線の軌跡区間の送り速度調整

送り速度多項式 F=FPO とスプライン送り速度 FCUB は常に、一定切削速度 CFC で移動して、加々速度一定の指令送り速度パターンを生成します。これにより、連続した加減速指令送り速度パターンが作成されます。

例:さまざまな送り速度パターン

次の例は、さまざまな送り速度パターンのプログラミングと図を示します。



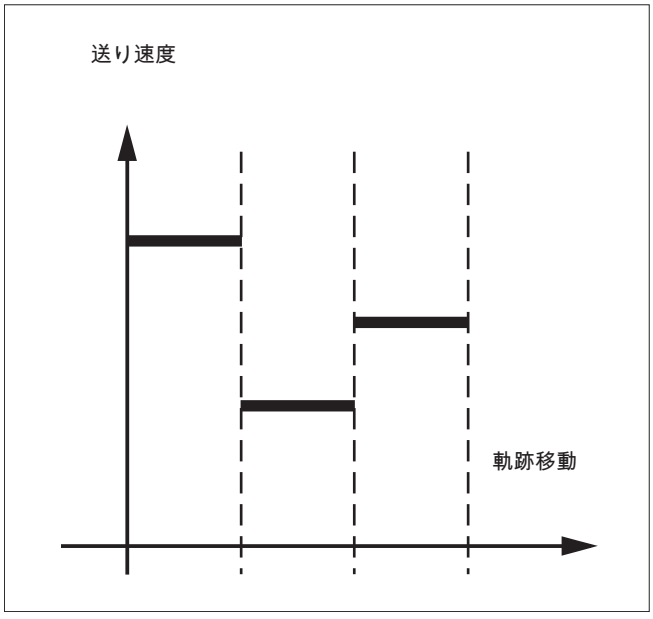
プログラムコード	コメント
N1 F1000 FNORM G1 X8 G91 G64	; 一定送り速度パターン、インクレメンタル指令データ
N2 F2000 X7	; 指令速度のステップ変更
N3 F=FPO(4000, 6000, -4000)	; ブロック終点での送り速度が 4000 となる多項式による、送り速度パターン
N4 X6	; 多項式の送り速度の 4000 がモーダル値として有効です
N5 F3000 FLIN X5	; 直線のの送り速度パターン
N6 F2000 X8	; 直線のの送り速度パターン
N7 X5	; 直線の送り速度パターンがモーダル値として有効です
N8 F1000 FNORM X5	; 加減速ステップ変更による一定送り速度パターン
N9 F1400 FCUB X8	; 複数ブロックでプログラム指令した次のすべての F 値をスプラインで接続します
N10 F2200 X6	
N11 F3900 X7	
N12 F4600 X7	
N13 F4900 X5	; スプラインパターンを解除します
N14 FNORM X5	
N15 X20	

詳細情報

FNORM

送りアドレス F は、DIN 66025 に従い、一定の値として軌跡送り速度を定義します。

これに関して詳しくは、『プログラミングマニュアル 基本編』を参照してください。

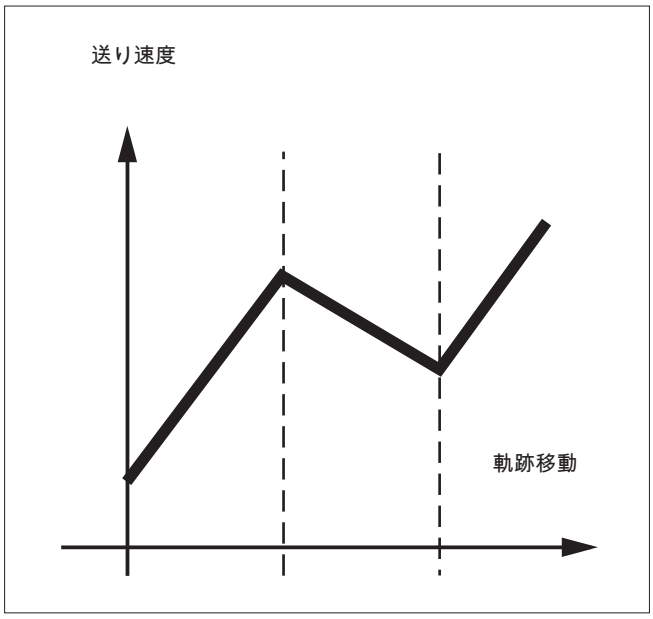


FLIN

送り速度特性は、現在の送り速度値から、プログラム指令した F 値へと、ブロック終点まで一次特性でアプローチします。

例:

```
N30 F1400 FLIN X50
```

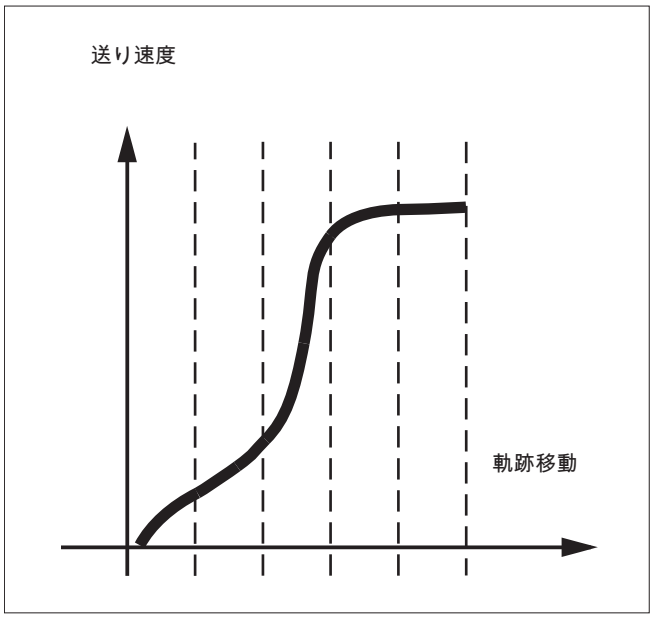


FCUB

送り速度は、現在の送り速度値から、プログラム指令した F 値へと、ブロック終点まで 3 次特性に従ってアプローチします。制御装置はスプラインを使用して、動作中の FCUB で、ノンモーダルでプログラム指令されたすべての送り速度値を接続します。送り速度値はここで、スプライン補間を計算するための補間点として機能します。

例:

```
N50 F1400 FCUB X50
N60 F2000 X47
N70 F3800 X52
```



F=FPO(…,…,…)

送り速度特性は、多項式で直接プログラム指令します。多項式係数は、多項式補間で使用するものと同じ方式で指定します。

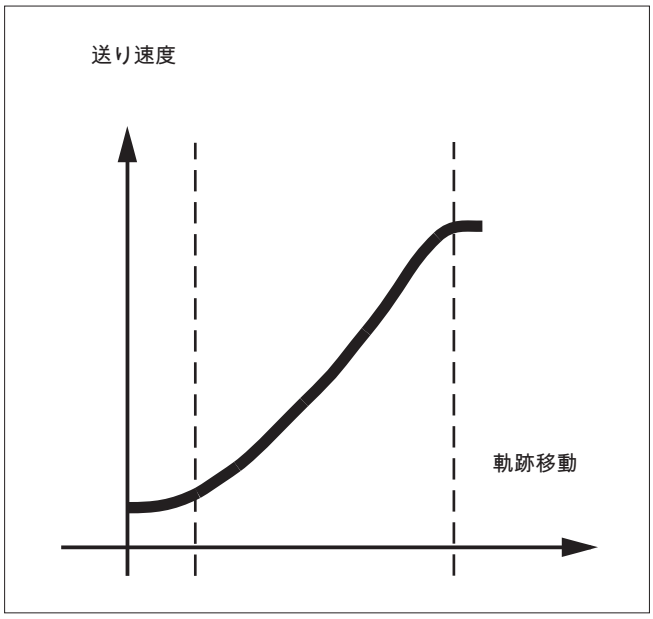
例:

```
F=FPO(endfeed, quadf, cubf)
```

endfeed、**quadf**、および **cubf** は、あらかじめ定義される変数です。

endfeed:	ブロック終点の送り速度
quadf:	2 次多項式係数
cubf:	3 次多項式係数

FCUB が有効な場合は、ブロック始点とブロック終点で FPO を使用して定義した特性へ、スプラインによって接線方向につなぎます。



必要条件

- 軌跡移動特性のプログラミングの機能は、プログラム指令した送り速度特性にかかわらず、適用されます。
- プログラム指令した送り速度特性は、G90 と G91 のいずれにも影響されず、常に絶対値です。
- 送り速度特性の曲線 FLIN と FCUB は、G93 と G94 で動作し、 G95、 G96/G961、および G97/G971 では動作し**ません**。
- コンプレッサ COMPON が動作中で、複数のブロックを結合してスプラインセグメントを作成する場合、以下が適用されます。

FNORM:	グループの最後のブロックの F ワードがスプラインセグメントに適用されます。
FLIN:	グループの最後のブロックの F ワードがスプラインセグメントに適用されます。 プログラム指令した F 値がセグメントの最後まで適用され、その後、直線的にアプローチします。
FCUB:	生成した送り速度スプラインは、プログラム指令終点から、マシンデータ MD20172 $MC_COMPESS_VELO_TOL で設定した値を超えない距離だけの誤差があります。
F=FPO(…,…,…):	これらのブロックは圧縮されません。

12.3 加減速動作

12.3.1 加減速モード(BRISK、BRISKA、SOFT、SOFTA、DRIVE、DRIVEA)

次のパートプログラム命令を使用して、現在の加減速モードをプログラム指令できます。

- "BRISK、BRISKA"
  単独軸または軌跡軸が、プログラム指令送り速度に達するまで最大加減速度で移動します(**加々速度制限のない加減速**)。
- "SOFT、SOFTA"
  単独軸または軌跡軸が、プログラム指令送り速度に達するまで一定の加減速度で移動します(**加々速度一定加減速**)。
- "DRIVE、DRIVEA"
  単独軸または軌跡軸が、プログラム指令制限速度まで最大加減速度で移動します(マシンデータの設定)。その後、加速度が、プログラム指令送り速度に達するまで減少します(マシンデータの設定)。

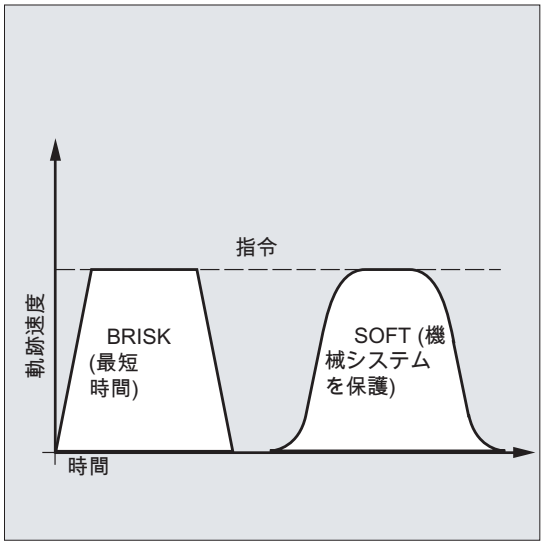


図 12-1 BRISK と SOFT による軌跡速度曲線

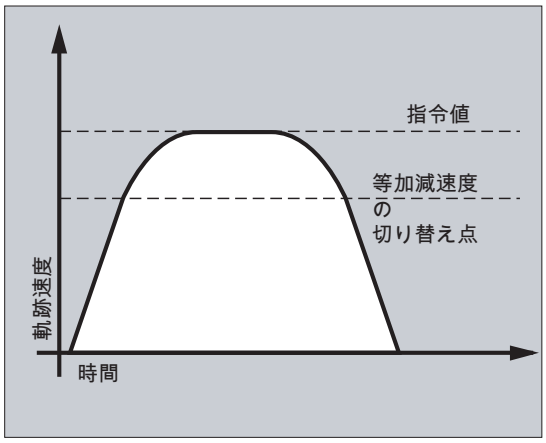


図 12-2　DRIVE による軌跡速度曲線

構文

```
BRISK
BRISKA(<軸 1>,<軸 2>,…)
SOFT
SOFTA(<軸 1>,<軸 2>,…)
DRIVE
DRIVEA(<軸 1>,<軸 2>,…)
```

意味

BRISK:	軌跡軸に対して「加々速度制限のない加減速」を有効にする命令です。
BRISKA:	単独軸移動に対して「加々速度制限のない加減速」を有効にする命令です(JOG、JOG/INC、位置決め軸、揺動軸など)。
SOFT:	軌跡軸に対して「加々速度一定加減速」を有効にする命令です。
SOFTA:	単独軸移動(JOG、JOG/INC、位置決め軸、揺動軸など)に対して「加々速度一定加減速」を有効にする命令です。
DRIVE:	軌跡軸に対して、設定制限速度 (MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT)以上で加減速低減を有効にする命令です。

DRIVEA:	単独軸移動(JOG、JOG/INC、位置決め軸、揺動軸など)に対して、設定制限速度(MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT)以上で加減速低減を有効にする命令です。
(<軸 1>,<軸 2>, ...):	呼び出した加減速モードが適用される単独軸です。

必要条件

加工中の加減速モードの変更

加工中にパートプログラムで加減速モードを変更(BRISK ↔ SOFT)すると、連続軌跡モードであっても、遷移中にブロックの終点でイグザクトストップによりブロックが切り替えられます。

例

例 1:SOFT と BRISKA

プログラムコード

```
N10 G1 X… Y… F900 SOFT
N20 BRISKA(AX5,AX6)
...
```

例 2 :DRIVE と DRIVEA

プログラムコード

```
N05 DRIVE
N10 G1 X… Y… F1000
N20 DRIVEA (AX4, AX6)
...
```

参照先

機能マニュアル基本機能; 加減速制御(B2)

12.3.2 スレーブ軸に対する加減速の動作(VELOLIMA、ACCLIMA、JERKLIMA)

軸連結の場合(法線方向制御の補正、連結移動、軸間連動機能、電子ギヤ;「軸連結 (ページ 661)」を参照してください)。

スレーブ軸/主軸の応答性制限は、軸連結がすでに有効な場合でも、パートプログラムまたはシンクロナイズドアクションから VELOLIMA、ACCLIMA、および JERKLIMA 機能を使用して操作できます。

注記

JERKLIMA 機能は、一部のタイプの連結では使用できません。

参照先:

- 機能マニュアル応用機能; 軸の連結(M3)
- 機能マニュアル上級機能; 主軸同期(S3)

注記

SINUMERIK 828D での適用

VELOLIMA、ACCLIMA、JERKLIMA 機能は、SINUMERIK 828D で「連結移動」機能と組み合わせた場合のみ使用することができます。

構文

```
VELOLIMA(<軸>)=<値>
ACCLIMA(<軸>)=<値>
JERKLIMA(<軸>)=<値>
```

意味

VELOLIMA:	最大パラメータ設定**速度**を補正する命令
ACCLIMA:	最大パラメータ設定**加減速度**を補正する命令
JERKLIMA:	最大パラメータ設定**加々速度**を補正する命令
<軸>:	応答性制限の補正が必要なスレーブ軸
<値>:	補正値の割合(%)

例

例 1:スレーブ軸(AX4)の応答性制限の補正

プログラムコード	コメント
...	
VELOLIMA[AX4]=75	; マシンデータに設定された最大軸速度の 75%の制限へ補正
ACCLIMA[AX4]=50	; マシンデータに設定された最大軸加減速度の 50%の制限へ補正
JERKLIMA[AX4]=50	; マシンデータに設定された最大軸加々速度の 50%の制限へ補正
...	

例 2 :電子ギヤ

軸 4 を、「電子ギヤ」連結により軸 X に連結します。スレーブ軸の加減速能力を最大加減速度の 70%に制限します。最大許容速度を最大速度の 50%に制限します。連結が正常に動作すると、最大許容速度が 100%に復帰します。

プログラムコード	コメント
...	
N120 ACCLIMA[AX4]=70	; 最大加減速度の制限
N130 VELOLIMA[AX4]=50	; 最大速度の制限
...	
N150 EGON(AX4,"FINE",X,1,2)	; EG 連結の適用
...	
N200 VELOLIMA[AX4]=100	; 最大速度へ復帰
...	

例 3 :内部的なシンクロナイズドアクションによる軸間連動機能への効果

軸間連動機能により軸 4 が X に連結されます。加減速動作は、100%の値から、内部的なシンクロナイズドアクション 2 により、80%の値に制限されます。

プログラムコード	コメント
...	
N120 IDS=2 WHENEVER $AA_IM[AX4] > 100 DO ACCLIMA[AX4]=80	; シンクロナイズドアクション
N130 LEADON(AX4, X, 2)	; 軸間連動機能オン
...	

12.3.3 テクノロジ用のダイナミック応答値の起動(DYNNORM、DYNPOS、DYNROUGH、DYNSEMIFIN、DYNFINISH)

「テクノロジ」G グループを使用して、当該のダイナミック応答を、5 種類の異なるテクノロジ加工ステップに対して有効にできます。

ダイナミック応答値と G 命令が設定できます。したがって、これらはマシンデータ設定に依存します(→ 工作機械メーカ)。

参照先:
機能マニュアル 基本機能; 連続軌跡モード、イグザクトストップと先読み(B1)

構文

ダイナミック応答値の起動:
DYNNORM
DYNPOS

```
DYNROUGH
DYNSEMIFIN
DYNFINISH
```

注記

ダイナミック応答値はすでに、関連する G 命令がプログラム指令されたブロックのなかで有効です。加工は停止されません。

特定のフィールド要素の読み取りまたは書き込み:
R<m>=$MA...[n,X]
$MA...[n,X]=<値>

意味

DYNNORM:	**標準のダイナミック応答**を起動する G 命令	
DYNPOS:	**位置決めモード、タッピング**のダイナミック応答を起動する G 命令	
DYNROUGH:	**荒削り**のダイナミック応答を起動する G 命令	
DYNSEMIFIN:	**仕上げ**のダイナミック応答を起動する G 命令	
DYNFINISH:	**滑らかな仕上げ**のダイナミック応答を起動する G 命令	
R<m>:	番号<m>をもつ R 変数	
$MA...[n,X]:	ダイナミック応答に作用するフィールド要素をもつマシンデータ	
<n>:	配列インデックス	
	値の範囲:	0 ... 4
	0	標準のダイナミック応答(DYNNORM)
	1	位置決めモードのダイナミック応答(DYNPOS)
	2	荒削りのダイナミック応答(DYNROUGH)
	3	仕上げのダイナミック応答(DYNSEMIFIN)
	4	滑らかな仕上げのダイナミック応答(DYNFINISH)
<X> :	軸アドレス	
<値>:	ダイナミック応答値	

例

例 1:ダイナミック応答値の起動

プログラムコード	コメント
DYNNORM G1 X10	; 初期設定
DYNPOS G1 X10 Y20 Z30 F…	; 位置決めモード、タッピング
DYNROUGH G1 X10 Y20 Z30 F10000	; 荒削り
DYNSEMIFIN G1 X10 Y20 Z30 F2000	; 仕上げ
DYNFINISH G1 X10 Y20 Z30 F1000	; 滑らかな仕上げ

例 2 :指定フィールド要素の読み取りまたは書き込み

荒削りの最大加減速度、X 軸

プログラムコード	コメント
R1=$MA_MAX_AX_ACCEL[2,X]	; 読み取り
$MA_MAX_AX_ACCEL[2,X]=5	; 書き込み

12.4 フィードフォワード制御による移動(FFWON、FFWOF)

フィードフォワード制御は、経路誤差ゼロに向かって輪郭切削をする際の速度による行き過ぎを低減します。フィードフォワード制御の移動により、高精度の軌跡が可能になり、加工レベルが向上します。

構文

```
FFWON
FFWOF
```

意味

FFWON:	フィードフォワード制御を**有効にする**命令
FFWOF:	フィードフォワード制御を**解除する**命令

注記

フィードフォワード制御のタイプと、フィードフォワード制御で移動する軌跡軸は、マシンデータで指定します。

初期設定:速度のフィードフォワード制御

オプション:加減速度のフィードフォワード制御

例

プログラムコード

```
N10 FFWON
N20 G1 X… Y… F900 SOFT
```

12.5 プログラム指令可能な輪郭精度(CPRECON、CPRECOF)

「プログラム指令可能な輪郭精度」機能は、速度の自動調整により、輪郭の曲線部での軌跡エラーを低減します。

維持される輪郭精度は、機械の構成に従って(MD20470 $MC_MC_CPREC_WITH_FFW; 工作機械メーカの仕様書を参照してください)、セッティングデータ$SC_CONTPREC またはプログラム指令輪郭許容範囲 CTOL のいずれかによって指定されます。値が小さいほど、またジオメトリ軸の K_V 係数が小さいほど、輪郭の曲線部での軌跡送り速度は大きく低減されます。

「プログラム指令可能な輪郭精度」機能を、NC プログラムで CPRECON と CPRECOF 命令によって有効または無効にします。

構文

```
CPRECON
...
CPRECOF
```

意味

CPRECON:	G 命令呼び出し:「プログラム指令可能な輪郭精度」を**オン**に切り替え	
	効果:	モーダル
CPRECOF:	G 命令呼び出し:「プログラム指令可能な輪郭精度」を**オフ**に切り替え	
	効果:	モーダル

CPRECON と CPRECOF を合わせて G 機能グループ 39 を形成します(プログラム指令可能な輪郭精度)。

注記

セッティングデータ$SC_MINFEED (CPRECON での最小軌跡送り速度)により、軌跡送り速度の最低速度を指定できます。

より低い F 値がプログラム指令されている場合または軸のダイナミック制限によってより低い軌跡速度が必要な場合を除いて、送り速度がこの値を下回って制限されることはありません。

注記

「プログラム指令可能な輪郭精度」機能は、軌跡のジオメトリ軸のみを考慮します。位置決め軸の速度には影響しません。

例

プログラムコード	コメント
N10 G0 X0 Y0	
N20 CPRECON	; プログラマブル輪郭精度を有効にします
N30 G1 G64 X100 F10000	; 連続軌跡モードで 10 m/min で加工します。
N40 G3 Y20 J10	; 円弧ブロックの送り速度を自動制限します。
N50 G1 X0	; 再度制限なしの送り速度(10 m/min)で加工します。
...	
N100 CPRECOF	;「プログラム指令可能な輪郭精度」を解除します。
N110 G0 ...	

参照先

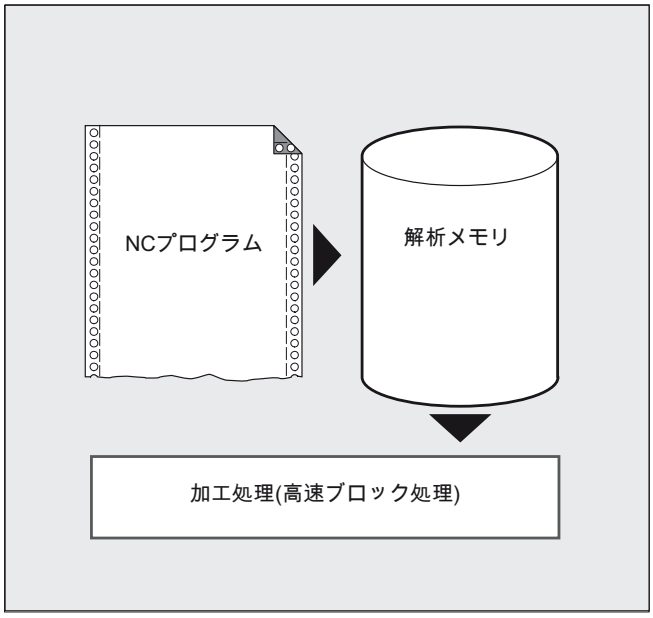
CTOL のプログラム指令については、「輪郭/旋回の許容範囲のプログラミング(CTOL、OTOL、ATOL) (ページ 653)」を参照してください。

「プログラム指令可能な輪郭精度」機能について詳しくは、以下を参照してください。

機能マニュアル 応用機能; 軌跡誤差監視(K6)、「プログラム指令可能な輪郭精度」の章

12.6 解析メモリによるプログラム手順(STOPFIFO、STARTFIFO、FIFOCTRL、STOPRE)

コントローラは、拡張レベルに応じて一定の量の解析メモリを備えています。ここに、解析されたブロックがプログラムの実行前に格納され、その後、加工の間に高速ブロック処理として出力されます。これらの処理により、短い軌跡の集合を高速で移動することができます。制御時間が十分に残っていれば、解析メモリは常にいっぱいです。



加工ステップの指定

解析メモリに保持される加工ステップの開始と終了はそれぞれ、パートプログラムで「STOPFIFO」と「STARTFIFO」を使用して指定します。解析して保持したブロックの処理が開始されるのは、「STARTFIFO」命令の後、または解析メモリがいっぱいの場合のみです。

解析メモリの自動制御

解析メモリの自動制御は、「FIFOCTRL」命令で呼び出します。「FIFOCTRL」は最初は「STOPFIFO」とまったく同様に機能します。プログラミングにかかわらず、解析メモリがいっぱいになるまで、処理は開始されません。ただし、解析メモリに空きがある場合の動作は異なります。「FIFOCTRL」を使用した場合は、メモリの使用率が 2/3 に達すると、完全に空になって停止状態まで減速することのないよう、軌跡速度がだんだん減速します。

先読み停止

「STOPRE」命令をブロックでプログラミングすると、ブロックの解析と一時的な記憶を停止します。次のブロックは、先読みがおこなわれて保持されたすべてのブロックが完

全に実行されるまで実行されません。先行ブロックは (G9 と同様に) イグザクトストップで停止します。

通知
プログラム中止 工具補正またはスプライン補間が有効な場合は、「STOPRE」命令をプログラム指令しないでください。プログラム指令すると、連続ブロック処理が中断されます。

構文

表 12-1　加工ステップの指定:

```
STOPFIFO
...
STARTFIFO
```

表 12-2　解析メモリの自動制御:

```
...
FIFOCTRL
...
```

表 12-3　先読み停止:

```
...
STOPRE
...
```

注記

「STOPFIFO」、「STARTFIFO」、「FIFOCTRL」、および「STOPRE」の命令は、それぞれ個別のブロックにプログラム指令してください。

意味

STOPFIFO:	「STOPFIFO」は、解析メモリに保持する加工ステップ開始を指定します。「STOPFIFO」は処理を停止し、以下の時点まで、解析メモリをいっぱいにします。 ● 「STARTFIFO」または「STOPRE」を認識する または ● 解析メモリがいっぱいになる または ● プログラム終点に達する
STARTFIFO:	「STARTFIFO」は、加工ステップの高速処理を開始します。これと並行して、解析メモリも満たされます。
FIFOCTRL:	解析メモリの自動制御を適用します
STOPRE:	解析を停止します

注記

バッファリングが不必要な動作(基準点のサーチ、計測機能、など)の命令が加工ステップに含まれる場合は、解析メモリがいっぱいにならないか、またはいっぱいになる前に処理を中断します。

注記

コントローラは、状態データ($SA...)にアクセスした場合、内部の先読み停止をおこないます。

例:解析を停止します

プログラムコード	コメント
...	
N30 MEAW=1 G1 F1000 X100 Y100 Z50	; 1 番目の計測点でのプローブによる計測ブロックと直線補間です。
N40 STOPRE	; 先読み停止
...	

12.7 停止遅延範囲(DELAYFSTON、DELAYFSTOF)の定義

パートプログラムで条件付き中断可能範囲を定義するために、事前定義された DELAYFSTON および DELAYFSTOF 手順を使用します(停止遅延範囲)。

注記

DELAYFSTON および DELAYFSTOF はシンクロナイズドアクションでは**許容されません!**

構文

```
DELAYFSTON
...
DELAYFSTOF
```

意味

DELAYFSTON:	停止遅延範囲の先頭の定義	
	単独ブロック指令:	あり
DELAYFSTOF:	停止遅延領域の終了を定義します	
	単独ブロック指令:	あり

プログラミング例

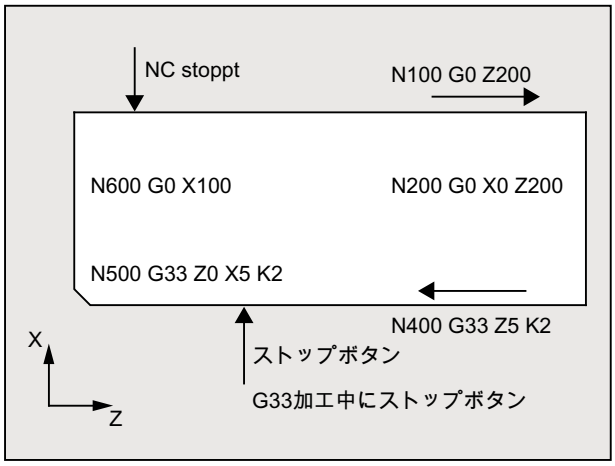
次のプログラムブロックは以下のループで繰り返されます。

プログラムコード

```
...
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
```

プログラムコード
...

次の図では、ユーザーが停止遅延範囲内で「停止」を押すと、停止遅延範囲(つまりブロック N100)の外側で NC が減速を開始することがわかります。これにより、NC が N100 の始点で停止します。



他の情報

サブプログラム終了

DELAYFSTOF は、DELAYFSTON を呼び出しているサブプログラムの終点で、自動的に有効になります。

ネスト

停止遅延領域でサブプログラム 1 がサブプログラム 2 を呼び出した場合は、サブプログラム 2 全体が停止遅延領域になります。特に、DELAYFSTOF はサブプログラム 2 では無効です。

例:

プログラムコード	コメント
N10010 **DELAYFSTON**	; N10xxx プログラムレベル 1 のブロックです。
N10020 R1 = R1 + 1	
N10030 G4 F1	; 停止遅延領域を開始します。
...	
N10040 subprogram2	
...	
...	; サブプログラム 2 を解釈します。
N20010 **DELAYFSTON**	; 無効です、停止遅延開始を繰り返します。2 番目のレベルです。
...	

プログラムコード	コメント
N20020 **DELAYFSTOF**	; 無効です、別のレベルで終了しています。
N20030 RET	
N10050 **DELAYFSTOF**	; 停止遅延範囲を同じレベルで終了します。
...	
N10060 R2 = R2 + 2	
N10070 G4 F1	; 停止遅延領域が終了します。これ以降は、動作を直ちに停止します。

システム変数

パートプログラム処理が現在、停止遅延領域にあるかどうかを特定するために、以下のシステム変数を確認することができます。

- パートプログラムでは、$P_DELAYFST
- シンクロナイズドアクションでは$AC_DELAYFST

値	意味
0	停止遅延範囲が無効です
1	停止遅延領域が有効です

12.8 SERUPRO のプログラム位置のマスク(IPTRLOCK、IPTRUNLOCK)

機械のメカニズムが複雑である事例では、停止ブロックサーチの SERUPRO が必要となる場合があります。

プログラム指令可能な中断ポインタを使用すると、「中断点の検索」を使用して検索できない点の直前に介入できます。

また、パートプログラム区間に検索できない区間を定義することもできます。その区間では NC は再介入できなくなります。プログラムが中断すると、NC は、処理した最後のブロックを記録して、それを HMI 操作画面から検索することができます。

構文

```
IPTRLOCK
IPTRUNLOCK
```

これらの命令はパートプログラム行にあり、プログラム指令可能な中断ポインタの使用を可能にします。

意味

IPTRLOCK:	検索不可プログラム区間の開始
IPTRUNLOCK:	検索不可プログラム区間の終了

両方の命令はパートプログラムでのみ使用でき、シンクロナイズドアクションでは**使用できません**。

例

自動的な IPTRUNLOCK による 2 つのプログラムレベルでの検索不可プログラム区間の入れ子指令。サブプログラム 1 の自動的な「IPTRUNLOCK」が検索不可区間を終了します。

プログラムコード	コメント
N10010 IPTRLOCK()	
N10020 R1 = R1 + 1	
N10030 G4 F1	; 検索をマスクするプログラム区間の保持ブロックを起動します。
...	
N10040 subprogram2	
...	; サブプログラム 2 を解釈します。
N20010 IPTRLOCK ()	; 無効です、起動を繰り返します。
...	
N20020 IPTRUNLOCK ()	; 無効です、別のレベルで終了しています。

プログラムコード	コメント
N20030 RET	
...	
N10060 R2 = R2 + 2	
N10070 RET	; サーチを抑制するプログラム区間の終了
N100 G4 F2	; メインプログラムを続行します。

ここで、中断ポインタが再度、100 で中断をおこないます。

詳細情報

検索不可区間の認識と検出

検索できないプログラム区間は、言語命令「IPTRLOCK」と「IPTRUNLOCK」で識別されます。

命令「IPTRLOCK」は、メインランの実行可能なシングルブロックで中断ポインタを解除します。(SBL1)これ以降では、このブロックを保持ブロックと呼びます。「IPTRLOCK」の後でプログラムが中止された場合は、この保持ブロックを HMI 操作画面から検索できます。

実行中のブロックからの続行

中断ポインタは「IPTRUNLOCK」で、実行中のブロックに、その後に続くプログラム区間の中断点として配置されます。

検索ターゲットが見つかると、新しい検索ターゲットを保持ブロックによって繰り返すことができます。

ユーザーが編集した中断ポインタは、HMI で再度削除してください。

入れ子の規則:

入れ子指令の言語命令「IPTRLOCK」と「IPTRUNLOCK」と、サブプログラム終了との間の連携には、次の点が適用されます。

1. 「IPTRUNLOCK」は、「IPTRLOCK」が呼び出されるサブプログラムの最後に自動的に起動されます。
2. 「IPTRLOCK」は、サーチ制御区間では無効です。
3. 検索不可区間でサブプログラム 1 がサブプログラム 2 を呼び出した場合は、サブプログラム 2 全体が検索不可の状態になります。特に「IPTRUNLOCK」は、サブプログラム 2 では効果がありません。

詳しくは、
『/FB1/ 機能マニュアル 基本機能; モードグループ、チャネル、プログラム運転モード (K1)』を参照してください。

システム変数

検索不可区間は、パートプログラムで「$P_IPTRLOCK」を使用して検出できます。

自動中断ポインタ

自動中断ポインタは、以前に定義した連結タイプを検索不可として自動的に定義します。使用するマシンデータは、次のとおりです。

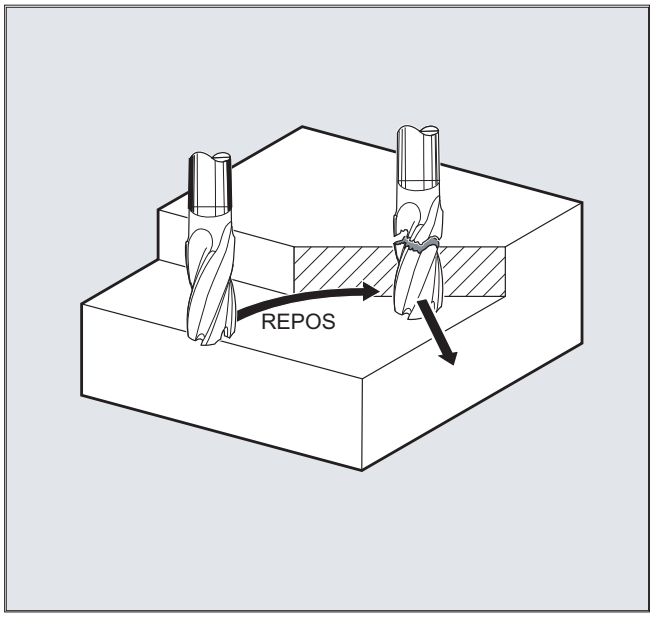
- 「EGON」の電子ギヤ
- 「LEADON」の軸間連動機能

上記を使用すると、自動中断ポインタを起動します。プログラム指令中断ポインタ、および自動中断ポインタで起動した中断ポインタが重複する場合は、最大限の検索不可区間が生成されます。

12.9 輪郭への再位置決め(REPOSA、REPOSL、REPOSQ、REPOSQA、REPOSH、REPOSHA、DISR、DISPR、RMIBL、RMBBL、RMEBL、RMNBL)

工具の破損、またはワークを計測するためなどの理由で、プログラムの実行を中断して、加工運転中に工具を退避した場合は、プログラムが制御している輪郭上の任意の点を選択して、再度位置決めできます。

REPOS 命令は、ASUB でサブプログラム戻り(M17 など)として機能します。次のブロックは実行されません。プログラム実行の中断については、「割り込みルーチン(ASUB) (ページ 147)」も参照してください。



構文

```
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=…
```

意味

アプローチ軌跡の選択

REPOSA:	直線上の輪郭へのジオメトリ軸による再位置決め。 その他のすべての軸も再位置決めされます。
REPOSL:	直線上の輪郭へのジオメトリ軸による再位置決め。 その他の軸は明示的にプログラム指令してください。
REPOSQ DISR=… :	半径 DISR の 4 分円に沿ったジオメトリ軸による輪郭への再位置決め。 その他の軸は明示的にプログラム指令してください。
REPOSQA DISR=… :	半径 DISR の 4 分円に沿ったジオメトリ軸による輪郭への再位置決め。 その他のすべての軸も再位置決めされます。
REPOSH DISR=… :	直径 DISR の半円に沿ったジオメトリ軸による輪郭への再位置決め。 その他の軸は明示的にプログラム指令してください。
REPOSHA DISR=… :	半径 DISR の半円に沿ったジオメトリ軸による輪郭への再位置決め。 その他のすべての軸も再位置決めされます。

再位置決め点の選択

RMIBL:	中断点へアプローチします
RMIBL DISPR=…:	中断点の手前からの距離 DISPR での開始点(mm/inch)
RMBBL:	ブロックの始点へアプローチします
RMEBL:	ブロックの終点へアプローチします
RMEBL DISPR=… :	終点の手前からの距離が DISPR のブロックの終点へアプローチします
RMNBL:	最も近い軌跡点へアプローチします
A0 B0 C0 :	アプローチがおこなわれる軸です

注記

互換性

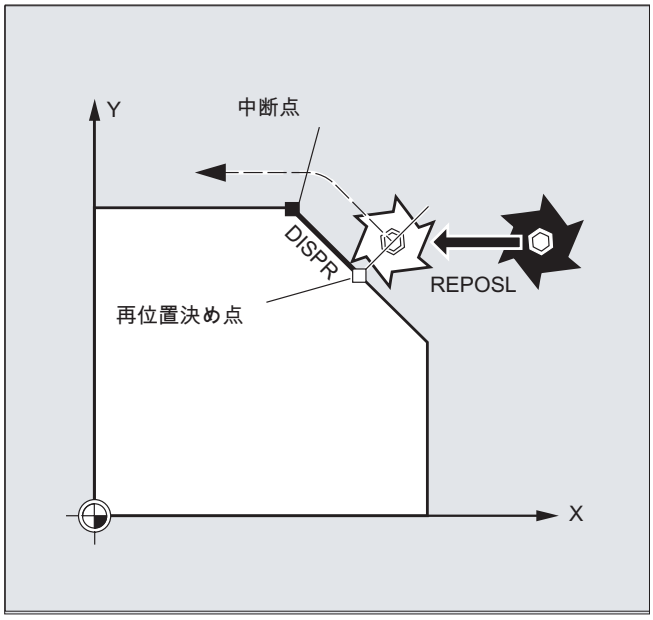
旧版のソフトウェアとの互換性を残すために、モーダル G 命令 RMI、RMB、RME、および RMN で REPOS アプローチモードをプログラム指令することもできます。ASUB 内で使用する場合は、PROC 命令でこれに SAVE 属性を割り当ててください。これをおこなわない場合、ASUB で使用されるモーダル REPOS アプローチモードが、予め設定されている RMI と異なっている場合に、その後の REPOS 処理でも有効になります。

直線上の輪郭への再位置決め REPOSA、REPOSL

工具は、直線上を再位置決め点へアプローチします。

例

```
REPOSL RMIBL DISPR=6 F400
```

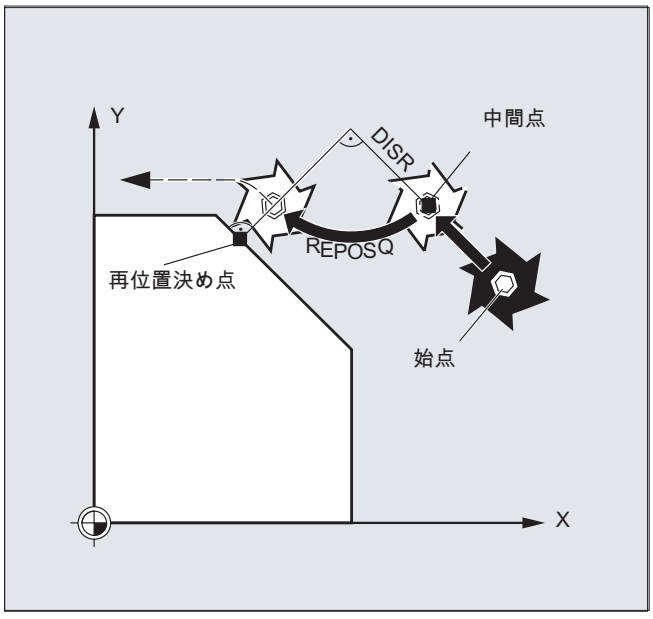


4 分円に沿った輪郭への再位置決め、REPOSQ、REPOSQA

工具は、半径 DISR=…の 4 分円に沿って再位置決め点へアプローチします。制御装置は自動的に、始点と再位置決め点の間の必要な中間点を計算します。

例

```
REPOSQ RMIBL DISR=10 F400
```

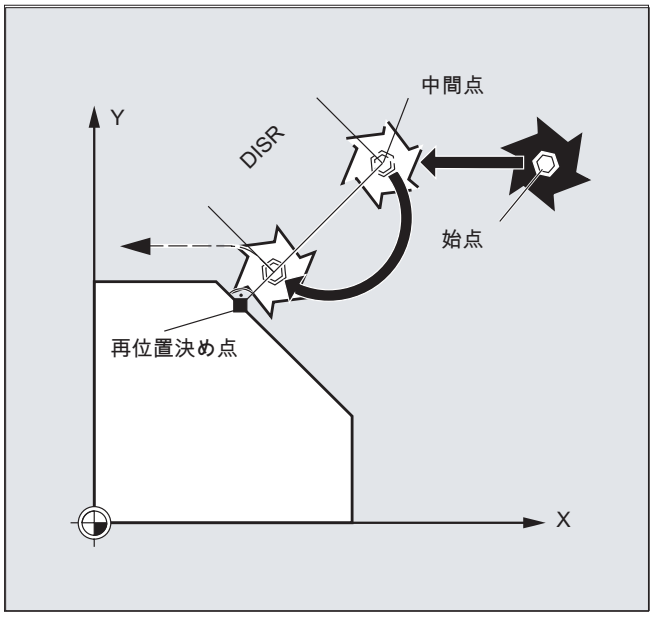


半円に沿った輪郭への再位置決め、REPOSH、REPOSHA

工具は、直径 DISR=…の半円上の再位置決め点へアプローチします。制御装置は自動的に、始点と再位置決め点の間の必要な中間点を計算します。

例

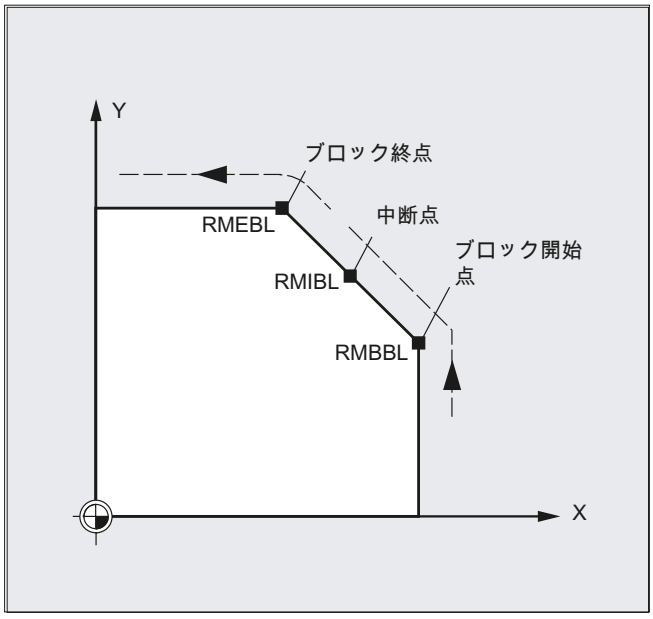
```
REPOSH RMIBL DISR=20 F400
```



再位置決め点の指定(RMNBL による SERUPRO アプローチではない場合)

プログラムの実行が中断した NC ブロックに対して、3 つの異なる再位置決め点から 1 つを選択できます。

- RMIBL、中断点
- RMBBL、ブロックの始点または最後の終点
- RMEBL、ブロックの終点



RMIBL DISPR=…または RME DISPR=…を使用すると、中断点またはブロックの終点の手前にある再位置決め点を選択できます。

DISPR=…を使用すると、再位置決め点と、終点の手前の中断点の間の輪郭距離(mm/inch)を指令できます。値が大きい場合でも、この点が、ブロックの始点を越えることはありません。

DISPR=…命令をプログラム指令しない場合は、DISPR=0 が適用され、これにより、(RMIBL による)中断点または(RMEBL による)ブロックの終点が適用されます。

DISPR 符号

DISPR の符号を評価します。正符号の場合は、以前と同様に動作します。

負符号の場合は、中断点の後ろに、または、RMBBL を使用している場合はブロックの始点の後ろにアプローチします。

中断点とアプローチ点の間の距離は、DISPR の値により異なります。値が大きい場合でも、この点が、ブロックの終点を越えることはありません。

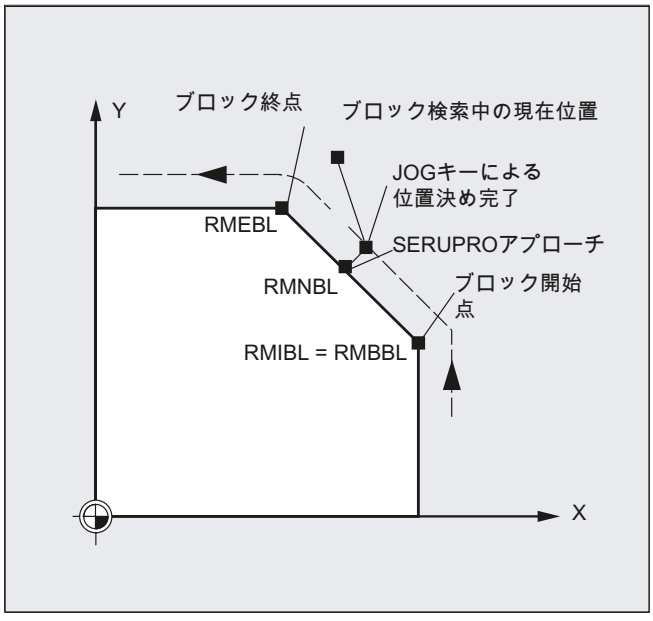
用途例：

クランプへアプローチすると、センサにより認識されす。ASUB が開始され、クランプをバイパスします。

その後、負の DISPR を、クランプの後ろの 1 点に再位置決めし、プログラムを続行します。

RMNBL による SERUPRO アプローチ

加工中に任意の位置で強制的に中止された場合は、SERUPRO アプローチと RMNBL で中止点からの最短軌跡でアプローチし、その後に処理されるのは残移動距離のみとなります。ユーザーは中断ブロックで SERUPRO 処理を開始し、JOG キーを使用して、ターゲットブロックの問題の場所の手前に移動します。



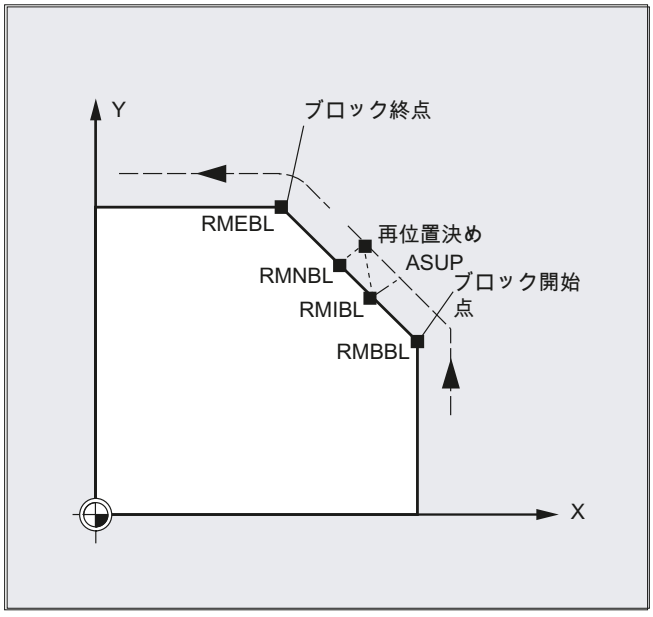
注記

SERUPRO

SERUPRO に対して、RMIBL と RMBBL は同じです。RMNBL は SERUPRO 機能に制限されることなく、一般的に有効です。

最も近い軌跡点 RMNBL からのアプローチ

REPOSA を解釈するとき、RMNBL による再位置決めブロック全体は中断後に再起動されませんが、残移動距離のみが処理されます。中断したブロックに最も近い軌跡点へアプローチします。



有効な REPOS モードの状態

中断されたブロックの有効な REPOS モードを、シンクロナイズドアクション、および変数$AC_ REPOS_PATH_MODE を使用して読み取ることができます。

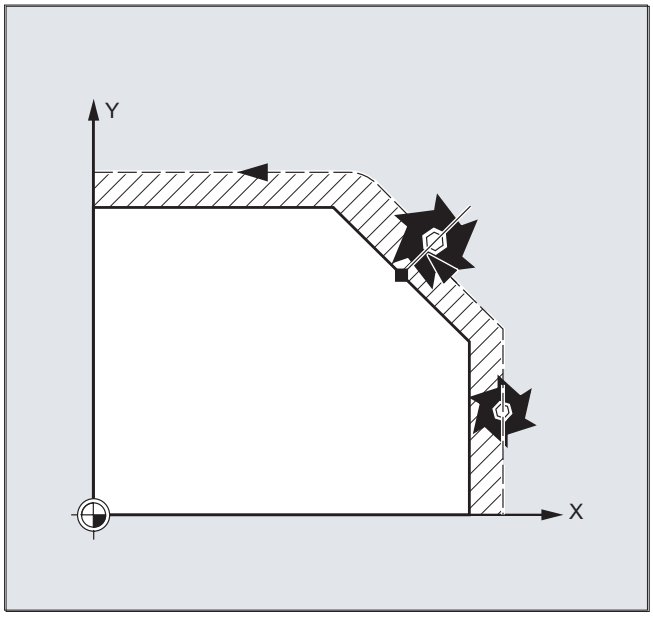
0　アプローチが定義されていません
1　RMBBL:始点へアプローチします
2　RMIBL:中断点へアプローチします
3　RMEBL:ブロックの終点へアプローチします
4　RMNBL:中断したブロックの次の軌跡点へアプローチします

新しい工具によるアプローチ

工具が破損したためにプログラムの実行を中止した場合は、以下が適用されます。

新しい D 番号をプログラム指令すると、再位置決め点では、加工プログラムは変更後の工具補正値を使用して続行します。

工具補正値を変更した場合は、中断点に再位置決めできない可能性があります。このような場合は、新しい輪郭上で、中断点に最も近い点(DISPR で変更された点の場合もあります)へアプローチします。



輪郭へのアプローチ

工具の輪郭上への再位置決め移動をプログラム指令できます。移動軸のアドレスに対してはゼロを入力してください。

REPOSA、REPOSQA、と REPOSHA の各命令は、すべての軸を自動的に再位置決めします。個々の軸の名称を指定する必要はありません。

REPOSL、REPOSQ、と REPOSH 命令をプログラム指令すると、自動的にすべてのジオメト軸が移動します。つまり、命令で軸を指定する必要はありません。他のすべての軸は、命令に指定してください。

REPOSH と REPOSQ の円弧移動には以下が適用されます。

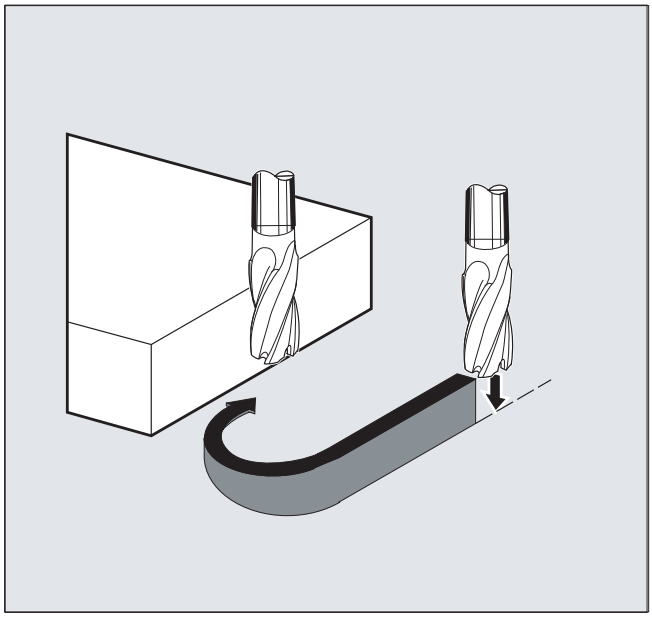
円弧は指定した作業平面 G17 ～ G19 で移動します。

アプローチブロックで 3 番目のジオメトリ軸(切り込み方向)を指定した場合は、切り込み方向のプログラム指令位置と工具位置が一致しない場合、らせんを描いて再位置決め点へアプローチします。

12.9 輪郭への再位置決め(REPOSA、REPOSL、REPOSQ、REPOSQA、REPOSH、REPOSHA、DISR、DISPR、RMIBL、RMBBL、RMEBL、RMNBL)

以下の場合には、直線アプローチ REPOSL に自動的に切り替わります。

- DISR に値を指定していない。
- 定義されたアプローチ方向がない(移動情報のないブロックのプログラム中断)。
- アプローチ方向が現在の作業平面に垂直。



12.10 モーションコントロールへの影響

12.10.1 加々速度補正の割合(JERKLIM)

NC 命令「JERKLIM」を使用すると、軌跡移動に対する軸の最大加々速度(マシンデータで設定)を、問題となるプログラム区間で減少または増加させることができます。

必要条件

加減速モード SOFT を有効にしてください。

有効性

機能は、次の場合に有効です。

- 運転モードが AUTOMATIC の場合。
- 軌跡軸のみの場合。

構文

```
JERKLIM[<軸>]=<値>
```

意味

JERKLIM:	加々速度補正命令	
<軸>:	加々速度制限値を適用する機械軸。	
<値>:	設定された軸移動の最大軸加々速度(MD32431 $MA_MAX_AX_JERK)を基準とした補正値の割合(%)です。	
	値の範囲:	1 ... 200
	値 100 は加々速度に影響しません。	

注記

パートプログラム終了時とチャネルリセット時の JERKLIM の動作は、マシンデータ MD32320 $MA_DYN_LIMIT_RESET_MASK のビット 0 で、次のように設定します。

- ビット 0 = 0:
  JERKLIM にプログラム指令した値は、チャネルリセット/M30 で 100%にリセットされます。
- ビット 0 = 1:
  JERKLIM にプログラム指令した値は、チャネルリセット/M30 後も保持されます。

例

プログラムコード	コメント
...	
N60 JERKLIM[X]=75	; X 方向への軸送りは、軸の許容加々速度の 75%以下で加速/減速します。
...	

12.10.2 速度補正の割合(VELOLIM)

軸の最大許容速度、またはマシンデータで設定される主軸の最大許容ギヤ選択依存速度は、パートプログラムまたはシンクロナイズドアクションで VELOLIM 命令を使用して下げることができます。

有効性

機能は、次の場合に有効です。

- 運転モードが AUTOMATIC の場合。
- 軌跡軸と位置決め軸の場合。
- 主軸運転/軸運転での主軸の場合。

構文

```
VELOLIM[<軸/主軸>]=<値>
```

意味

VELOLIM:	速度補正命令
<軸/主軸>:	速度または速度制限値の調整が必要な軸または主軸。 **主軸の VELOLIM** パートプログラムのプログラミング時にマシンデータ (MD30455 $MA_MISC_FUNCTION_MASK、ビット 6)を使用して、「VELOLIM」が、主軸か軸かに無関係に有効である(ビット 6 = 1)か、または各運転モードで個別にプログラム指令できる(ビット 6 = 0)かを設定できます。個別に有効にする場合は、プログラミング時に次のように識別子で選択します。 ● 主軸運転モードの場合は主軸識別子 S<n> ● 軸識別子、例:「C」、軸運転の場合
<値>:	補正値の割合(%) 補正値は、以下を基準にします。 ● 軸運転モードでの軸/主軸(MD30455 ビット 6 == 0): 設定した最大軸速度(MD32000 $MA_MAX_AX_VELO)。 ● 主軸運転または軸運転での主軸(MD30455 ビット 6 == 1): 動作中のギヤ選択の最大速度 (MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[<n>])
	値の範囲: 1 ... 100
	値 100 は速度には影響しません。

注記

パートプログラムの終了時とチャネルリセット時の動作

パートプログラム終了時とチャネルリセット時の「VELOLIM」の動作は、マシンデータで次のように設定できます。MD32320 $MA_DYN_LIMIT_RESET_MASK、ビット 0

主軸運転での有効な速度制限の検出

「VELOLIM」による速度制限(100 %未満)は、主軸運転で次のシステム変数を使用して検出できます。

- $AC_SMAXVELO (最大許容主軸速度)
- $AC_SMAXVELO_INFO (速度制限要因の識別子)

例

例 1: 速度制限、機械軸

プログラムコード	コメント
...	
N70 VELOLIM[X]=80	; X 方向への軸送りは、軸の許容速度の 80%以下で移動します。
...	

例 2: 速度制限、主軸

プログラムコード	コメント
N05 VELOLIM[S1]=90	; 主軸 1 の最大速度を、1000 rpm の 90%に制限します。
...	
N50 VELOLIM[C]=45	; 1000 rpm の 45%に速度を制限します。C は S1 の軸識別子です。
...	

主軸 1(AX5)のマシンデータ設定

- ギヤ選択の最高速度 1 = 1000 rpm:
  MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[1, AX5] = 1000
- 「VELOLIM」をプログラム指令すると、次の指定した識別子に関係なく、主軸と軸の運転に同時に機能します。
  MD30455 $MA_MISC_FUNCTION_MASK[AX5]、ビット 6 = 1

12.10.3 JERKLIM と VELOLIM のプログラム例

以下のプログラムは、加々速度制限と速度制限の割合の使用例を示します。

プログラムコード	コメント
N1000 G0 X0 Y0 F10000 SOFT G64	
N1100 G1 X20 RNDM=5 ACC[X]=20	
ACC[Y]=30	
N1200 G1 Y20 VELOLIM[X]=5	; X 方向への軸送りは、軸の許容速度の 5%以下でおこないます。
JERKLIM[Y]=200	; Y 方向への軸送りは、軸の許容加々速度の 200%まで加速/減速できます。
N1300 G1 X0 JERKLIM[X]=2	; X 方向への軸送りは、軸の許容加々速度の 2%以下で加速/減速します。
N1400 G1 Y0	
M30	

12.11 輪郭/旋回の許容範囲のプログラミング(CTOL、OTOL、ATOL)

パートプログラムのコンプレッサ機能、スムージング、および旋回のスムージングについて、マシンデータとセッティングデータを使用してパラメータ設定された加工許容範囲を調整するために、アドレス CTOL、OTOL、および ATOL を使用できます。

プログラムされた許容値は、再度プログラムされるか、負の値の割り当てにより削除されるまで有効です。さらにこの許容値は、プログラムの終了時またはリセット時に、削除されます。パラメータ設定された許容値は、削除後に再度有効になります。

構文

```
CTOL=<Value>
OTOL=<Value>
ATOL[<Axis>]=<Value>
```

意味

CTOL:	**輪郭許容範囲**をプログラムするアドレス			
	用途:	● すべてのコンプレッサ機能 ● G641 と G644 を除く、すべての丸み付けタイプ		
	先読み停止:	なし		
	効果:	モーダル		
	<Value>:	輪郭の許容範囲の値は長さで指定します。		
		タイプ:	REAL	
		単位:	inch/mm (現在の寸法設定で決まります)	
		値の範囲:	≥ 0:	許容誤差値
			< 0:	プログラムされた許容範囲値が削除されます ⇒ マシンデータまたはセッティングデータでパラメータ設定された許容範囲値が再度有効になります。

OTOL:	**旋回許容範囲**をプログラムするアドレス			
	用途:	• すべてのコンプレッサ機能 • ORISON 旋回スムージング • G641、G644、および OSD を除く、すべてのスムージングタイプ		
	先読み停止:	なし		
	効果:	モーダル		
	<Value>:	旋回の許容範囲は角度で指定します。		
		タイプ:	REAL	
		単位:	°	
		値の範囲:	≥ 0:	許容誤差値
			< 0:	プログラムされた許容範囲値が削除されます ⇒ マシンデータまたはセッティングデータでパラメータ設定された許容範囲値が再度有効になります。

ATOL:	**軸別の許容範囲**のプログラミングのアドレス			
	用途:	• すべてのコンプレッサ機能 • ORISON 旋回スムージング • G641、G644、および OSD を除く、すべてのスムージングタイプ		
	先読み停止:	なし		
	効果:	モーダル		
	<Axis>:	指令された許容範囲が適用されるチャネル軸の名前		
	<Value>:	軸の許容範囲の値は、軸タイプ(直線軸または回転軸)に応じて、長さまたは角度で指定します。		
		タイプ:	REAL	
		単位:	直線軸の場合:	inch/mm (現在の寸法設定で決まります)
			回転軸の場合:	°
		値の範囲:	≥ 0:	許容誤差値
			< 0:	プログラムされた許容範囲値が削除されます ⇒ マシンデータまたはセッティングデータでパラメータ設定された許容範囲値が再度有効になります。

注記

CTOL および OTOL でプログラム指令されたチャネル別許容範囲値が、ATOL プログラム指令された軸別許容範囲値より優先して適用されます。

注記

スケーリングフレーム

スケーリングフレームは、軸位置と同様にプログラム指令の許容範囲に影響します。つまり、相対的な許容範囲はそのまま同じです。

例

プログラムコード	コメント
COMPCAD G645 G1 F10000	; COMPCAD コンプレッサ機能を起動します。
X...Y...Z...	; ここで、マシンデータとセッティングデータが適用されます。
X...Y...Z...	

プログラムコード	コメント
X...Y...Z...	
CTOL=0.02	; これ以降、0.02mm の輪郭の許容範囲の適用が開始されます。
X...Y...Z...	
X...Y...Z...	
X...Y...Z...	
ASCALE X0.25 Y0.25 Z0.25	; これ以降、0.005mm の輪郭の許容範囲の適用が開始されます。
X...Y...Z...	
X...Y...Z...	
X...Y...Z...	
CTOL=-1	; これ以降、再度マシンデータとセッティングデータの適用が開始されます。
X...Y...Z...	
X...Y...Z...	
X...Y...Z...	

システム変数

先読み停止による読み取り

以下のシステム変数を使用して、現在有効な許容範囲を、パートプログラムとシンクロナイズドアクションで読み取ることができます。

- $AC_CTOL
  現在のメインランブロックが先読みされたときの有効なチャネル別輪郭許容範囲。
  有効な輪郭の許容範囲がない場合は、$AC_CTOL が、ジオメトリ軸の許容範囲の二乗の合計の平方根を返します。
- $AC_OTOL
  現在のメインランブロックが先読みされたときの有効なチャネル別旋回許容範囲。
  有効な向きの許容範囲がない場合は、$AC_OTOL が、動作中の方向座標変換で、旋回軸の許容範囲の二乗の合計の平方根を返します。それ以外の場合は、値「-1」を返します。
- $AA_ATOL[<軸>]
  現在のメインランブロックが先読みされたときの有効な軸別輪郭許容範囲。
  有効な輪郭許容範囲がない場合、$AA_ATOL[<ジオメトリ軸>]は、ジオメトリ軸の数の平方根で除算した輪郭許容範囲を返します。
  向きの許容範囲と方向座標変換が有効な場合、$AA_ATOL[<旋回軸>]は、旋回軸の数の平方根で除算した向きの許容範囲を返します。

注記

許容範囲値がプログラム指令されている場合は、個々の機能の許容範囲を区別できるほど、$A 変数が差別化されていません。

このような状況は、マシンデータとセッティングデータで、コンプレッサ機能、スムージング、および旋回スムージングに異なる許容範囲を設定した場合に発生する可能性があります。システム変数は、現在有効な機能で発生する最大の値を返します。たとえば、その向きの許容範囲 0.1°のコンプレッサ機能、および 1°の ORISON 旋回スムージングが有効な場合は、$AC_OTOL 変数は値「1」を返します。向きのスムージングが無効な場合は、$AC_OTOL が値「0.1」を返します。

先読み停止のない読み取り

以下のシステム変数を使用して、現在有効な許容範囲を、パートプログラムで読み取ることができます。

- $P_CTOL
  現在有効なチャネル別の輪郭許容範囲。
- $P_OTOL
  現在有効なチャネル別の向きの許容範囲。
- $PA_ATOL
  現在有効な軸別の輪郭許容範囲。

必要条件

CTOL、OTOL、および ATOL でプログラムされた許容範囲は、それらの許容範囲に間接的に依存している機能にも影響します。

- 指令値演算でのコードエラーの制限
- 自由曲面モードの基本機能

以下のスムージング機能は、CTOL、OTOL、および ATOL のプログラミングには影響**されません**。

- OSD による旋回のスムージング
  OSD は、許容範囲を使用せず、ブロック遷移部からの距離を使用します。
- G644 によるスムージング
  G644 は、スムージングには使用されず、工具交換の最適化、および加工を伴わないその他の動作に使用されます。
- G645 によるスムージング
  G645 は、実質的に常に G642 と同様に動作し、プログラム指令された許容範囲を使用します。マシンデータ MD33120 $MA_PATH_TRANS_POS_TOL の許容範囲の値は、曲率の不連続変化を含む、接線変化が均一なブロック遷移部(接線の変化が連続的な円弧/直線遷移部など)でのみ使用します。これらの点での丸み付け軌跡は、プログラム指令輪郭の外にある場合もあり、この場合は、多くの用途で許容範囲が小さくなります。さらに、通常は、NC プログラマには無関係な曲率の変化の種類を補正するために、小さい固定許容範囲が必要となります。

12.12 有効な連結によるブロック変更動作(CPBC)

CPBC 命令は、パートプログラムで有効な連結によってブロック変更を実行するために満たさなければならないブロック変更条件を指定します。

構文

```
CPBC[<従動軸>] = <条件>
```

意味

CPBC:	有効な連結によるブロック変更条件	
<従動軸>:	従動軸の軸識別子	
<条件>:	ブロック変更条件	
	タイプ:	STRING
	規格値	**意味:**ブロック変更が実行されます
	"NOC"	連結状態に関わりなく
	"IPOSTOP"	指令同期制御用
	"COARSE"	現在値同期制御「汎用」用
	"FINE"	現在値同期制御「精密」用

例

プログラムコード

```
; 以下の場合、ブロック変更が行われます:
;  - 従動軸 X2 への連結 == 有効
;  - 指令同期制御 == 有効
CPBC[X2]="IPOSTOP"
```

軸連結 13

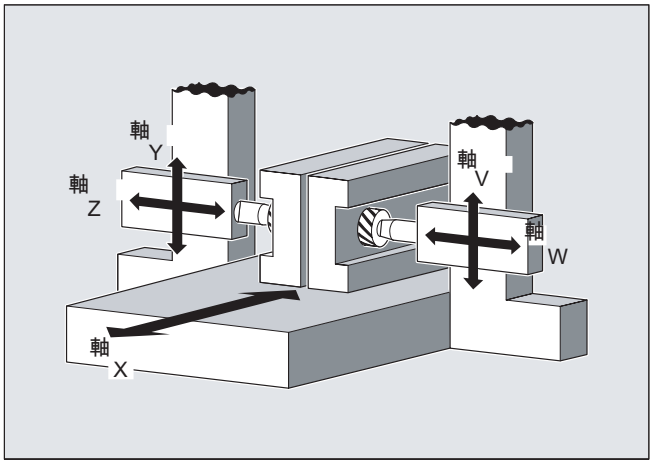
13.1 連結移動(TRAILON、TRAILOF)

定義したマスタ軸が移動すると、それに割り当てた連結移動軸(=スレーブ軸)は、連結係数を考慮に入れて、マスタ軸が指令する距離だけ移動します。

このようなマスタ軸とスレーブ軸を合わせて、連結軸と呼びます。

用途

- シミュレーション軸を使用した軸の移動。マスタ軸はシミュレーション軸で、連結軸は実軸です。このようにして、連結係数に応じて、実軸を移動できます。
- 2 つの連結移動グループによる両面加工:
  1 番目のマスタ軸 Y、連結移動軸 V
  2 番目のマスタ軸 Z、連結移動軸 W



構文

```
TRAILON(<スレーブ軸>,<マスタ軸>,<連結係数>)
TRAILOF(<スレーブ軸>,<マスタ軸>,<マスタ軸 2>)
TRAILOF(<スレーブ軸>)
```

意味

TRAILON:	連結軸グループを起動、および定義する命令です。	
	効果:	モーダル

<スレーブ軸>:	パラメータ 1:スレーブ軸の軸名称 **注:** 連結移動軸は、他の連結移動軸のマスタ軸としても機能できます。このようには、各種さまざまな連結軸グループを作成できます。
<マスタ軸>:	パラメータ 2:スレーブ軸の軸名称
<連結係数>:	パラメータ 3:連結係数 連結係数は、連結移動軸とマスタ軸の軌跡間に必要な関係を、次のように指定します。 **<**連結係数**> =**連結移動軸の移動量/マスタ軸の移動量
	タイプ: REAL
	初期設定: 1
	負の値を入力すると、マスタ軸と連結軸が逆方向に移動します。 連結係数をプログラム指令しない場合は、連結係数 1 が自動的に適用されます。
TRAILOF:	連結軸グループを解除する命令です。
	効果: モーダル
	2 個のパラメータを持つ TRAILOF は、以下のように指定したマスタ軸との連結のみを解除します。 TRAILOF(<スレーブ軸>,<マスタ軸>) 連結移動軸にマスタ軸が 2 つ含まれる場合は、3 個のパラメータで TRAILOF を呼び出して、両方の連結を解除できます。 TRAILOF(<スレーブ軸>,<マスタ軸>,<マスタ軸 2>) 次のように、マスタ軸を指定せずに TRAILOF をプログラミングしても同じ結果となります。 TRAILOF(<スレーブ軸>)

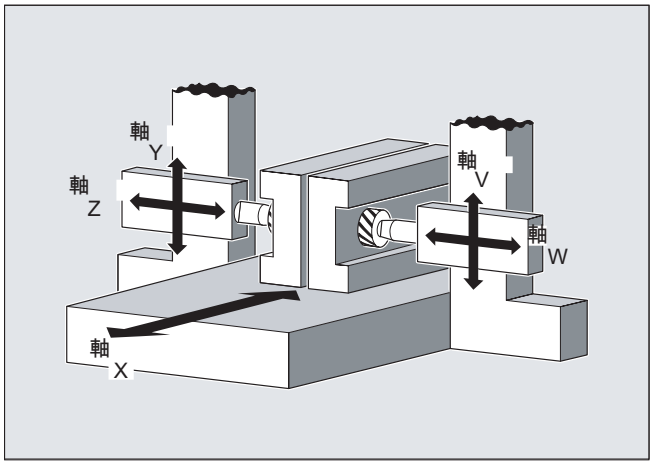
注記

連結軸の移動は常に、基本座標系(BCS)で実行されます。

同時に起動できる連結軸グループの数は、機械上の軸の組み合わせ数の最大値によってのみ制限されます。

例

ワークは、図に示す軸構成により両面加工されます。これをおこなうには、連結軸の組み合わせを 2 つ作成してください。



プログラムコード	コメント
…	
N100 TRAILON(V,Y)	; 1 番目の連結軸グループを適用します。
N110 TRAILON(W,Z,-1)	; 2 番目の連結軸グループを適用します。負の連結係数です。連結移動軸は、マスタ軸から反対方向に移動します。
N120 G0 Z10	; それぞれ反対の軸方向へ Z 軸と W 軸が切り込みます。
N130 G0 Y20	; 同じ軸方向へ Y 軸と V 軸が切り込みます。
…	
N200 G1 Y22 V25 F200	; 連結移動軸 V の連結移動と独立移動を重畳します。
…	
TRAILOF(V,Y)	; 1 番目の連結軸グループを解除します。
TRAILOF(W,Z)	; 2 番目の連結軸グループを解除します。

詳細情報

軸タイプ

連結軸グループは、任意の直線軸と回転軸を組み合わせて作成できます。 シミュレーション軸は、マスタ軸としても定義できます。

連結移動軸

2 つまでのマスタ軸を同時に、1 つのスレーブ軸に割り当てることができます。 連結軸の組合わせが異なるように割り当てます。

連結移動軸は、使用可能なすべての範囲の動作命令(G0、G1、G2、G3 など)でプログラム指令できます。 連結軸は、独立して定義された軌跡だけでなく、連結係数に基づく連結軸のマスタ軸から計算された軌跡も移動します。

応答性制限

応答性制限は、連結軸グループの適用タイプによって、次のように異なります。

- パートプログラムでの起動
  起動がパートプログラムで実行され、起動されたチャネルですべてのマスタ軸がプログラム軸として有効である場合は、マスタ軸の移動中に、すべての連結移動軸のダイナミック応答が考慮され、連結移動軸の過負荷が防止されます。
  起動中のチャネルで、プログラム軸として有効でないマスタ軸によりパートプログラムで起動が実行された場合($AA_TYP ≠ 1)は、マスタ軸の移動中に、連結移動軸のダイナミック応答は考慮されません。 これにより、連結移動軸が過負荷になり、連結に必要なダイナミック応答が不足する場合があります。
- シンクロナイズドアクションでの起動
  起動がシンクロナイズドアクションで実行された場合は、マスタ軸の移動中に、連結移動軸のダイナミック応答は考慮されません。 これにより、連結移動軸が過負荷になり、連結に必要なダイナミック応答が不足する場合があります。

⚠ 注意
軸の過負荷 連結軸グループが有効な場合は、以下のようになります。 ● このため、シンクロナイズドアクションの場合 ● パートプログラムが、連結移動軸のチャネルにプログラム軸ではないマスタ軸を含む場合は、 マスタ軸上の移動により連結移動軸の過負荷が発生しないよう、ユーザー/工作機械メーカが適切な処置をおこなってください。

連結状態

軸の連結状態は、パートプログラムで次のシステム変数を使用して確認できます。

$AA_COUP_ACT[<軸>]

規格値	意味
0	連結は無効
8	連結移動が有効

モジュロ回転軸用の連結移動軸の残移動距離の表示

マスタ軸と連結移動軸がモジュロ回転軸の場合、マスタ軸での n * 360°(n = 1、2、3 ...) からの移動動作は、連結をオフにするまで連結移動軸の残移動距離表示で加算されます。

例: TRAILON でマスタ軸 B とスレーブ軸 C を定義するプログラム区間

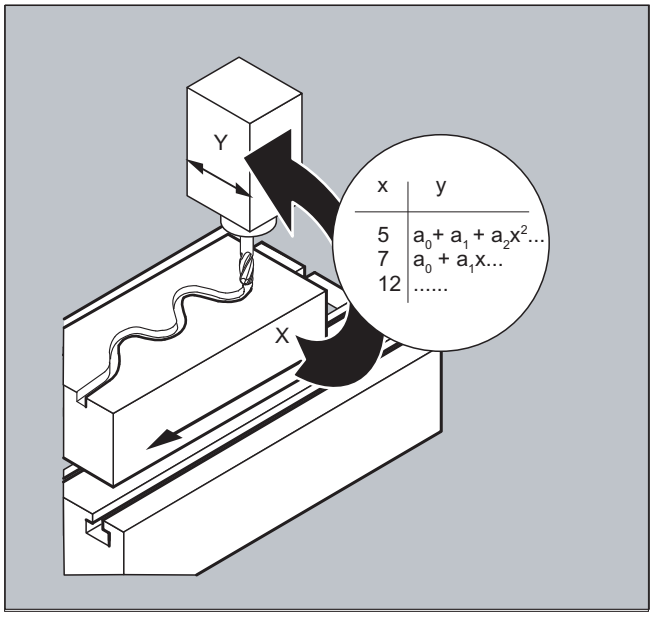
プログラムコード	コメント
TRAILON(C,B,1)	; 連結を起動します。
G0 B0	; 開始位置
	; ブロック始点の残移動距離表示:
G91 B360	; B=360, C=360
G91 B720	; B=720, C=1080
G91 B360	; B=360, C=1440

13.2 カーブテーブル(CTAB)

カーブテーブルを使用すると、2 つの軸(マスタ軸とスレーブ軸)の間の位置関係と速度関係をプログラムできます。カーブテーブルはパートプログラムで定義します。

用途

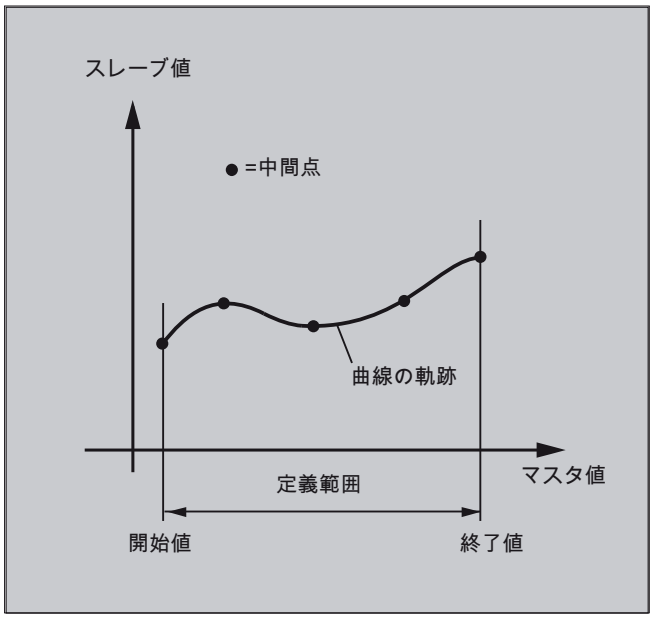
カーブテーブルは、メカニカルカムの代わりに使用できます。カーブテーブルは、マスタ値とスレーブ値の間の関数関係を作成して、軸間連動機能の基礎を形成します。適切にプログラム指令すると、制御装置は、マスタ軸とスレーブ軸の相対位置から、カムに対応する多項式を計算します。



13.2.1 カーブテーブルの定義(CTABDEF、CATBEND)

カーブテーブルとは、パートプログラムまたは、先頭の CTABDEF と末尾の CTABEND で囲まれた.パートプログラム区間です。

このパートプログラム区間内で、移動命令で固有のスレーブ軸位置を、マスタ軸の個々の位置に割り当てます。これらのスレーブ軸位置は、5 次までの多項式で曲線定義を計算するときの中間点として使用します。



必要条件

マシンデータは、カーブテーブルの定義に十分なメモリ空間が確保されるよう設定してください(→工作機械メーカ)。

構文

```
CTABDEF(<スレーブ軸>,<マスタ軸>,<n>,<周期性>[,<メモリロケーション>])
...
CTABEND
```

意味

CTABDEF( ):	カーブテーブル定義の開始
CTABEND:	カーブテーブル定義の終了
<スレーブ軸>:	カーブテーブルを使用して移動が計算される軸
<マスタ軸>:	スレーブ軸移動の計算用のマスタ値を提供する軸
<n>:	カーブテーブルの番号(ID) カーブテーブルの番号は一義的であり、メモリロケーションには影響されません。スタティックとダイナミック NC メモリの両方に、同じ番号のカーブテーブルは作成できません。

<周期性>:	カーブテーブルの周期性	
	0	カーブテーブルは周期的ではありません(カーブテーブルが処理されるのは、回転軸の場合でも 1 回のみです)
	1	カーブテーブルはマスタ軸に関して周期的です
	2	カーブテーブルはマスタ軸とスレーブ軸に関して周期的です
<メモリロケーション>:	メモリロケーションの指定(任意選択)	
	"SRAM"	カーブテーブルを**スタティック** NC メモリに作成します。
	"DRAM"	カーブテーブルを**ダイナミック** NC メモリに作成します。
	注: このパラメータに値をプログラム指令しない場合は、MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE で設定した既定のメモリロケーションを使用します。	

注記

上書き

カーブテーブルは、その番号(<n>)が別のカーブテーブル定義で使用されると、直ちに上書きされます(例外:カーブテーブルが軸連結で有効な場合、または CTABLOCK でロックされている場合)。**カーブテーブルが上書きされるときに、警告は出力されません。**

例

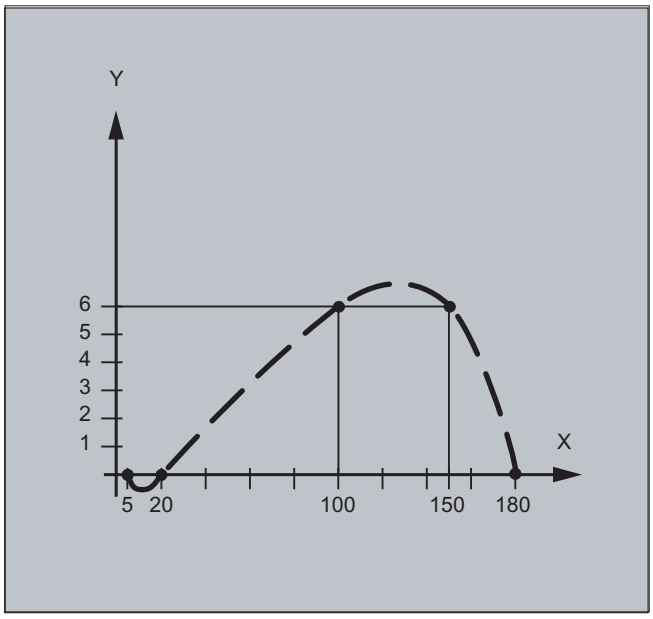
例 1:カーブテーブル定義のプログラム区間

プログラム区間をそのまま使用して、カーブテーブルを定義します。先読み停止の STOPRE 命令はそのまま変更されずに残すことができます。その後、プログラム区間がカーブテーブル定義に使用されなくなり、CTABDEF と CTABEND が削除されると、再びすぐに有効になります。

プログラムコード	コメント
…	
CTABDEF(Y,X,1,1)	; カーブテーブルの定義です。
…	
IF NOT ($P_CTABDEF)	
Stopre	
ENDIF	

プログラムコード	コメント
…	
CTABEND	

例 2 :周期的でないカーブテーブルの定義



プログラムコード	コメント
N100 CTABDEF(Y,X,3,0)	; 周期的でないカーブテーブル (番号 3) の定義の開始。
N110 X0 Y0	; 1 番目の移動命令。開始値と 1 番目の中間点を定義します。 マスタ値:0、スレーブ値:0
N120 X20 Y0	; 2 番目の補間点 マスタ値:0…20、スレーブ値:開始値…0
N130 X100 Y6	; 3 番目の補間点 マスタ値:20…100、スレーブ値:0…6
N140 X150 Y6	; 4 番目の補間点 マスタ値:100…150、スレーブ値:6…6
N150 X180 Y0	; 5 番目の補間点 マスタ値:150…180、スレーブ値:6…0
N200 CTABEND	; 定義の終了カーブテーブルは、5 次以下の多項式としての内部形式で生成されます。指定した中間点による曲線定義の計算は、モーダルで選択された補間タイプ (円弧補間、直線補間、スプライン補間) によって異なります。定義を開始する前のパートプログラムの状態に復帰します。

例 3 :周期的なカーブテーブルの定義

番号 2 の周期的なカーブテーブル、マスタ値の範囲 0 ～ 360、0 から 45 に移動して 0 に戻るスレーブ軸で定義します。

プログラムコード	コメント
N10 DEF REAL DEPPOS	
N20 DEF REAL GRADIENT	
N30 CTABDEF(Y,X,2,1)	; 定義の開始
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	; 定義の終了
;Y と X の連結による曲線のテスト:	
N120 G1 F1000 X0	
N130 LEADON(Y,X,2)	
N140 X360	
N150 X0	
N160 LEADOF(Y,X)	
N170 DEPPOS=CTAB(75.0,2,GRADIENT)	; カーブテーブル機能でマスタ値 75.0 を読み取ります。
N180 G0 X75 Y=DEPPOS	; マスタ軸とスレーブ軸を位置決めします。
;連結の起動後は、スレーブ軸の同期制御は不要です。	
N190 LEADON(Y,X,2)	
N200 G1 X110 F1000	
N210 LEADOF(Y,X)	
N220 M30	

詳細情報

カーブテーブルの開始値と終了値

カーブテーブルの定義範囲の始まりを示す開始値は、カーブテーブル定義内で指定された、最初の当該の軸位置(最初の移動命令)です。カーブテーブルの定義範囲の終了値は、最後の移動指令に特定されます。

使用言語

カーブテーブル定義内では、すべての NC 言語を使用できます。

注記

以下の入力は、カーブテーブル定義で使用できません。

- 先読み停止
- マスタ軸移動の解除(座標変換を変更した場合など)
- スレーブ軸のみの移動命令
- マスタ軸の反転(つまり、マスタ軸の位置は常に一定であることが必要です)
- さまざまなプログラムレベルでの CTABDEF 命令と CTABEND 命令

モーダル命令の有効性

カーブテーブル定義内で作成したモーダル命令はすべて、カーブテーブル定義が完了すると、無効になります。したがって、カーブテーブル定義されたパートプログラムの前後では同じ状態になります。

R パラメータへの割り当て

カーブテーブル定義での R パラメータへの割り当ては、CTABEND 後にリセットされます。

例:

プログラムコード	コメント
...	
R10=5 R11=20	;R10=5
...	
CTABDEF	
G1 X=10 Y=20 F1000	
R10=R11+5	;R10=25
X=R10	
CTABEND	
...	;R10=5

ASPLINE、BSPLINE、CSPLINE の起動

ASPLINE、BSPLINE、または CSPLINE をカーブテーブル定義 CTABDEF ～ CTABEND 内で有効にした場合は、このスプラインの適用前に 1 個以上の始点をプログラム指令してください。CTABDEF の直後には適用しないでください。適用した場合は、カーブテーブル定義の前の現在の軸位置によって、スプラインが異なります。

例:

プログラムコード

```
...
CTABDEF(Y,X,1,0)
X0 Y0
ASPLINE
X=5 Y=10
X10 Y40
...
CTABEND
```

カーブテーブルの反復使用

カーブテーブルをスタティック NC メモリ(SRAM)に保存した場合は、カーブテーブルを使用して計算したマスタ軸とスレーブ軸の関数関係が、パートプログラムの終了、および POWER OFF の後も、選択したテーブル番号で保持されます。

ダイナミックメモリ(DRAM)で作成されたカーブテーブルは POWER ON 時に削除されるため、再作成が必要となる場合があります。

一度作成したカーブテーブルは、マスタ軸とスレーブ軸のあらゆる組み合わせに適用でき、作成に使用した軸には影響されません。

カーブテーブルの上書き

カーブテーブルは、その番号が別のカーブテーブル定義で使用されると、直ちに上書きされます。

例外:カーブテーブルが軸連結で有効な場合、または CTABLOCK でロックされている場合は例外です。

注記

カーブテーブルが上書きされるときに、警告は出力されません。

カーブテーブル定義の有効性

$P_CTABDEF システム変数は、いつでもパートプログラムで使用して、カーブテーブル定義が有効かどうかをチェックできます。

カーブテーブル定義の無効化

カーブテーブル定義に関連する命令を解除した後は、そのパートプログラム区間を、実際のパートプログラムとして再度使用できます。

「外部メモリからの実行」によるカーブテーブルのロード

カーブテーブルを外部メモリから実行する場合は、
MD18360 $MN_MM_EXT_PROG_BUFFER_SIZE での再ロードバッファ(DRAM)サイズを選択して、カーブテーブル定義全体を再ロードバッファに同時に格納できるようにしてください。これをおこなわない場合は、アラームが発生して、パートプログラムの処理がキャンセルされます。

スレーブ軸のスキップ

マシンデータ
MD20900 $MC_CTAB_ENABLE_NO_LEADMOTION
の設定に応じて、マスタ軸で移動が生じない場合のスレーブ軸のスキップが可能となります。

13.2.2 カーブテーブルの有無の確認(CTABEXISTS)

CTABEXISTS 命令を使用すると、特定の番号のカーブテーブルが NC メモリにあるかどうかを確認できます。

構文

```
CTABEXISTS(<n>)
```

意味

CTABEXISTS:	スタティックまたはダイナミック NC メモリのカーブテーブル番号<n>の有無を確認します。	
	0	カーブテーブルがありません
	1	カーブテーブルがあります
<n>:	カーブテーブルの番号(ID)	

13.2.3 カーブテーブルの削除(CTABDEL)

CTABDEL を使用すると、カーブテーブルを削除できます。

注記

軸連結で有効なカーブテーブルは削除できません。

構文

```
CTABDEL(<n>)
CTABDEL(<n>,<m>)
CTABDEL(<n>,<m>,<メモリロケーション>)
CTABDEL ()
CTABDEL(,,<メモリロケーション>)
```

意味

CTABDEL:	カーブテーブルを削除する命令	
<n>:	削除するカーブテーブルの番号(**ID**) カーブテーブルの範囲 CTABDEL(<n>,<m>) を削除する場合、<n> は、その範囲内の最初のカーブテーブルの番号を指定します。	
<m>:	カーブテーブルの範囲 CTABDEL(<n>,<m>) を削除する場合、<m> は、その範囲内の最後のカーブテーブルの番号を指定します。 <m>には、<n>より大きい値を指定してください。	
<メモリロケーション>:	メモリロケーションの指定(任意選択) メモリロケーションを**指定せずに**削除すると、指定したカーブテーブルをスタティックとダイナミック NC メモリで削除します。 メモリロケーションを**指定して**削除すると、指定したカーブテーブルのうち、指定したメモリロケーションのカーブテーブルのみを削除します。残りのカーブテーブルは保持されます。	
	"SRAM"	**スタティック** NC メモリの削除
	"DRAM"	**ダイナミック** NC メモリの削除

削除するカーブテーブルをプログラム指令せずに CTABDEL をプログラム指令した場合は、**すべての**カーブテーブル、または指定したメモリのすべてのカーブテーブルを、次のように削除します。

CTABDEL():	スタティックとダイナミック NC メモリのすべてのカーブテーブルを削除します
CTABDEL(,,"SRAM"):	スタティック NC メモリのすべてのカーブテーブルを削除します
CTABDEL(,,"DRAM"):	ダイナミック NC メモリのすべてのカーブテーブルを削除します

注記

CTABDEL(<n>,<m>)または CTABDEL()で複数のカーブテーブルを削除するときに、それらのカーブテーブルの 1 つ以上が連結で有効な場合は、削除命令が実行されません。つまり、指定したカーブテーブルは、**すべて**削除されません。

13.2.4 カーブテーブルの削除、および上書きを防止するロック(CTABLOCK、CTABUNLOCK)

ロックを設定すると、不用意な削除と上書きからカーブテーブルを保護できます。ロックは、設定後はいつでも解除できます。

構文

ロック:
```
CTABLOCK(<n>)
CTABLOCK(<n>,<m>)
CTABLOCK(<n>,<m>,<メモリロケーション>)
CTABLOCK()
CTABLOCK(,,<メモリロケーション>)
```

ロック解除:
```
CTABUNLOCK(<n>)
CTABUNLOCK(<n>,<m>)
CTABUNLOCK(<n>,<m>,<メモリロケーション>)
CTABUNLOCK()
CTABUNLOCK(,,<メモリロケーション>)
```

意味

CTABLOCK:	削除/上書きを防止するロックを**設定**する命令
CTABUNLOCK:	削除/上書きを防止するロックを**解除**する命令 CTABUNLOCK は、CTABLOCK でロックしたカーブテーブルのロックを解除します。動作中の連結で使用されるカーブテーブルのロックは解除されず、削除できません。CTABLOCK によるロックは、連結が解除されたタイミングで動作中の連結に適用されていたロックが解除されると同時に、解除されます。これによって、このカーブテーブルは削除できます。CTABUNLOCK を再度呼び出す必要はありません。

<n>:	ロック/ロック解除するカーブテーブルの番号(ID) カーブテーブルの範囲 CTABLOCK(<n>,<m>)/ CTABUNLOCK(<n>,<m>)をロック/ロック解除する場合、<n>を使用してその範囲内の最初のカーブテーブルの番号を指定します。
<m>:	カーブテーブルの範囲 CTABLOCK(<n>,<m>)/ CTABUNLOCK(<n>,<m>)をロック/ロック解除する場合、<m>を使用してその範囲内の最後のカーブテーブルの番号を指定します。 <m>には、<n>より大きい値を指定してください。
<メモリロケーション>:	メモリロケーションの指定(任意選択) メモリロケーションを**指定せずに**ロック/ロック解除すると、指定したカーブテーブルをスタティックとダイナミック NC メモリでロック/ロック解除します。 メモリロケーションを**指定して**ロック/ロック解除すると、指定したカーブテーブルのうち、指定したメモリロケーションのカーブテーブルのみをロック/ロック解除します。残りのカーブテーブルはロック/ロック解除されません。
	"SRAM" **スタティック** NC メモリでのロック/ロック解除
	"DRAM" **ダイナミック** NC メモリでのロック/ロック解除

ロック/ロック解除するカーブテーブルをプログラム指令せずに CTABLOCK/ CTABUNLOCK をプログラム指令した場合は、**すべての**カーブテーブル、または指定したメモリのすべてのカーブテーブルを、次のようにロック/ロック解除します。

CTABLOCK():	スタティックとダイナミック NC メモリのすべてのカーブテーブルをロックします
CTABLOCK(,,"SRAM"):	スタティック NC メモリのすべてのカーブテーブルをロックします
CTABLOCK(,,"DRAM"):	ダイナミック NC メモリのすべてのカーブテーブルをロックします
CTABUNLOCK():	スタティックとダイナミック NC メモリのすべてのカーブテーブルをロック解除します

CTABUNLOCK(,,"SRAM"):	スタティック NC メモリのすべてのカーブテーブルをロック解除します
CTABUNLOCK(,,"DRAM"):	ダイナミック NC メモリのすべてのカーブテーブルをロック解除します

13.2.5 カーブテーブル: カーブテーブル機能の特定(CTABID、CTABISLOCK、CTABMEMTYP、CTABPERIOD)

これらの命令を使用すると、カーブテーブルの重要な機能(テーブル番号、ロック状態、メモリロケーション、周期性)を調べることができます。

構文

```
CTABID(<p>)
CTABID(<p>,<メモリロケーション>)
CTABISLOCK(<n>)
CTABMEMTYP(<n>)
TABPERIOD(<n>)
```

意味

CTABID:	指定したメモリの<p>番目のカーブテーブルとして入力した**テーブル番号**を返します。 例: CTABID(1,"SRAM")は、スタティック NC メモリの 1 番目のカーブテーブルの番号を返します。この場合、1 番目のカーブテーブルとは、最も大きいテーブル番号のカーブテーブルです。 **注:** 連続する CTABID の呼び出しの間に、CTABDEL によるカーブテーブルの削除などの理由で、メモリのカーブテーブルの順序が変わった場合は、CTABID(<p>,...)が、同じ<p>番目の別のカーブテーブルを返す場合があります。	
CTABISLOCK:	番号<n>のカーブテーブルの**ロック状態**を、次のように返します。	
	0	カーブテーブルはロックされていません
	1	CTABLOCK によりカーブテーブルがロックされています
	2	動作中の連結によりカーブテーブルがロックされています
	3	CTABLOCK、および動作中の連結によりカーブテーブルがロックされています
	-1	カーブテーブルがありません
CTABMEMTYP:	番号<n>のカーブテーブルの**メモリロケーション**を、次のように返します。	
	0	スタティック NC メモリのカーブテーブル
	1	ダイナミック NC メモリのカーブテーブル
	-1	カーブテーブルがありません
CTABPERIOD:	番号<n>のカーブテーブルの**周期性**を、次のように返します。	
	0	カーブテーブルは周期的ではありません
	1	カーブテーブルはマスタ軸で周期的です
	2	カーブテーブルはマスタ軸とスレーブ軸で周期的です
	-1	カーブテーブルがありません
<p>:	メモリの登録番号	
<n>:	カーブテーブルの番号(**ID**)	

<メモリロケーション>:	メモリロケーションの指定(任意選択)	
	"SRAM"	スタティック NC メモリ
	"DRAM"	ダイナミック NC メモリ
	注: このパラメータに値をプログラム指令しない場合は、MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE で設定した既定のメモリロケーションを使用します。	

13.2.6 カーブテーブル値の読み取り(CTABTSV、CTABTEV、CTABTSP、CTABTEP、CTABSSV、CTABSEV、CTAB、CTABINV、CTABTMIN、CTABTMAX)

パートプログラムで、次のカーブテーブル値を読み取ることができます。

- カーブテーブルの開始と終了のスレーブ軸値とマスタ軸値
- カーブセグメントの開始と終了のスレーブ軸値
- マスタ軸値に対するスレーブ軸値
- スレーブ軸値に対するマスタ軸値
- 以下のスレーブ軸の最小値と最大値
  - カーブテーブルの定義範囲全体
    または
  - 定義したカーブテーブル区間

構文

```
CTABTSV(<n>,<傾き>[,<スレーブ軸>])
CTABTEV(<n>,<傾き>[,<スレーブ軸>])
CTABTSP(<n>,<傾き>[,<マスタ軸>])
CTABTEP(<n>,<傾き>[,<マスタ軸>])
CTABSSV(<マスタ値>,<n>,<傾き>[,<スレーブ軸>])
CTABSEV(<マスタ値>,<n>,<傾き>[,<スレーブ軸>])
CTABSEV(<マスタ値>,<n>,<傾き>[,<スレーブ軸>])
CTABINV(<スレーブ値>,<近似値>,<n>,<傾き>[,<スレーブ軸>,<マスタ軸>]
CTABTMIN(<n>[,<スレーブ軸>])
CTABTMAX(<n>[,<スレーブ軸>])
CTABTMIN(<n>,<a>,<b>[,<スレーブ軸>,<マスタ軸>])
CTABTMAX(<n>,<a>,<b>[,<スレーブ軸>,<マスタ軸>])
```

意味

CTABTSV:	番号<n>のカーブテーブルの**開始位置**でスレーブ軸値を読み取ります
CTABTEV:	番号<n>のカーブテーブルの**終了位置**でスレーブ軸値を読み取ります
CTABTSP:	番号<n>のカーブテーブルの**開始位置**でマスタ軸値を読み取ります
CTABTEP:	番号<n>のカーブテーブルの**終了位置**でマスタ軸値を読み取ります
CTABSSV:	指定したマスタ軸値(マスタ値>)のカーブセグメントの**開始位置**で、スレーブ軸値を読み取ります
CTABSEV:	指定したマスタ軸値(マスタ値>)のカーブセグメントの**終了位置**で、スレーブ軸値を読み取ります
CTAB:	指定したマスタ軸値(マスタ値>)に対するスレーブ軸値を読み取ります
CTABINV:	指定したスレーブ軸値(スレーブ値>)に対するマスタ軸値を読み取ります
CTABTMIN:	以下のスレーブ軸の**最小値**を定義します ● カーブテーブルの定義範囲全体 または ● <a> ... <b>までの、定義した区間
CTABTMAX:	以下のスレーブ軸の**最大値**を定義します ● カーブテーブルの定義範囲全体 または ● <a> ... <b>までの、定義した区間
<n>:	カーブテーブルの番号(**ID**)
<傾き>:	傾き>パラメータは、計算した位置でのカーブテーブル機能の**傾斜**を返します。
<スレーブ軸>:	移動がカーブテーブルを使用して計算される軸(任意選択)
<マスタ軸>:	スレーブ軸移動の計算のマスタ値を提供する軸(任意選択)
<スレーブ値>:	CTABINV の関連するマスタ軸値を読み取るためのスレーブ軸値
<マスタ値>:	次の用途のマスタ軸値です。 ● CTAB による、関連するスレーブ軸値の読み取り または ● CTABSSV/CTABSEV によるカーブセグメントの選択

<近似値>:	CTABINV によるマスタ軸値のスレーブ軸値への割り当ては、常に一義的である必要はありません。したがって、CTABINV にはパラメータとして、予想されるマスタ軸値の近似値が必要となります。
<a>:	CTABTMIN/CTABTMAX によるマスタ値区間の**下限**
<b>:	CTABTMIN/CTABTMAX によるマスタ値区間の**上限**
	注: a>から b>までのマスタ値区間は常に、カーブテーブルの定義範囲内にしてください。

例

例 1:

カーブテーブルの定義範囲全体で、スレーブ軸の最小値と最大値により、カーブテーブルの開始位置と終了位置のスレーブ軸値とマスタ軸値を、定義します。

```
プログラムコード                          コメント
N10 DEF REAL STARTPOS
N20 DEF REAL ENDPOS
N30 DEF REAL STARTPARA
N40 DEF REAL ENDPARA
N50 DEF REAL MINVAL
N60 DEF REAL MAXVAL
N70 DEF REAL GRADIENT
...
N100 CTABDEF(Y,X,1,0)                     ; カーブテーブル定義の開始。
N110 X0 Y10                               ; 1 番目のカーブテーブルセグメントの開始位置。
N120 X30 Y40                              ; 1 番目のカーブテーブルセグメントの終了位置=
                                          2 番目のカーブテーブルセグメントの開始位置
N130 X60 Y5                               ; 2 番目のカーブテーブルセグメントの終了位置= ...
N140 X70 Y30
N150 X80 Y20
N160 CTABEND                              ; カーブテーブル定義の終了。
...
N200 STARTPOS=CTABTSV(1,GRADIENT)         ; カーブテーブル開始位置のスレーブ軸値= 10
N210 ENDPOS=CTABTEV(1,GRADIENT)           ; カーブテーブル終了位置のスレーブ軸値= 20
N220 STARTPARA=CTABTSP(1,GRADIENT)        ; カーブテーブル開始位置のマスタ軸値= 0
N230 ENDPARA=CTABTEP(1,GRADIENT)          ; カーブテーブル終了位置のマスタ軸値= 80
N240 MINVAL=CTABTMIN(1)                   ; Y=5 のスレーブ軸の最小値
N250 MAXVAL=CTABTMAX(1)                   ; Y=40 のスレーブ軸の最大値
```

例 2 :

マスタ軸値 X=30 に関連するカーブセグメントの開始位置と終了位置のスレーブ軸値を特定します。

プログラムコード	コメント
N10 DEF REAL STARTPOS	
N20 DEF REAL ENDPOS	
N30 DEF REAL GRADIENT	
...	
N100 CTABDEF(Y,X,1,0)	; カーブテーブル定義の開始。
N110 X0 Y0	; 1 番目のカーブテーブルセグメントの開始位置。
N120 X20 Y10	; 1 番目のカーブテーブルセグメントの終了位置= 2 番目のカーブテーブルセグメントの開始位置
N130 X40 Y40	; 2 番目のカーブテーブルセグメントの終了位置= ...
N140 X60 Y10	
N150 X80 Y0	
N160 CTABEND	; カーブテーブル定義の終了。
...	
N200 STARTPOS=CTABSSV(30.0,1,GRADIENT)	; 2 番目のセグメントの開始位置 Y = 10
N210 ENDPOS=CTABSEV(30.0,1,GRADIENT)	; 2 番目のセグメントの終了位置 Y = 40

詳細情報

シンクロナイズドアクションでの使用

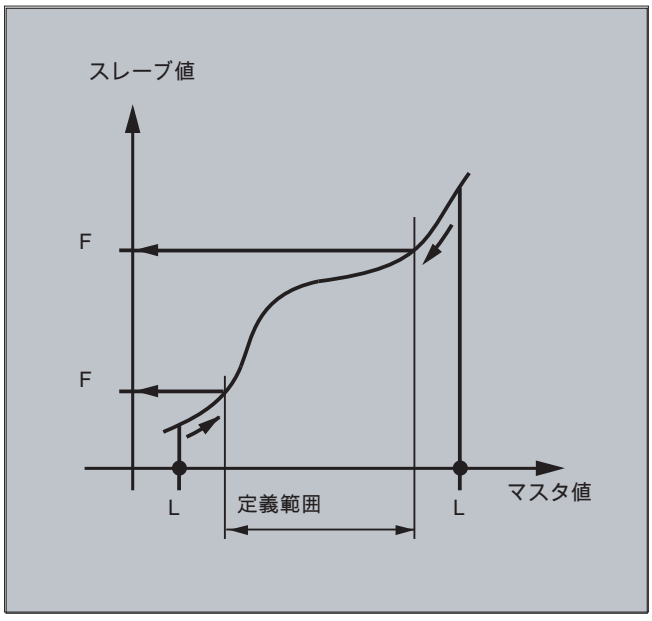
カーブテーブル値を読み取るための命令はすべて、シンクロナイズドアクションでも使用できます(章「シンクロナイズドアクション」も参照してください)。

CTABINV、CTABTMIN、および CTABTMAX の各命令の使用時には、以下を確認してください。

- 実行時には充分な NC 処理能力が使用できる。
  または
- カーブテーブルのセグメントの数を呼び出し前に確認しており、当該カーブテーブルが、必要に応じて分割可能である。

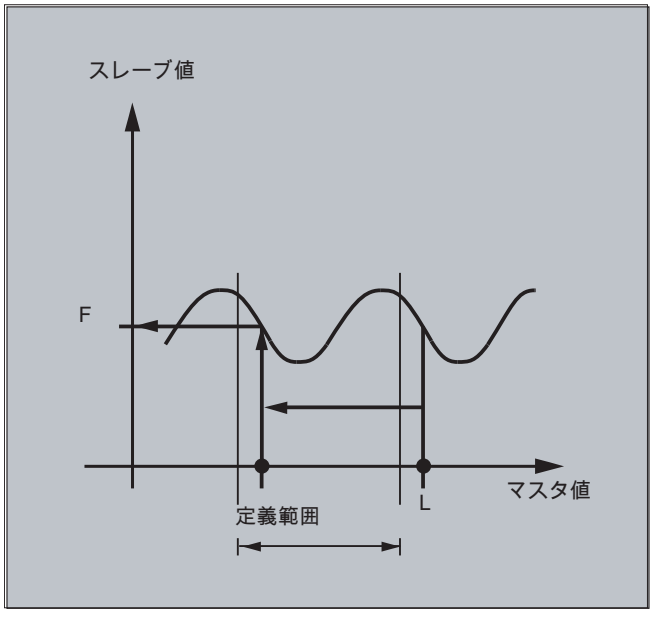
周期的でないカーブテーブルによる CTAB

指定した<マスタ値>が定義範囲外にある場合は、上限または下限がスレーブ値として出力されます。



周期的なカーブテーブルによる CTAB

指定した<マスタ値>が定義範囲外にある場合は、マスタ値は定義範囲で使用されたモジュロとなり、対応するスレーブ値が出力されます。



CTABINV の近似値

したがって、CTABINV 命令には、予想されるマスタ値の近似値が必要となります。CTABINV は、近似値に最も近いマスタ値を返します。近似値には、以前の補間クロックサイクルのマスタ値などがあります。

カーブテーブル機能の傾斜

出力された傾斜(<傾き>)により、対応する位置のマスタ軸またはスレーブ軸の速度の計算が可能となります。

マスタ軸またはスレーブ軸の指定

マスタ軸とスレーブ軸を異なる長さの単位で設定している場合は、マスタ軸とスレーブ軸のいずれか、または両方の選択指定が必要です。

CTABSSV、CTABSEV

CTABSSV と CTABSEV 命令は、以下の場合は、プログラム指令されたセグメントを確認するためには**不適切**です。

- 円弧またはインボリュート曲線をプログラム指令している。
- CHF/ RND による面取りまたは丸み付けが有効である。
- G643 によるスムージングが有効である。
- COMPON/COMPCURV/COMPCAD による NC ブロック圧縮が有効である。

13.2.7 カーブテーブル: メモリ使用の確認(CTABNO、CTABNOMEM、CTABFNO、CTABSEGID、CTABSEG、CTABFSEG、CTABMSEG、CTABPOLID、CTABPOL、CTABFPOL、CTABMPOL)

プログラマは、これらの命令を使用して、カーブテーブル、そのセグメント、および多項式のメモリ使用に関する最新情報を得ることができます。

構文

```
CTABNO
CTABNOMEM(<メモリロケーション>)
CTABFNO(<メモリロケーション>)
CTABSEGID(<n>,<メモリロケーション>)
CTABSEG(<メモリロケーション>,<セグメントタイプ>)
CTABFSEG(<メモリロケーション>,<セグメントタイプ>)
CTABMSEG(<メモリロケーション>,<セグメントタイプ>)
CTABPOLID(<n>)
CTABPOL(<メモリロケーション>)
CTABFPOL(<メモリロケーション>)
CTABMPOL(<メモリロケーション>)
```

意味

CTABNO:	**定義した**(スタティック、およびダイナミック NC メモリの)カーブテーブルの合計数を特定します。	
CTABNOMEM:	指定した<メモリロケーション>の、**定義した**カーブテーブルの数を特定します。	
CTABFNO:	指定した<メモリロケーション>で**引き続き有効な**カーブテーブルの数を特定します。	
CTABSEGID:	番号<n>のカーブテーブルに使用する、指定した<セグメントタイプ>のカーブセグメントの数を特定します。	
CTABSEG:	指定した<メモリロケーション>で**使用**される、指定した<セグメントタイプ>のカーブセグメントの数を特定します。	
CTABFSEG:	指定した<メモリロケーション>で**引き続き有効な**、指定した<セグメントタイプ>のカーブセグメントの数を特定します。	
CTABMSEG:	指定した<メモリロケーション>での指定した<セグメントタイプ>のカーブセグメントの**最大可能**数を特定します。	
CTABPOLID:	番号<n>のカーブテーブルが使用するカーブ多項式の数を特定します	
CTABPOL:	指定した<メモリロケーション>で**使用される**カーブ多項式の数を特定します	
CTABFPOL:	指定した<メモリロケーション>で**引き続き有効な**カーブ多項式の数を特定します	
CTABMPOL:	指定した<メモリロケーション>で**最大**のカーブ多項式数を特定します	
<n>:	カーブテーブルの番号(**ID**)	
<メモリロケーション>:	メモリロケーションの指定(任意選択)	
	"SRAM"	**スタティック** NC メモリ
	"DRAM"	**ダイナミック** NC メモリ
	注: このパラメータに値をプログラム指令しない場合は、MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE で設定した既定のメモリロケーションを使用します。	

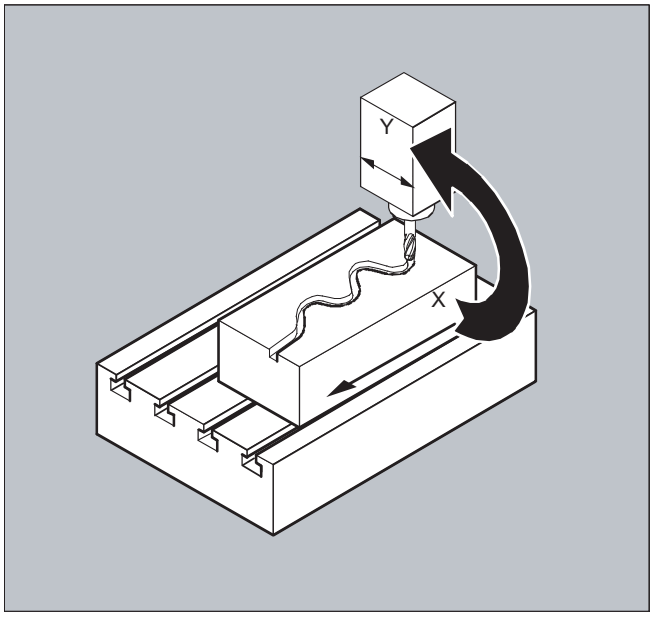
<セグメントタイプ>:	セグメントタイプの指定(任意選択)	
	"L"	直線セグメント
	"P"	多項式セグメント
	注: このパラメータに値をプログラム指令しない場合は、直線セグメントと多項式セグメントの合計が出力されます。	

13.3 軸間連動機能(LEADON、LEADOF)

注記

この機能は SINUMERIK 828D では使用できません。

軸間連動機能を使用すると、マスタ軸とスレーブ軸が同期して移動します。カーブテーブルを使用して、スレーブ軸の位置、または得られる多項式を、マスタ軸の位置に一義的に割り当てることができます。また、必要に応じて、そのシミュレーションも実行できます。



マスタ軸とは、カーブテーブルの入力値を与える軸です。**スレーブ軸**は、カーブテーブルを使用して計算した位置を使用する軸です。

現在値連結と指令値連結

スレーブ軸の位置計算の出力値として、以下をマスタ軸として使用できます。

- マスタ軸位置の現在値:現在値連結
- マスタ軸位置の指令値:指令値連結

軸間連動機能は常に、基本座標系で適用されます。

カーブテーブルの作成については、「カーブテーブル」の章を参照してください。

構文

```
LEADON(<スレーブ軸>,<マスタ軸>,<n>)
```

LEADOF(<スレーブ軸>,<マスタ軸>)

または、マスタ軸を指定せずに解除する場合は:
LEADOF(<スレーブ軸>)

軸間連動機能は、パートプログラムから、および動作中のシンクロナイズドアクションからのいずれでも起動/解除ができます。

意味

LEADON:	軸間連動機能を起動します
LEADOF:	軸間連動機能を解除します
<スレーブ軸>:	スレーブ軸
<マスタ軸>:	マスタ軸
<n>:	カーブテーブル番号
$SA_LEAD_TYPE:	指令値連結と現在値連結の切り替え

軸間連動機能の解除、LEADOF

軸間連動機能を解除すると、スレーブ軸が再び、通常のコマンド軸となります。

軸間連動機能、および様々な運転状態、RESET

マシンデータの設定によっては、軸間連動機能が RESET で解除されます。

シンクロナイズドアクションによる軸間連動機能の例

プレス工場では、マスタ軸(駆動軸)、および搬送軸と補助軸から構成される搬送システムの軸間の通常の機械的連結に代わって、電子連結システムに置き換えられています。

ここでは、機械的搬送システムに代わって電子的搬送システムを使用する方法を示します。連結とその解除の処理は、**内部的なシンクロナイズドアクション**として実行されます。

マスタ軸 LV (駆動軸)から、カーブテーブルで定義したスレーブ軸として搬送軸と補助軸を制御します。

スレーブ軸

X 送りまたは長手方向軸
YL 閉軸または径方向軸
ZL 上昇軸
U ロール送り、補助軸

V ガイドヘッド、補助軸
W 注油、補助軸

動作

実行される動作には、以下のシンクロナイズドアクションなどがあります。

- 連結の起動、LEADON(<スレーブ軸>,<マスタ軸>,<カーブテーブル番号>)
- 連結の解除、LEADOF(<スレーブ軸>,<マスタ軸>)
- 現在値の設定、PRESETON(<軸>,<値>)
- マークの設定、$AC_MARKER[i]=<値>
- 連結タイプ:実/仮想マスタ値
- 軸位置へのアプローチ、POS[<軸>]=<値>

条件

高速ディジタル入力部、リアルタイム変数$AC_MARKER および位置比較演算をブール演算子 AND で連結し、条件として使用します。

注記

以下の例では、プログラムを読み取りやすくするためにのみ、改行、インデント、および**太字**タイプを使用しています。制御装置の場合、行番号の後はすべて 1 行であると見なされます。

コメント

プログラムコード	コメント
	; すべての内部的なシンクロナイズドアクションを定義します。
	; ****マーカのリセット
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 $AC_MARKER[7]=0	
	; **** E1 0=>1 転送オン
N10 IDS=1 EVERY ($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) $AC_MARKER[0]=1	
	; **** E1 0=>1 連結ローラ送りオン
N20 IDS=11 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[5]==0) DO LEADON(U,LW,4) PRESETON(U,0) $AC_MARKER[5]=1	
	; **** E1 0=>1 ガイドヘッド連結オン
N21 IDS=12 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[6]==0) DO LEADON(V,LW,4) PRESETON(V,0) $AC_MARKER[6]=1	
	; **** E1 0=>1 注油連結オン
N22 IDS=13 EVERY ($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 連結オフ

プログラムコード	コメント

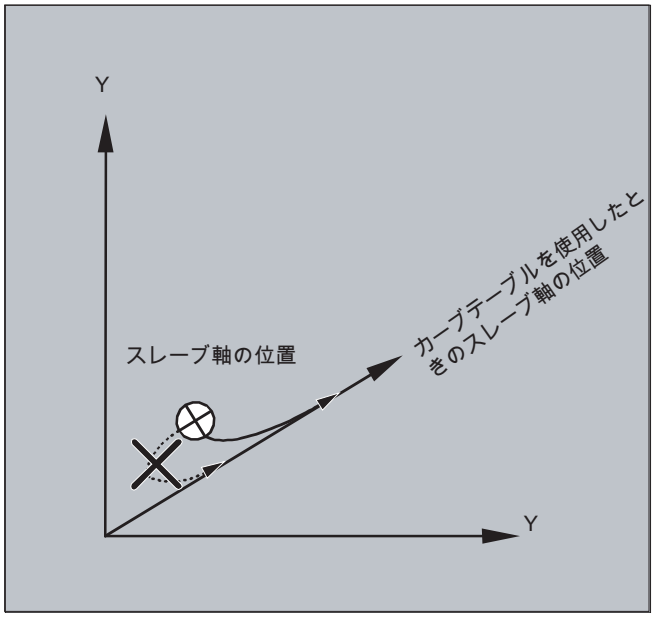
```
N30 IDS=3 EVERY ($A_IN[2]==1)
DO LEADOF(X,LW) LEADOF(YL,LW) LEADOF(ZL,LW) LEADOF(U,LW) LEADOF(V,LW) LEADOF(W,LW) $AC_MARKER[0]=0
$AC_MARKER[1]=0 $AC_MARKER[3]=0 $AC_MARKER[4]=0 $AC_MARKER[5]=0 $AC_MARKER[6]=0 $AC_MARKER[7]=0
....
N110 G04 F01
N120 M30
```

説明

軸間連動機能には、マスタ軸とスレーブ軸の同期制御が必要です。この同期制御は、軸間連動機能が有効なときに、スレーブ軸がカーブテーブルから計算されたカーブ定義の許容範囲内にある場合にのみ可能となります。

スレーブ軸位置の許容範囲は、マシンデータ MD 37200:COUPLE_POS_POL_COARSE A_LEAD_TYPE で定義します。

軸間連動機能が有効なときに、スレーブ軸がまだ正しい位置にない場合は、スレーブ軸に対して計算された位置指令値が実際のスレーブ軸位置とほぼ同じになると、直ちに同期制御の実行が自動的に開始されます。同期制御処理中に、スレーブ軸の指令速度(CTAB カーブテーブルを使用してメイン主軸から計算されます)で定義された方向にスレーブ軸を移動します。

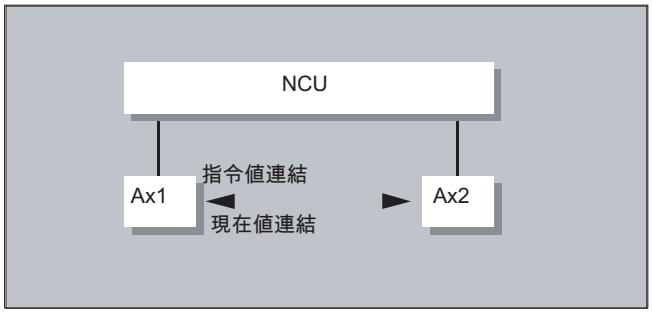


同期制御なし

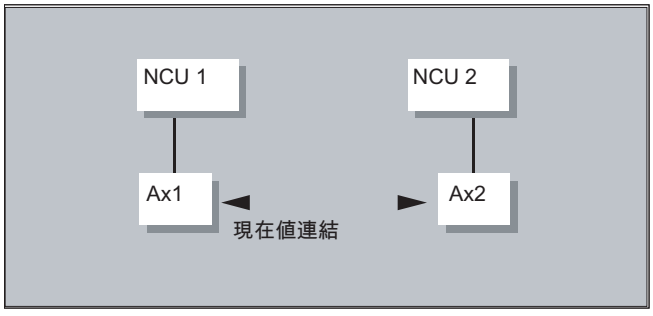
軸間連動機能が有効なときに、計算されたスレーブ軸位置が現在のスレーブ軸位置から離れていく場合は、同期制御を確立できなくなります。

現在値連結と指令値連結

マスタ軸とスレーブ軸の同期制御は、指令値連結の方が現在値連結より優れているため、初期設定は指令値連結となっています。



指令値連結は、マスタ軸とスレーブ軸が同じ NCU によって補間される場合にのみ可能です。外部のマスタ軸を使用する場合は、現在値でのみ、スレーブ軸をマスタ軸に連結できます。



切り替えは、セッティングデータ$SA_LEAD_TYPE を使用してプログラム指令できます。

現在値連結と指令値連結間の切り替えは、必ずスレーブ軸の停止時におこなってください。切り替え後の再原点同期は、軸が停止状態になったときのみ可能です。

用途例

機械が大きく振動しているときは、現在値には誤差を含みます。軸間連動機能をプレス搬送で使用する場合は、最も振動が大きい作業ステップでは、現在値連結から指令値連結への切り替えが必要となる場合があります。

指令値連結による**マスタ値シミュレーション**

マシンデータを使用して、マスタ軸の補間器とサーボの接続を切断できます。この方法で、実際にはマスタ軸を移動せずに、指令値連結の指令値を生成できます。

指定値連結から生成したマスタ値は、以下の変数から読み取って、シンクロナイズドアクションなどで使用できます。

- $AA_LEAD_P　　マスタ値の位置
- $AA_LEAD_V　　マスタ値の速度

マスタ値の作成

マスタ値を他の自動プログラム指令方式で生成できるオプションもあります。このようにして生成したマスタ値は、以下の変数から読み取ったり、これらの変数に書き込んだりできます。

- $AA_LEAD_SP　　マスタ値の位置
- $AA_LEAD_SV　　マスタ値の速度

これらの変数を使用する前に、セッティングデータ$SA_LEAD_TYPE = 2 を設定してください。

連結の状態

NC プログラムで次のシステム変数を使用すると、連結の状態を確認できます。

```
$AA_COUP_ACT[[軸]]
```
0:動作中の連結はありません
16:軸間連動機能が動作中です

シンクロナイズドアクションの状態管理

事象の切り替えと連結は、以下のリアルタイム変数で管理します。

```
$AC_MARKER[i] = n
```
以下を使用して管理します。
i フラグ番号
n 状態値

13.4 電子ギヤ(EG)

「電子ギヤ」機能を使用すると、**スレーブ軸**の移動を、5 つまでの**マスタ軸**に応じて、移動ブロックに従って制御できます。各マスタ軸とスレーブ軸の関係は連結係数で定義します。

スレーブ軸移動部分は、個々のマスタ軸移動部分をそれぞれの連結係数で乗算し、それらを加算して算出します。EG 軸グループが有効になると、定義位置に対してスレーブ軸を同期できます。ギヤグループには、次の動作を実行できます。

- 定義
- 有効化
- 無効化
- 削除

.

任意選択で、スレーブ軸移動を以下から計算することもできます。

- マスタ軸の指令値、および
- マスタ軸の現在値

各マスタ軸とスレーブ軸の非直線の関係は、**カーブテーブル**を使用した拡張機能として実現することもできます(「軌跡移動動作」の章を参照してください)。電子ギヤは多段にできます。つまり、電子ギヤのスレーブ軸が、その他の電子ギヤのマスタ軸となる場合があります。

13.4.1 電子ギヤの定義(EGDEF)

EG 軸グループは、スレーブ軸および、当該の連結タイプで少なくとも 1 つ以上、最大5つのそれぞれのマスタ軸を指定して定義します。

必要条件

EG 軸グループの定義の必要条件は次のとおりです。

スレーブ軸の軸連結は定義できません(または、既存の軸連結は最初に EGDEL で解除してください)。

構文

```
EGDEF(スレーブ軸,マスタ軸 1,連結タイプ 1,マスタ軸 2,連結タイプ 2,...)
```

意味

EGDEF:	電子ギヤの定義	
スレーブ軸:	マスタ軸に影響される軸	
マスタ軸 1 ,..., マスタ軸 5	スレーブ軸に影響する軸	
連結タイプ 1 ,..., 連結タイプ 5	連結タイプ 連結タイプは、すべてのマスタ軸で同じである必要はありません。また、それぞれのマスタ軸毎に個々にプログラム指令してください。	
	値:	**意味**
	0	スレーブ軸は、対応するマスタ軸の**現在値**に影響されます。
	1	スレーブ軸は、対応するマスタ軸の**指令値**に影響されます。

注記

連結係数は、EG 軸グループを定義すると、ゼロに設定されます。

注記

EGDEF は、先読み停止をおこないます。また、システムで、複数のマスタ軸が**カーブテーブル**によってスレーブ軸に影響を及ぼす場合には、EGDEF によるギヤの定義をそのまま使用してください。

例

プログラムコード	コメント
EGDEF(C,B,1,Z,1,Y,1)	; EG 軸グループの定義です。マスタ軸 B、Z、Y は指令値によって、スレーブ軸 C に影響を及ぼします。

13.4.2 電子ギヤの起動(EGON、EGONSYN、EGONSYNE)

EG 軸グループを起動する方法は 3 つあります。

構文

タイプ 1:

以下を使用して、選択した EG 軸グループを同期せずに起動します。
```
EGON(FA,"ブロック切り替えモード",LA1,Z1,N1,LA2,Z2,N2,...,LA5,Z5,N5)
```

タイプ 2:

以下を使用して、選択した EG 軸グループを同期して起動します。
```
EGONSYN(FA,"ブロック切り替えモード",SynPosFA,[,LAi,SynPosLAi,Zi,Ni])
```

タイプ 3:

以下で指定したアプローチモードで、選択した EG 軸グループを同期して起動します。
```
EGONSYNE(FA,"ブロック切り替えモード",SynPosFA,アプローチモード
[,LAi,SynPosLAi,Zi,Ni])
```

意味

タイプ 1:

FA	スレーブ軸	
ブロック切り替えモード:	以下のモードを使用できます。	
	"NOC"	ブロック切り替えは直ぐに実行されます
	"FINE"	ブロック切り替えは「精密同期」で実行されます
	"COARSE"	ブロック切り替え「汎用同期」で実行されます
	"IPOSTOP"	ブロック切り替えが、指令値に基づいた同期制御に対して実行されます
LA1, ... LA5	マスタ軸	
Z1, ... Z5	連結係数 i の分子	
N1, ... N5	連結係数 i の分母 連結係数 i =分子 i /分母 i	

起動行にプログラム指令できるのは、以前に EGDEF 命令で指定したマスタ軸のみです。
少なくとも 1 つのマスタ軸をプログラム指令してください。

タイプ 2:

FA	スレーブ軸	
ブロック切り替えモード:	以下のモードを使用できます。	
	"NOC"	ブロック切り替えは直ぐに実行されます
	"FINE"	ブロック切り替えは「精密同期」で実行されます
	"COARSE"	ブロック切り替え「汎用同期」で実行されます
	"IPOSTOP"	ブロック切り替えが、指令値に基づいた同期制御に対して実行されます
[,LAi,SynPosLAi,Zi,Ni]	(角括弧は記述不要です) 以下を最小 1 個、最大 5 個、順に記述します。	
LA1, ... LA5	マスタ軸	
SynPosLAi	i 番目のマスタ軸の同期位置	
Z1, ... Z5	連結係数 i の分子	
N1, ... N5	連結係数 i の分母 連結係数 i =分子 i /分母 i	

起動行にプログラム指令できるのは、以前に EGDEF 命令で指定したマスタ軸のみです。スレーブ軸とマスタ軸に対してプログラム指令した「同期位置」(SynPosFA と SynPosLA)を使用して位置が定義され、この位置に対して軸グループが*同期している*と解釈されます。グループの起動時に電子ギヤが同期状態でない場合は、スレーブ軸が、定義されたその同期位置に移動します。

タイプ 3:

タイプ 2 のパラメータに対応するパラメータ、および以下のパラメータがあります。

アプローチモード:	以下のモードを使用できます。	
	"NTGT"	時間を優先して、次の歯間へアプローチします
	"NTGP"	距離を優先して、次の歯間へアプローチします
	"ACN"	負の方向にアブソリュートで回転軸を移動します
	"ACP"	正の方向にアブソリュートで回転軸を移動します
	"DCT"	プログラム指令の同期位置に対して時間を優先します
	"DCP"	プログラム指令した同期位置に対して距離を優先します

タイプ 3 は、モジュロマスタ軸に連結されたモジュロスレーブ軸にのみ有効です。時間を優先する場合は、スレーブ軸の速度制限が考慮されます。

詳細情報

起動タイプの説明

タイプ 1:

グループが起動した瞬間のマスタ軸とスレーブ軸の位置が、「同期位置」として格納されます。「同期位置」はシステム変数$AA_EG_SYN で読み取ることができます。

タイプ 2:

連結グループにモジュロ軸が含まれる場合は、その位置の値に対してモジュロ演算がおこなわれます。これにより、次の同期位置(次の歯間などへアプローチすることを*相対同期*と呼びます)に移動できます。同期位置へアプローチするのは、「スレーブ軸オーバライド有効」インタフェース信号 DB(30 +軸番号)、DBX 26 ビット 4 がスレーブ軸に対して設定された場合のみです。その代わりに、プログラムは EGONSYN ブロックで停止して、上記の信号が設定されるまで自己クリアアラーム 16771 が発生します。

タイプ 3:

歯間距離(°)は、「360 * Zi/Ni」のように計算されます。スレーブ軸が呼び出し時に停止した場合は、距離優先の場合と、時間優先の場合は同じ応答を返します。

スレーブ軸がすでに移動中の場合は、NTGP は、スレーブ軸の現在の速度にかかわらず、次の歯間で同期をおこないます。スレーブ軸がすでに移動中の場合は、NTGT は、スレーブ軸の現在の速度に応じて、次の歯間で同期が異なります。必要に応じて、軸の減速もおこなわれます。

カーブテーブル

マスタ軸のいずれかに**カーブテーブル**を使用する場合は、次のようになります。

Ni　直線連結では、連結係数の分母を 0 に設定してください(直線連結では分母 0 は別の意味を持ちます)。分母がゼロの場合は、

ZI　がカーブテーブルの番号であることを指令します。指定した番号のカーブテーブルは常に、POWER ON 時に定義されます。

LAI　指定したマスタ軸は、連結係数によって連結用に指定したマスタ軸に対応します(直線連結)。

カーブテーブルの使用、および電子ギヤの多段と同期制御について詳しくは、以下を参照してください。

参照先:
総合機能説明書 応用機能; 連結軸および ESR (M3)、「連結移動および軸間連動機能」

Power On、RESET、操作モード変更、ブロックサーチでの電子ギヤの動作

- POWER ON 後には、連結は**無効**です。
- 動作中の連結の状態は、RESET にも操作モードの切り替えにも影響されません。
- ブロックサーチのときは、電子ギヤの切り替え、解除、および定義をおこなう命令は、実行も記録もされずにスキップされます。

電子ギヤのシステム変数

電子ギヤのシステム変数を使用して、パートプログラムで EG 軸グループの現在の状態を特定し、必要に応じて、それに対応できます。

電子ギヤのシステム変数は、次のように指定します。

$AA_EG_ ...

または

$VA_EG_ ...

参照先:
システム変数一覧表

13.4.3 電子ギヤの起動(EGOFS、EGOFC)

動作中の EG 軸グループを解除する方法は 3 つあります。

プログラミング

タイプ 1:

構文	意味
EGOFS (スレーブ軸)	電子ギヤが解除されます。スレーブ軸が減速し、停止状態となります。この呼び出しは、先読み停止をおこないます。

タイプ 2:

構文	意味
EGOFS(従動軸、マスタ軸 1、 …, マスタ軸 5)	この命令パラメータを設定すると、スレーブ軸移動への個々のマスタ軸の影響を**選択して**、解除できます。

少なくとも 1 つのマスタ軸を指定してください。指定したマスタ軸のスレーブ軸への影響は、選択して禁止されます。この呼び出しは、先読み停止をおこないます。呼び出しにまだ動作中のマスタ軸が含まれている場合、スレーブ軸はその影響で動作し続けます。すべてのマスタ軸の影響が、この方法により取り除かれた場合は、スレーブ軸が減速して、停止状態となります。

タイプ 3:

構文	意味
EGOFC(スレーブ主軸 1)	電子ギヤが解除されます。スレーブ主軸が、解除時に適用される速度で移動し続けます。この呼び出しは、先読み停止をおこないます。

注記

このタイプは、主軸にのみ使用できます。

13.4.4 電子ギヤの定義の解除(EGDEL)

EG 軸グループは、その定義が解除される前に解除してください。

プログラミング

構文	意味
EGDEL(スレーブ軸)	軸グループの連結定義を解除します。EGDEF を使用して、同時に起動する軸グループの最大数に達するまで、追加の軸グループを定義できます。この呼び出しは、先読み停止をおこないます。

13.4.5 毎回転送り速度(G95) / 電子ギヤ(FPR)

FPR 命令を使用すると、電子ギヤのスレーブ軸を、毎回転送り速度を特定する軸として指定できます。この命令に関しては、以下の点に注意してください。

- 送り速度は、電子ギヤのスレーブ軸の指令速度で特定されます。
- 指令速度は、マスタ主軸とモジュロ軸(軌跡軸でない)の速度、およびそれらに関連する連結係数から計算されます。
- 直線または非モジュロのマスタ軸の速度部分、およびスレーブ軸の重畳移動は考慮されません。

13.5 主軸同期

同期運転には、スレーブ主軸(FS)とマスタ主軸(LS)が関連し、これは**主軸同期ペア**と呼ばれます。連結が有効であるとき(同期運転)、定義されている機能の相互関係に従って、スレーブ主軸はマスタ主軸の移動をコピーします。

各機械の主軸同期ペアは、チャネルマシンデータで固定設定として割り当てるか、または CNC パートプログラムで特定の用途のために定義することができます。各 NC チャネルで同時に運転できる主軸同期ペアは 2 組までです。

パートプログラムからの連結動作は、以下のとおりです。

- 定義または変更
- 有効化
- 無効化
- 解除

.

また、ソフトウェア条件に応じて、以下が適用されます。

- 同期条件が満たされるまで待機できます。
- ブロック切り替え方式を変更できます。
- 指令値連結または現在値連結のいずれかのタイプを選択するか、またはメイン主軸とスレーブ主軸の間の角度オフセットを指定します。
- 連結を起動すると、それ以前にプログラム指令したスレーブ軸で連結をおこないます。
- 検出された同期の不一致、または既知の同期の不一致は補正されます。

13.5.1 主軸同期: プログラミング(COUPDEF、COUPDEL、COUPON、COUPONC、COUPOF、COUPOFS、COUPRES、WAITC)

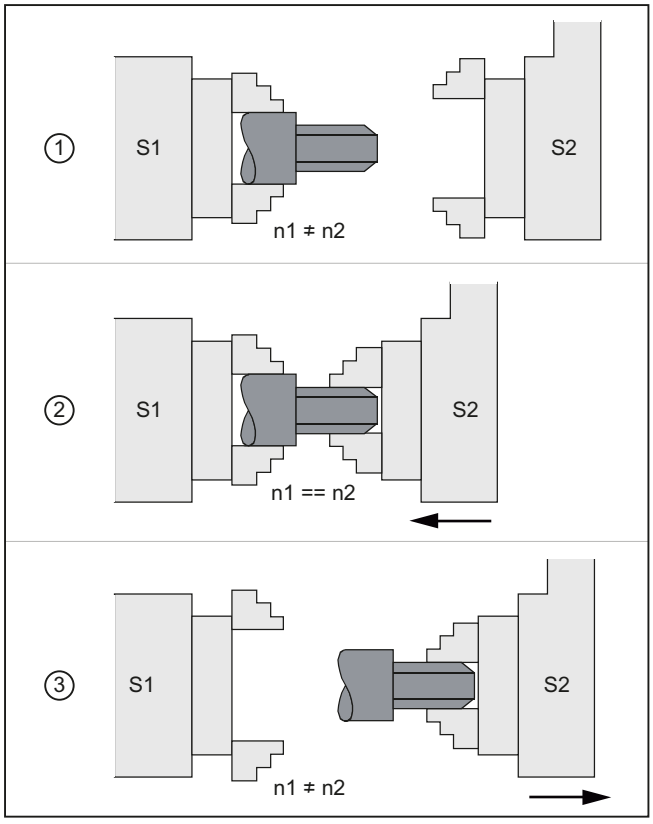
「主軸同期」では、スレーブ主軸(FS)とマスタ主軸 (LS)の速度同期移動をプログラム指令可能な伝達比でおこなうことができます。

この機能は次のモードをサポートしています。

- 速度同期制御($n_{FS} = n_{LS}$)
- 位置同期制御($\phi_{FS} = \phi_{LS}$)
- 角度オフセットによる位置同期制御($\phi_{FS} = \phi_{LS} + \Delta\phi$)

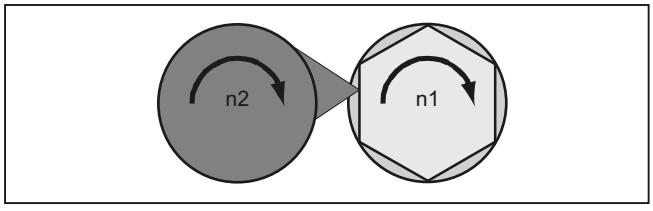
用途例:

- 背面加工などのための高速ワーク移動。伝達比:1:1



① 速度の同期化
② ワークの移動
③ 背面の加工

- 多角形加工(ポリゴン加工)、速度同期制御。伝達比:n_1:n_2



構文

```
COUPDEF(<FS>,<LS>,<ZFS>,<NLS>,<ブロック切り替え>,<連結タイプ>)
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>,<ブロック切り替え>,<LS>,<ブロック切り替え>)
```

注記

略式表記

COUPOF、COUPOFS、COUPRES、および COUPDEL の各命令には、マスタ主軸を指定しない省略表記を使用できます。

意味

COUPDEF:	ユーザー用に連結を定義/変更します
COUPON:	連結を起動します。スレーブ主軸が、現在の速度に基づいてマスタ主軸に同期します
COUPONC:	以前にプログラム指令した M3 S...または M4 S...による起動時に連結をおこないます。 スレーブ主軸の速度の差は、直ちに処理されます。
COUPOF:	連結を解除します。 ● 即時のブロック切り替え: COUPOF(<S2>,<S1>) ● <POSFS>または<POSLS>解除位置を通過後にのみブロック切り替え: COUPOF(<S2>,<S1>,<POSFS>) COUPOF(<S2>,<S1>,<POSFS>,<POSLS>)
COUPOFS:	スレーブ主軸の停止で連結を解除します。 即時のブロック切り替えにより、高速でブロック切り替えをおこないます: COUPOFS(<S2>,<S1>) 解除位置の通過後にのみブロック切り替えをおこないます: COUPOFS(<S2>,<S1>,<POSFS>)

COUPRES:	連結パラメータを、構成のマシンデータとセッティングデータに再設定します。	
COUPDEL:	ユーザー定義の連結を解除します	
WAITC:	同期実行条件 (ブロック切り替え時に NOC が IPO に復帰します)が満たされるまで待機します	
<FS>:	スレーブ主軸を指定します	
任意選択パラメータ:		
<LS>:	マスタ主軸を指定します 主軸番号で指定します例: S2, S1	
<ZFS>, <NLS>:	FS と LS の間の伝達比。 <ZFS>/<NLS> = 分子/分母 初期設定: <ZFS>/<NLS> = 1.0 ; 任意選択の分母の指定	
<ブロック切り替え>:	ブロック切り替え動作 ブロック切り替えは次のようにおこなわれます。	
	"NOC"	即時に実行する
	"FINE"	「精密同期」へ到達時に実行する
	"COARSE"	「汎用同期」へ到達時に実行する
	"IPOSTOP"	IPOSTOP への到達時; つまり、指令値に基づいた同期制御後に実行する(初期設定)
	ブロック切り替え動作はモーダルに有効です。	
<連結タイプ>:	連結タイプ:FS と LS の連結	
	"DV"	指令値連結(初期設定)
	"AV"	現在値連結
	"VV"	速度連結
	連結タイプはモーダルです。	
<POSFS>:	マスタ主軸とスレーブ主軸の間の角度オフセット	
	値の範囲:	0°… 359.999°
<POSFS>,<POSLS>:	スレーブ主軸とマスタ主軸の解除位置 「ブロック切り替えは、POS_{FS}, POS_{LS} を通過すると有効になります」	
	値の範囲:	0°… 359.999°

例

マスタ主軸とスレーブ主軸による加工

プログラムコード	コメント
	マスタ主軸=メイン主軸=主軸 1
	スレーブ主軸=主軸 2
N05 M3 S3000 M2=4 S2=500	マスタ主軸は 3000 rpm で、 スレーブ主軸は 500 rpm で回転します。
N10 COUPDEF(S2,S1,1,1,"NOC","Dv")	連結の定義 (設定も可能です)。
...	
N70 SPCON	マスタ主軸を閉ループ位置制御にします (指令値連結)。
N75 SPCON(2)	スレーブ主軸を閉ループ位置制御にします。
N80 COUPON (S2, S1, 45)	オフセット位置への高速連結= 45°。
...	
N200 FA [S2] = 100	位置決め速度= 100°/min
N205 SPOS[2] = IC(-90)	負方向に 90°重畳して移動します。
N210 WAITC(S2、"Fine")	「精密」同期を待機します。
N212 G1 X...Y...F...	加工
...	
N215 SPOS[2] = IC(180)	正方向に 180°重畳して移動します。
N220 G4 S50	ドウェル時間=メイン主軸の 50 回転に相当します
N225 FA [S2] = 0	設定速度を有効にします (マシンデータ)。
N230 SPOS[2] = IC (-7200)	20 回転。設定速度で負方向に移動します。
...	
N350 COUPOF (S2、S1)	高速連結解除、S=S2=3000
N355 SPOSA[2] = 0	0°で FS を停止します。
N360 G0 X0 Y0	
N365 WAITS(2)	主軸 2 を待機します。
N370 M5	FS を停止します。
N375 M30	

速度差のプログラミング

プログラムコード	コメント
	マスタ主軸=メイン主軸=主軸 1
	スレーブ主軸=主軸 2
N01 M3 S500	マスタ主軸が 500 rpm で回転します。
N02 M2=3 S2=300	スレーブ主軸が 300 rpm で回転します。
...	
N10 G4 F1	メイン主軸のドウェル時間です。
N15 COUPDEF(S2,S1,-1)	連結係数の割合 -1:1
N20 COUPON(S2,S1)	連結を起動します。スレーブ主軸の速度が、マスタ主軸の速度と連結係数から得られます。
...	
N26 M2=3 S2=100	速度差のプログラミング。

速度差に対する移動処理の例

1.COUPON を使用して、これ以前にプログラム指令したスレーブ主軸の連結を起動

プログラムコード	コメント
	マスタ主軸=メイン主軸=主軸 1
	スレーブ主軸=主軸 2
N05 M3 S100 M2=3 S2=200	マスタ主軸は 100 rpm で、スレーブ主軸は 200 rpm で回転します。
N10 G4 F5	ドウェル時間=メイン主軸の 5 秒に相当します。
N15 COUPDEF(S2,S1,1)	FS から LS への伝達比は 1.0 です(初期設定)。
N20 COUPON(S2,S1)	マスタ主軸へ高速連結します。
N10 G4 F5	スレーブ主軸が 100 rpm で回転します。

2.COUPONC を使用して、これ以前にプログラム指令したスレーブ主軸の連結の起動

プログラムコード	コメント
	マスタ主軸=メイン主軸=主軸 1
	スレーブ主軸=主軸 2
N05 M3 S100 M2=3 S2=200	マスタ主軸は 100 rpm で、スレーブ主軸は 200 rpm で回転します。
N10 G4 F5	ドウェル時間=メイン主軸の 5 秒に相当します。
N15 COUPDEF(S2,S1,1)	FS から LS への伝達比は 1.0 です(初期設定)。
N20 COUPONC(S2,S1)	マスタ主軸に高速連結し、以前の速度を S2 に転送します。
N10 G4 F5	S2 が 100 rpm + 200 rpm = 300 rpm で回転します

3.COUPON による、停止状態のスレーブ主軸との連結の起動

プログラムコード	コメント
	マスタ主軸=メイン主軸=主軸 1
	スレーブ主軸=主軸 2
N05 SPOS=10 SPOS[2]=20	スレーブ主軸 S2 を位置決めモードにします。
N15 COUPDEF(S2,S1,1)	FS から LS への伝達比は 1.0 です(初期設定)。
N20 COUPON(S2,S1)	マスタ主軸へ高速連結します。
N10 G4 F1	連結を起動し、S2 が 20°で停止します。

4.COUPONC による、停止状態のスレーブ主軸との連結の起動

注記

位置決めモードまたは軸モード

スレーブ主軸が連結前に位置決めモードまたは軸モードの場合、スレーブ主軸は、COUPON(<FS>,<LS>) および COUPONC(<FS>,<LS>) の場合と同様に動作します。

注記

マスタ主軸と軸運転

連結を定義する前にマスタ主軸が軸運転をおこなっている場合は、マシンデータ MD32000 $MA_MAX_AX_VELO (最大軸速度)による速度制限値が、連結が有効になった後にも適用されます。

この動作を回避するには、連結を定義する前に、軸を主軸モード(M3 S...または M4 S...)に切り替えてください。

詳細情報

連結設定

連結設定の場合、LS と FS をマシンデータで定義します。主軸設定は、パートプログラムでは変更できません。連結はパートプログラムで、COUPDEF を使用して設定できます(ただし、書き込み保護は無効にしてください)。

ユーザー定義連結

COUPDEF を使用すると、パートプログラムで連結を再定義または変更できます。連結がすでに有効な場合は、新しい連結を定義する前に、最初に COUPDEL で有効な連結を解除してください。

連結は、以下を使用して全体を定義します。

COUPDEF(<FS>,<LS>,<TFS>,<TLS>, ブロック切り替え動作, 連結タイプ)

スレーブ主軸(FS)とマスタ主軸(LS)

連結は、FS と LS の軸名称を使用して一意に定義します。軸名称は、COUPDEF 命令毎にプログラム指令してください。他の連結パラメータはモーダルであるため、それらを変更する場合にのみプログラム指令してください。

例:

```
COUPDEF(S2,S1)
```

伝達比

伝達比は、FS と LS 間の速度比として定義します。

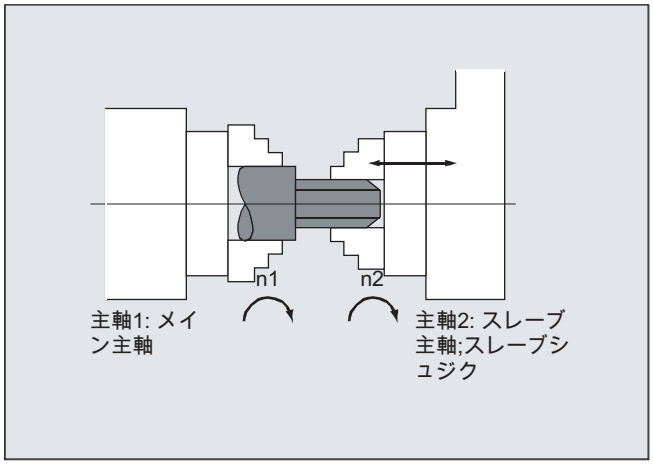
スレーブ主軸/マスタ主軸 = 分子/分母

分子をプログラム指令し、分母はプログラム指令しないでください。初期値 1.0 が分母に設定されます。

例:

スレーブ主軸 S2 とマスタ主軸 S1、伝達比= 1/1 の場合の例です。

```
COUPDEF(S2, S1, 1.0)
```



注記

伝達比は、(連結の動作中で各主軸が回転中の場合でも)、変更できます。

ブロック切り替え動作 NOC、FINE、COARSE、IPOSTOP

ブロック切り替え動作のプログラミング時には、以下の略式表記を使用できます。

- "NO":即時に実行する(初期設定)
- "FI":「精密同期」へ到達時に実行する
- "CO":「汎用同期」へ到達時に実行する
- "IP":IPOSTOP への到達時; つまり、指令値に基づいた同期制御後に実行する

連結のタイプ

注記

連結タイプは、連結を解除したときにのみ変更できます。

同期モードの起動　COUPON, <POSFS>

- LS と FS の間の任意の角度オフセットでの連結の起動:
  - COUPON(S2,S1)
  - COUPON(S2)
- 角度オフセット<POSFS>による連結の起動
  <POSFS>、正方向へ回転ではマスタ主軸の 0°を基準とします。
  <POSFS>値の範囲:0°… 359,999°
  - COUPON(S2,S1,30)

注記

角度オフセットは、連結が有効なときにも変更できます。

スレーブ主軸の位置決め

主軸同期連結が有効の場合でも、FS を LS とは無関係に±180°の範囲に位置決めすることができます。

- SPOS による FS の主軸位置決め
  例:SPOS[2] = IC(-90)
  SPOS について詳しくは、以下を参照してください。
  参照先:
  プログラミングマニュアル 基本編

速度差

速度差の結果、速度制御モードとなり、LS の移動による FS 速度と主軸のプログラム指令による FS 速度との符号付き重畳により、 主軸同期結合が有効になります。

- COUPONC による主軸同期連結
- S<FS>=<速度> [M<FS>=<回転方向>]

注記

必要条件

- 速度 S...も、回転方向 M3/M4 でプログラム指令します。
- 主軸同期結合 COUPONC による、LS の移動を介した主軸速度の重畳(M <回転方向> S<FS>)は、重畳が有効になっている場合にだけ有効になります。
- マスタ主軸のダイナミック応答は、スレーブ主軸への重畳の適用時に、その応答性制限値を超えないような範囲に制限してください。

速度差について詳しくは、以下を参照してください。
参照先:
機能マニュアル 上級機能; 主軸同期(S3)

速度、加速度:FA、ACC、OVRA、VELOLIMA

スレーブ主軸の軸速度と軸加速度は、以下を使用してプログラム指令できます。

- FA[SPI(S<n>)]または FA[S<n>] (軸の速度)
- ACC[SPI(S<n>)]または ACC[S<n>] (軸の加速度)
- OVRA[SPI(S<n>)]および OVRA[S<n>] (軸のオーバライド)
- VELOLIMA[SPI(S<n>)]および VELOLIMA[S<n>] (軸の速度の加速と減速)

ここでは n> = 1、2、3、... (スレーブ主軸の主軸番号)の場合です

参照先:
プログラミングマニュアル 基本編

注記

最大軸加々速度の減速または加速は、主軸には無効です。
軸のダイナミック応答について詳しくは、以下で説明しています。
参照先:
機能マニュアル、上級機能; 回転軸(R2)

プログラム指令可能なブロック切り替え動作 WAITC

WAITC を使用すると、連結パラメータまたは位置決め動作への変更後などに、さまざまな同期条件(汎用、精密、IPOSTOP)によってブロック切り替え動作を定義できます。同期条件を指定しない場合は、COUPDEF 定義で指定したブロック切り替え動作が適用されます。

例

- スレーブ主軸 S2 に対する同期条件 FINE、およびスレーブ主軸 S4 に対する同期条件 COARSE が満たされるまで待機する場合:WAITC(S2,"FINE",S4,"COARSE")
- COUPDEF に従って同期条件が満たされるまで待機する場合 WAITC( )

連結の解除 COUPOF

COUPOF を使用すると、連結の解除動作を、以下のように定義できます。

- 即時のブロック切り替えにより連結の解除をおこなう:
  - COUPOF(S2,S1) (マスタ主軸を指定する)
  - COUPOF(S2) (マスタ主軸を指定しない)
- 解除位置通過後に連結を解除します。ブロック切り替えは、解除位置を通過した後に実行されます。
  - COUPOF(S2,S1,150) (解除位置 FS:150°)
  - COUPOF(S2,S1,150,30) (解除位置 FS:150°、LS:30°)

スレーブ主軸停止による連結の解除 COUPOFS

COUPOFS を使用すると、スレーブ主軸停止による連結の解除動作を、以下のように定義できます。

- スレーブ主軸停止と即時のブロック切り替えにより連結の解除をおこなう:
  - COUPOFS(S2,S1) (マスタ主軸を指定する)
  - COUPOFS(S2) (マスタ主軸を指定しない)
- 解除位置通過後に、スレーブ主軸停止により連結を解除します。ブロック切り替えは、解除位置を通過した後に実行されます。
  - COUPOFS(S2,S1,150) (解除位置 FS:150°)

連結の解除 COUPDEL

COUPDEL は以下のように、連結を解除します。

- COUPDEL(S2,S1) (マスタ主軸を指定する)
- COUPDEL(S2) (マスタ主軸を指定しない)

連結パラメータのリセット、COUPRES

COUPRES は以下のように、 マシンデータとセッティングデータでパラメータ設定した連結値を有効にします。

- COUPRES(S2,S1) (マスタ主軸を指定する)
- COUPRES(S2) (マスタ主軸を指定しない)

システム変数

- スレーブ主軸の現在の連結状態
  スレーブ主軸の現在の結合状態は、以下のビット指定値で読み取ることができます。
  <値> = $AA_COUP_ACT[<FS>]

ビット	<値>	意味
-	0	連結は無効
2	4	主軸同期連結が有効

注

- 上記以外のすべての値は、軸モードを表しています。
- 主軸がスレーブ主軸または複数の連結の場合、すべての連結の結合状態の値が全体の状態として返されます。

- 現在の角度オフセット
  スレーブ主軸とマスタ主軸との間の現在の角度オフセットは、以下で読み取ることができます。
  - $AA_COUP_OFFS[<FS>] (指令値側の角度オフセット)
  - $VA_COUP_OFFS[<FS>] (現在値側の角度オフセット)

  用途例
  フォローアップモードのキャンセル後に、NC プログラムで角度オフセットの差分を修正
  角度オフセットの差分 = プログラム指令された角度オフセット - システム変数

参照先

システム変数に関する詳細情報は、次の関連資料にあります。

「リストマニュアル、システム変数一覧表」

13.6 汎用連結(CP...)

「汎用連結」は、既存の結合タイプ(連結移動、軸間連動機能、電子ギヤ、主軸同期).のすべての連結特性を組み合わせる汎用連結機能です。

この機能を使用して、フレキシブルなプログラム指令がおこなえます。

- ユーザーは、自分のアプリケーションに必要な結合特性を選択できます(ブロック構築原理)。
- それぞれの結合特性を個別にプログラム指令できます。
- 定義された結合の結合特性(結合係数など)を変更できます。
- 追加の結合特性を後から使用できます。
- スレーブ軸の座標系基準システム(基本座標系または機械座標系)をプログラム指令できます。
- 特定の結合特性をシンクロナイズドアクションでもプログラム指令できます。
  参照先:『機能マニュアル、シンクロナイズドアクション』

注記

結合移動(TRAIL*)、軸間連動機能(LEAD*)、電子ギヤ(EG*)、主軸同期(COUP*)の以前の結合の呼び出しは適応サイクルでサポートします。

すべてのキーワードと結合特性の一覧

下の表に、汎用結合のすべてのキーワードとプログラム指令可能な結合特性の一覧を示します。

キーワード	連結特性/意味	構文
CPDEF	連結モジュールの作成	CPDEF=(<FAx>)
CPDEL	連結モジュールの削除	CPDEL=(<FAx>)
CPLA	マスタ軸の定義	CPLA[<FAx>]=(<LAx>)
CPLDEF	マスタ軸の定義と連結モジュールの作成 (CPDEF + CPLA でも可能)	CPLDEF[<FAx>]=(<LAx>) または CPDEF=(<FAx>) CPLA[<FAx>]=(<LAx>)
CPLDEL	連結モジュールのマスタ軸の削除 (CPDEF + CPLA でも可能)	CPLDEL[<FAx>]=(<LAx>) または CPDEL=(<FAx>) CPLA[<FAx>]=(<LAx>)

キーワード	連結特性/意味	構文		
CPON	連結モジュールのスイッチオン	CPON=(<FAx>)		
CPOF	連結モジュールのスイッチオフ	CPOF=(<FAx>)		
CPLON	連結モジュールのマスタ軸のスイッチオン	CPLON[<FAx>]=<LAx>		
CPLOF	連結モジュールのマスタ軸のスイッチオフ	CPLOF[<FAx>]=<LAx>		
CPLNUM	連結係数の分子	CPLNUM[FAx,LAx]=<値>		
CPLDEN	連結係数の分母	CPLDEN[FAx,LAx]=<値>		
CPLCTID	カーブテーブルの番号	CPLCTID[FAx,LAx]=<値>		
CPLSETVAL	連結基準	CPLSETVAL[FAx,LAx]="<連結基準>"		
		"<連結基準>":	"CMDPOS"	指令値連結
			"CMDVEL"	速度連結
			"ACTPOS"	現在値連結
CPFRS	座標系基準システム	CPFRS[FAx]="<座標系基準>"		
		"<座標系基準>":	"BCS"	基本座標系
			"MCS"	機械座標系

キーワード	連結特性/意味	構文		
CPBC	ブロック切り替え条件	CPBC[FAx]="<ブロック切り替え条件>"		
		"<ブロック切り替え条件>":	"NOC"	ブロック切り替えは、連結状態とは無関係に実行されます。
			"IPOSTOP"	ブロック切り替えは、指令値同期制御で実行されます。
			"COARSE"	ブロック切り替えは現在値同期制御「汎用」で実行されます。
			"FINE"	ブロック切り替えは現在値同期制御「精密」で実行されます。
CPFPOS + CPON	連結をオンにしたときのスレーブ軸の同期位置	CPON=FAx CPFPOS[FAx]=<値>		
CPLPOS + CPON	連結をオンにしたときのマスタ軸の同期位置	CPLPOS[FAx,LAx]=<値>		

キーワード	連結特性/意味	構文		
CPFMSON	同期モード	CPFMSON[FAx]="<同期モード>"		
		"<同期モード>":	"CFAST"	連結は時間を優先して接続されます。
			"CCOARSE"	連結は、連結規則に従って必要なスレーブ軸位置が、現在のスレーブ軸位置の範囲内にある場合にのみ接続されます。
			"NTGT"	時間を優先して、次の歯間へアプローチします。
			"NTGP"	軌跡を優先して、次の歯間へアプローチします。
			"NRGT"	歯数に対するギア数の比率に従い、時間を優先して次のセグメントにアプローチします。
			"NRGP"	歯数に対するギア数の比率に従い、軌跡を優先して次のセグメントにアプローチします。
			"ACN"	回転軸の場合のみ 回転軸が、負の軸方向の同期位置に移動します。すぐに同期制御がおこなわれます。
			"ACP"	回転軸の場合のみ 回転軸が、正の軸方向の同期位置に移動します。すぐに同期制御がおこなわれます。
			"DCT"	回転軸の場合のみ

キーワード	連結特性/意味	構文		
				回転軸が、時間を優先してプログラム指令された同期位置に移動します。すぐに同期制御がおこなわれます。
			"DCP"	回転軸の場合のみ 回転軸が、軌跡を優先してプログラム指令された同期位置に移動します。すぐに同期制御がおこなわれます。
CPFMON	スイッチオンでのスレーブ軸の動作	CPFMON[FAx]= "<スイッチオンの動作>"		
		"<スイッチオンの動作>":	"STOP"	主軸の場合のみ スレーブ主軸の動作中の動作がスイッチオンの前に停止します。
			"CONT"	主軸とメイン移動軸の場合のみ スレーブ軸/主軸の現在の動作が、開始動作として連結に引き継がれます。
			"ADD"	主軸の場合のみ 現在の重畳移動に加えられた連結動作の動作要素、つまり、スレーブ軸/主軸の現在の動作が現在の 重畳移動として保持されます。

キーワード	連結特性/意味	構文		
CPFMOF	すべてスイッチオフでのスレーブ軸の動作	CPFMOF[FAx]="<スイッチオフの動作>"		
		"<スイッチオフの動作>":	"STOP"	スレーブ軸/主軸の停止。 動作中の重畳動作も停止状態まで減速します。その後、連結が解除されます。
			"CONT"	主軸とメイン移動軸の場合のみ スレーブ主軸が、解除時に適用される回転数/速度で移動し続けます。
CPFPOS + CPOF	スイッチオフ時のスレーブ軸のスイッチオフ位置	CPOF=(FAx) CPFPOS[FAx]=<値>		

キーワード	連結特性/意味	構文		
CPMRESET	RESET に対する連結動作	CPMRESET[FAx]="<リセット応答>"		
		"<リセット応答>":	"NONE"	連結の現在の状態が保持されます。
			"ON"	適切な連結モジュールが作成されると、連結がスイッチオンされます。定義されているすべてのマスタ軸の関係が有効になります。これは、マスタ軸の関係のすべてまたは一部が有効な場合も実行されます。つまり、連結が完全に有効な状態でも再原点同期が実行されます。
			"OF"	動作中の重畳動作も停止状態まで減速します。その後、連結が解除されます。関連する連結モジュールが明示的な定義(CPDEF)なしで作成されると、連結モジュールは削除されます。そうでない場合は、保持されます。つまり、引き続き使用できます。
			"OFC"	主軸でのみ可能 スレーブ主軸が、解除時に適用される回転数/速度で移動し続けます。連結がスイッチオフされます。関連する連結モジュールが明示的な定義(CPDEF)なしで作成されると、連結モジュールは削除されます。

キーワード	連結特性/意味	構文		
				そうでない場合は、保持されます。つまり、引き続き使用できます。
			"DEL"	動作中の重畳動作も停止状態まで減速します。連結が解除された後、削除されます。
			"DELC"	主軸でのみ可能 スレーブ主軸が、解除時に適用される回転数/速度で移動し続けます。連結が解除された後、削除されます。

キーワード	連結特性/意味	構文		
CPMSTART	パートプログラム開始時の連結動作	CPMSTART[FAx]="<開始動作>"		
		"<開始動作>":	"NONE"	連結の現在の状態が保持されます。
			"ON"	連結のスイッチオン。定義されているすべてのマスタ軸の関係が有効になります。これは、マスタ軸の関係のすべてまたは一部が有効な場合も実行されます。つまり、連結が完全に有効な状態でも再原点同期が実行されます。
			"OF"	連結がスイッチオフされます。関連する連結モジュールが明示的な定義(CPDEF)なしで作成されると、連結モジュールは削除されます。そうでない場合は、保持されます。つまり、引き続き使用できます。
			"DEL"	連結が解除された後、削除されます。
CPMPRT	プログラムテストによるブロック検索実行での、パートプログラム開始時の連結動作	CPMPRT[FAx]="<開始動作>"		
		"<開始動作>":	CPMSTART を参照ください	
CPLINTR	マスタ軸の入力値のオフセット値	CPLINTR[FAx,LAx]=<値>		
CPLINSC	マスタ軸の入力値の単位変換係数	CPLINSC[FAx,LAx]=<値>		
CPLOUTTR	連結の出力値のオフセット値	CPLOUTTR[FAx,LAx]=<値>		

キーワード	連結特性/意味	構文		
CPLOUTSC	連結の出力値の単位変換係数	CPLOUTSC[FAx,LAx]=<値>		
CPSYNCOP	位置同期制御「汎用」の検出値	CPSYNCOP[FAx]=<値>		
CPSYNFIP	位置同期制御「精密」の検出値	CPSYNFIP[FAx]=<値>		
CPSYNCOP2	「汎用」位置同期制御の 2 番目の検出値	CPSYNCOP2[FAx]=<値>		
CPSYNFIP2	「精密」位置同期制御の 2 番目の検出値	CPSYNFIP2[FAx]=<値>		
CPSYNCOV	速度同期制御「汎用」の検出値	CPSYNCOV[FAx]=<値>		
CPSYNFIV	速度同期制御「精密」の検出値	CPSYNFIV[FAx]=<値>		
CPMBRAKE	特定の停止信号と停止命令に対するスレーブ軸の応答	CPMBRAKE[FAx]=<ビット指定値>		
CPMVDI	特定の NC/PLC インタフェース信号に対する、スレーブ軸の応答	CPMVDI[FAx]=<ビット指定値>		
CPMALARM	特別な連結関連アラーム出力のマスク	CPMALARM[FAx]=<ビット指定値>		
CPSETTYPE	連結タイプ	CPSETTYPE[FAx]="<連結タイプ>"		
		"<連結タイプ>":	"CP"	自由にプログラム指令が可能
			"TRAIL"	連結タイプ「連結移動」
			"LEAD"	連結タイプ「軸間連動機能」
			"EG"	連結タイプ「電子ギヤ」
			"COUP"	連結タイプ「同期制御主軸」

FAx:スレーブ軸/主軸

LAx:マスタ軸/主軸

注記

(シンクロナイズドアクションのパートプログラムで)明示的にプログラム指令されていない連結特性は、初期設定で有効になります。

プリセットされた連結特性は、初期設定(CPSETTYPE="CP")の代わりに、キーワード CPSETTYPE の設定に従って有効になります。

参照先

汎用連結について詳しくは、以下を参照してください。

- 機能マニュアル、応用機能; M3:軸連結:「汎用連結」の章

13.7 マスタ/スレーブ連結(MASLDEF、MASLDEL、MASLON、MASLOF、MASLOFS)

「マスタ/スレーブ連結」により、以下のことが可能になります。

- 関与する軸が停止状態の場合は、スレーブ軸をマスタ軸に連結できます。
- **回転中**の速度制御された主軸を連結/連結解除できます。
- ダイナミック設定。

注記

位置決めモード

位置決めモードの軸と主軸の場合は、連結は停止状態のときにのみ、起動と解除がおこなわれます。

構文

```
MASLON(<スレーブ_1>,<スレーブ_2>,...)
MASLOF(<スレーブ_1>,<スレーブ_2>,...)
MASLOFS(<スレーブ_1>,<スレーブ_2>,...)
```

ダイナミック設定

```
MASLDEF(<スレーブ_1>,<スレーブ_2>, ... ,<マスタ>)
MASLDEL(<スレーブ_1>,<スレーブ_2>,...)
```

意味

MASLON:	一時的なマスタ/スレーブ連結の有効化	
	<スレーブ_x>,...:	スレーブ軸 1 ... n
MASLOF:	動作中のマスタ/スレーブ連結の連結解除	
	<スレーブ_1>,...:	スレーブ軸 1 ... n
MASLOFS:	マスタ/スレーブ連結の連結解除とスレーブ主軸の自動的な減速(注「速度制御モードでの主軸の連結動作」を参照してください)	
	<スレーブ_1>,...:	スレーブ軸 1 ... n
MASLDEF:	パートプログラムからのマスタ/スレーブグループの作成/変更	
	<スレーブ_1>,...:	スレーブ軸 1 ... n
	<マスタ>:	マスタ軸

MASLDEL:	マスタ/スレーブ連結の解除とグループの定義の削除	
	<スレーブ_1>, . . .:	スレーブ軸 1 ... n
	注: マシンデータで設定されたマスタ/スレーブ定義は保持されます。	

注記

速度制御モードでの主軸の連結動作

速度制御モードの主軸の場合 MASLON、MASLOF、MASLOFS、および MASLDEL の連結動作は、次のマシンデータで明示的に指定します。

MD37263 $MA_MS_SPIND_COUPLING_MODE

MD37263 = 0 の初期設定の場合は、使用される軸が停止状態のときにのみ、スレーブ軸が連結、および連結解除されます。MASLOFS は MASLOF に対応します。

MD37263 = 1 の場合は、連結命令が直ちに実行され、移動も実行されます。MASLON の場合は、連結が直ちに起動され、MASLOFS または MASLOF の場合は、直ちに解除されます。MASLOF では、このとき回転中のスレーブ主軸は、新しい速度がプログラム指令されるまで、その速度を保持します。ただし、MASLOFS では、自動的に減速します。

注記

MASLOF/MASLOFS の場合は、自動先読み停止はおこないません。先読み停止をおこなわないため、次のプログラミングまで、スレーブ軸の$P システム変数は値が更新されません。

注記

スレーブ軸の場合は、PRESETON を使用して、現在値をマスタ軸の同一値に同期できます。これを行うには、電源投入によって原点確立されていないスレーブ軸の現在値をマスタ/軸の値に設定するために、固定の/スレーブ連結を一時的に解除してください。これで、連結が固定して、再確立されます。

固定のマスタ/スレーブ連結は、次のマシンデータの設定:
MD37262 $MA_MS_COUPLING_ALWAYS_ACTIVE = 1
を使用して起動します。この設定は、一時的な連結のための言語命令には無効です。

例

例 1: マスタ/スレーブ連結のスレーブ軸に現在値を設定します。

固定のマスタ/スレーブ連結をおこなうために、PRESETON でスレーブ軸の現在値をマスタ軸の値に設定します。

プログラムコード	コメント
$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=0	; スレーブ軸の固定連結を解除します。
NEWCONF	; マシンデータの変更を有効にします。
Stopre	
MASLOF(Y1)	; 一時的な連結を解除します
PRESETON(AX2,$VA_IM(M_AX))	; スレーブ軸の現在値 = マスタ軸の現在値
$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=1	; スレーブ軸の固定連結を有効にします。
NEWCONF	;マシンデータの変更を有効にします。

例 2 :マスタ/スレーブ連結のダイナミック設定

軸コンテナ回転後に別の主軸との連結を有効にするには、以前の連結を解除し、設定を消去して、新しい連結を設定してください。

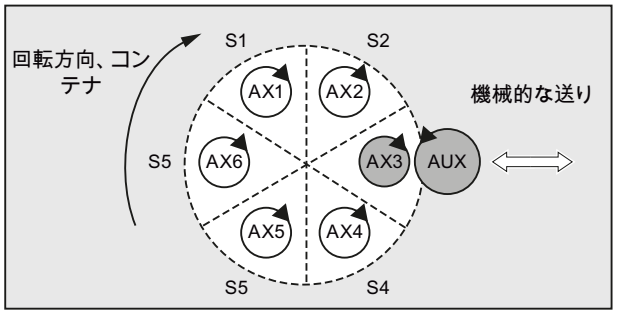


図 13-1　軸コンテナ回転前

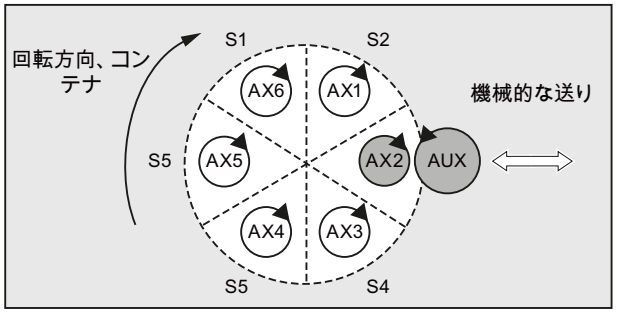


図 13-2　軸コンテナを 1 スロットだけ回転後

プログラムコード	コメント
MASLDEF(AUX,S3)	; AUX:スレーブ、S3:マスタ = AX3
MASLON(AUX)	; 連結のオン

プログラムコード	コメント
M3=3 S3=4000	; マスタを回転します。
MASLDEL(AUX)	; 切断して連結を解除します。
AXCTSWE(CT1)	; 軸コンテナ回転を有効にします。
MASLDEF(AUX,S3)	; AUX:スレーブ、S3:マスタ = AX2

参照先

- 機能マニュアル 応用機能、「TE3:速度/トルク連結、マスタスレーブ」の章
- 機能マニュアル 上級機能、「B3:分散システム - 840D sl のみ」 > 「NCU リンク」 > 「軸コンテナ」の章

シンクロナイズドアクション 14

14.1 シンクロナイズドアクションの定義

シンクロナイズドアクションは、パートプログラムのブロック内で定義されます。シンクロナイズドアクションの一部でないその他のすべての命令は、このブロック内でプログラミングできません。

シンクロナイズドアクションは、以下のコンポーネントから構成されます。

適用範囲、ID 番号 (オプション)	条件部分 (オプション)			アクション部分		
	周波数	G 命令 (オプション)	条件	キーワード	G 命令 (オプション)	動作
--- [1] ID=<no.> IDS=<no.>	--- [1] WHENEVER FROM WHEN EVERY	G...	論理式	DO	G...	アクション 1 ... アクション n

[1] プログラム指令なし

構文

```
DO <アクション 1> ... <アクション n>
<頻度> [<G 機能>] <条件> DO <アクション 1> ... <アクション n>
ID=<番号> <頻度> [<G 機能>] <条件> DO <アクション 1> ... <アクション n>
IDS=<番号> <頻度> [<G 機能>] <条件> DO <アクション 1> ... <アクション n>
```

参照先

シンクロナイズドアクションの機能の詳細は、下記を参照してください。

機能マニュアル、シンクロナイズドアクション

揺動 15

15.1 非同期揺動(OS、OSP1、OSP2、OST1、OST2、OSCTRL、OSNSC、OSE、OSB)

揺動軸は、揺動動作を解除するまで、2 つの反転点 1 と 2 の間を、定義した送り速度で前後に移動します。

その他の軸は、揺動動作中に自由に補間できます。連続切り込みは、軌跡移動または位置決め軸を使用して実行できますが、揺動移動と切り込み移動は**無関係**です。

非同期揺動の機能

- 非同期揺動は、軸毎にブロック境界を越えて有効です。
- パートプログラムを使用して、揺動移動をブロック単位で起動できます。
- 複数の軸の補間の組合せと揺動軌跡の重畳は実行できません。

プログラミング

以下の命令を使用すると、パートプログラムから非同期揺動の起動と制御ができます。

プログラム指令値は、メインラン時のブロック同期制御により、対応するセッティングデータに入力され、再度変更されるまでそのまま有効です。

構文

```
OSP1[<軸>]=<値> OSP2[<軸>]=<値>
OST1[<軸>]=<値> OST2[<軸>]=<値>
FA[<軸>]=<値>
OSCTRL[<軸>]=(<設定オプション>,<リセットオプション>)
OSNSC[<軸>]=<値>
OSE[<軸>]=<値>
OSB[<軸>]=<値>
OS[<軸>] = 1
OS[<軸>] = 0
```

意味

<軸>:	揺動軸の名称		
OS:	揺動を起動/解除します		
	値:	1	揺動を**オン**にします
		0	揺動を**オフ**にします
OSP1:	反転点 1 の位置を定義します		
OSP2:	反転点 2 の位置を定義します **注:** インクレメンタル移動が有効の場合は、NC プログラムでプログラム指令した最新の対応する反転位置まで、位置をインクレメンタルで計算します。		
OST1:	反転点 1 の停止時間を秒単位で定義します		
OST2:	反転点 2 の停止時間を秒単位で定義します		
	<値>:	-2	イグザクトストップを待機せずに補間が継続します
		-1	汎用イグザクトストップを待機します
		0	精密イグザクトストップを待機します
		>0	精密イグザクトストップを待機した後、指定した停止時間だけ待機します **注:** 停止時間の単位は、G4 でプログラム指令した停止時間と同じです。
FA:	送り速度を定義します 送り速度は、位置決め軸の定義送り速度です。送り速度を定義しない場合は、マシンデータに格納された値が適用されます。		

OSCTRL:	設定オプションとリセットオプションを指定します オプション値 0 ～ 3 は、解除時の反転点での動作コードです。0 ～ 3 のいずれかのタイプを選択できます。その他の設定内容は、選択したタイプと自由に組み合わせることができます。複数のオプションをプラス記号(+)で付加します。		
	<値>:	0	揺動の解除時に次の反転点で停止します(初期設定) **注:** 値 1 と 2 をリセットした場合にのみ可能です。
		1	揺動を解除すると、反転点 1 で停止します
		2	揺動を解除すると、反転点 2 で停止します
		3	スパークアウトストロークをプログラム指令していない場合は、揺動を解除するときに反転点へアプローチしません
		4	スパークアウト処理後に終了位置へアプローチします
		8	揺動移動が残移動距離削除によって取り消されたときに、スパークアウトストロークの実行が必要になり、必要に応じて、終了位置へアプローチします。
		16	揺動移動が残移動距離削除によって取り消されたときに、停止の場合と同様に、対応する反転点へのアプローチが必要になります。
		32	新しい切り込みは、次の反転点の後にのみ有効です
		64	FA が 0 と等しい、つまり FA = 0 の場合:軌跡の重畳が有効です FA が 0 と等しくない、つまり FA <> 0 の場合:速度の重畳が有効です
		128	回転軸の DC 指令(最短軌跡)
		256	スパークアウトストロークがデュアルストロークです(初期設定)。1=シングルストローク。
		512	最初に開始位置へアプローチします
OSNSC:	スパークアウトストローク数を定義します		

OSE:	(ワーク座標系の)終了位置を定義し、揺動の解除後にそこへアプローチします。 **注:** 「OSE」をプログラム指令すると、OSCTRL に対してオプション 4 が自動的に有効になります。
OSB:	(ワーク座標系の) 開始位置を定義し、揺動の起動前にそこへアプローチします。 反転点 1 の前に開始位置にアプローチします。開始位置が反転点 1 と同じである場合は、次に反転点 2 にアプローチします。開始位置に到達したとき、この位置が反転点 1 と同一であっても停止時間は適用されず、軸は精密イグザクトストップ信号を待機します。設定されているすべてのイグザクトストップ条件が満たされます。 **注:** セッティングデータ SD43770 $SA_OSCILL_CTRL_MASK のビット 9 は、開始位置へのアプローチを開始するように設定してください。

例

例 1:2 つの反転点間で揺動する揺動軸

揺動軸 Z は、位置 10 と 100 の間で揺動します。反転点 1 は精密イグザクトストップにより、反転点 2 は汎用イグザクトストップにより実行されます。揺動軸の送り速度は 250 です。加工運転の終了時には 3 つのスパークアウトストロークを実行し、揺動軸は終了位置 200 へアプローチします。切り込み軸の送り速度は 1 で、X 方向の切り込みは位置 15 で終了します。

プログラムコード	コメント
WAITP(X,Y,Z)	; 初期設定:
G0 X100 Y100 Z100	; 軸の位置決め運転へ切り替え。
WAITP(X,Z)	
OSP1[Z]=10 OSP2[Z]=100	; 反転点 1、反転点 2。
OSE[Z]=200	; 終了位置:
OST1[Z]=0 OST2[Z]=-1	; U1 の停止時間:精密イグザクトストップ
	; U2 の停止時間:汎用イグザクトストップ
FA[Z]=250 FA[X]=1	; 揺動軸と切り込み軸の速度
OSCTRL[Z]=(4,0)	; オプションの設定
OSNSC[Z]=3	; 3 つのスパークアウトストローク。
OS[Z]=1	; 揺動を起動します。
WHEN $A_IN[3]==TRUE DO DELDTG(X)	; 残移動距離削除。
POS[X]=15	; X 軸の開始位置。

プログラムコード	コメント
POS[X]=50	; X 軸の終了位置。
OS[Z]=0	; 揺動を停止します。
M30	

注記

「OSP1[Z]=...」～「OSNCS[Z]=...」命令処理は、1 つのブロックでプログラム指令することもできます。

例 2 :反転位置のオンライン変更による揺動

非同期の揺動に必要なセッティングデータは、パートプログラムに設定できます。

セッティングデータをプログラムに直接記述した場合、先読みのときに、この変更が有効になります。同期制御動作は、先読み停止(STOPRE)を使用すると、実現できます。

プログラムコード	コメント
$SA_OSCILL_REVERSE_POS1[Z]=-10	
$SA_OSCILL_REVERSE_POS2[Z]=10	
G0 X0 Z0	
WAITP(Z)	
ID=1 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS1[Z] DO $AA_OVR[X]=0	; 揺動軸の現在位置が反転点を超えた場合は、切り込み軸が停止します。
ID=2 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[X]=0	
OS[Z]=1 FA[X]=1000 POS[X]=40	; 揺動を起動します。
OS[Z]=0	; 揺動を解除します。
M30	

詳細情報

揺動軸

揺動軸には以下が適用されます。

- すべての軸を揺動軸として使用できます。
- 複数の揺動軸を同時に有効にすることができます(最大値:位置決め軸数)。
- 揺動軸には、現在プログラムで有効な G 命令とは無関係に、直線補間 G1 が常に有効です。

揺動軸では、以下が可能です。

- ダイナミック座標変換の入力軸として機能します
- ガントリ軸と連結移動軸のガイド軸として機能します
- 以下のように移動します。
  - 加々速度制限なし「BRISK」
    または
  - 加々速度制限あり「SOFT」
    または
  - 膝形加減速カーブで(位置決め軸として)

揺動の反転点

揺動位置を次のように定義する場合は、現在のオフセットを考慮してください。

- アブソリュート指定
  "OSP1[Z]=<値>"
  反転点の位置=オフセットの合計+プログラム指令値
- 相対指定
  "OSP1[Z]=IC(<値>)"
  反転点の位置=反転点 1 +プログラム指令値

例:

プログラムコード

```
N10 OSP1[Z]=100 OSP2[Z]=110
...
N40 OSP1[Z]=IC(3)
```

WAITP

揺動をジオメトリ軸で実行する場合は、「WAITP」でこの軸を揺動のために有効にしてください。

揺動が終了すると、「WAITP」を使用して、揺動軸を位置決め軸として再度入力します。これで、通常の使用を再開できます。

シンクロナイズドアクションと停止時間による揺動

設定停止時間を過ぎると、揺動中に内部ブロック切り替えが実行されます(軸の新しい残移動距離によって示されます)。ブロック切り替え時には、解除機能がチェックされます。解除機能は、動作処理 OSCTRL の制御設定に従って定義されます。*このダイナミック応答は送りオーバライドの作用を受けます。*

スパークアウトストロークが開始されるか、終了位置にアプローチ前に、揺動ストロークが実行されることがあります。*解除動作が変更されたようにみえますが、そうではありません。*

15.2 シンクロナイズドアクションによる揺動制御(OSCILL)

この揺動モードでは、切り込み移動を実行できるのは、反転点または、定義した反転領域内のみです。

揺動移動は、必要条件に応じて、次のように動作します。

- 続行する、または
- 切り込みの実行が終了するまで停止する。

構文

1. **揺動のパラメータを定義する**
2. **シンクロナイズドアクションを定義する**
3. **軸を割り当てて、切り込みを定義する**

意味

OSP1[<揺動軸>]=	反転点 1 の位置
OSP2[<揺動軸>]=	反転点 2 の位置
OST1[<揺動軸>]=	反転点 1 の停止時間(秒単位)
OST2[<揺動軸>]=	反転点 2 の停止時間(秒単位)
FA[<揺動軸>]=	揺動軸の送り速度
OSCTRL[<揺動軸>]=	オプションの設定またはリセット
OSNSC[<揺動軸>]=	スパークアウトストローク回数
OSE[<揺動軸>]=	終了位置
WAITP(<揺動軸>)	揺動のための軸を有効にします

軸割り当て、切り込み
OSCILL[<揺動軸>]=(<切り込み軸 1>,<切り込み軸 2>,<切り込み軸 3>)
POSP[<切り込み軸>]=(<終了位置>,<区間長>,<モード>)

OSCILL:	切り込み軸または軸を揺動軸に割り当てます
POSP:	全体または区間切り込みを定義します(「ファイルとプログラム管理」の章を参照してください)
終了位置:	すべての区間切り込みが移動後の切り込み軸の終了位置

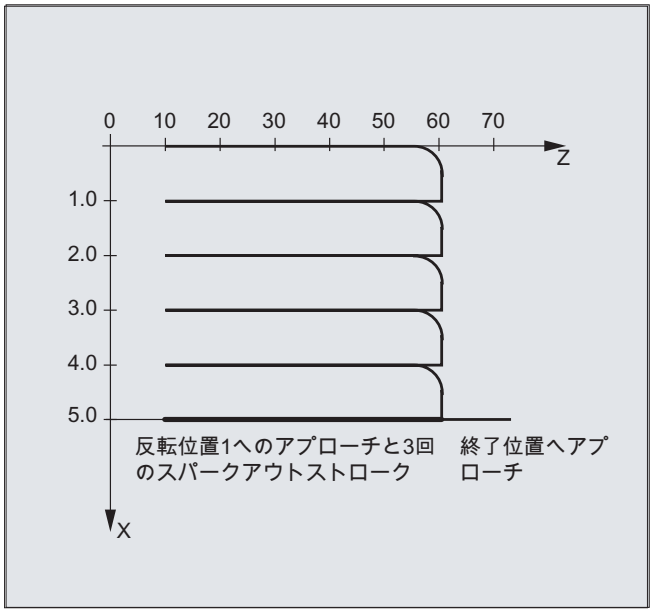
区間長:	反転点/反転領域での区間切り込みの長さ
モード:	切り込み全体を区間切り込みに分割 =残りを同じサイズの 2 区間に分割(初期設定) =すべての区間切り込みを同じサイズに分割

シンクロナイズドアクション

WHEN… … DO	…のときに、…を実行します
WHENEVER … DO	…のときは常に、…を実行します

例

反転点 1 で切り込みを実行しません。反転点 2 では、反転点 2 より距離 ii2 だけ前で切り込みを開始します。揺動軸はこの反転点では区間切り込みの終了を待機しません。Z 軸が揺動軸で、X 軸が切り込み軸です



1.揺動のパラメータ

プログラムコード	コメント
DEF INT ii2	; 反転領域 2 の変数を定義します
OSP1[Z]=10 OSP2[Z]=60	; 反転点 1 と 2 を定義します
OST1[Z]=0 OST2[Z]=0	; 反転点 1:精密イグザクトストップ
	反転点 2: 精密イグザクトストップ
FA[Z]=150 FA[X]=0.5	; 揺動軸 Z の送り速度、切り込み軸 X の送り速度

プログラムコード	コメント
OSCTRL[Z]=(2+8+16.1)	; 反転点 2 で揺動動作を解除します; 残移動距離削除後にスパークアウトして終了位置へアプローチします; 残移動距離削除後に反転位置へアプローチします
OSNC[Z]=3	; スパークアウトストローク
OSE[Z]=70	; 終了位置 = 70
ii2=2	; 反転点範囲を設定します
WAITP(Z)	; Z 軸の揺動を有効にします

2.シンクロナイズドアクション

プログラムコード	コメント
WHENEVER $AA_IM[Z]<$SA_OSCILL_REVERSE_POS2[Z] DO -> $AA_OVR[X]=0 $AC_MARKER[0]=0	; MCS の揺動軸 Z の実位置が反転範囲 2 の開始位置より小さい場合は、切り込み軸 X の軸オーバライドを常に 0%に、インデックス 0 のビットメモリを常に値 0 に設定します。
WHENEVER $AA_IM[Z]>=$SA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[Z]=0	; MCS の揺動軸 Z の実位置が反転位置 2 より大きい場合は、揺動軸 Z の軸オーバライドを常に 0%に設定します。
WHENEVER $AA_DTEPW[X] == 0 DO $AC_MARKER[0]=1	; 区間切り込みの残移動距離が 0 の場合は、インデックス 0 のビットメモリを常に値 1 に設定します。
WHENEVER $AC_MARKER[0]==1 DO $AA_OVR[X]=0 $AA_OVR[Z]=100	; インデックス 0 のビットメモリが 1 に等しい場合は必ず、切り込み軸 X の軸オーバライドを 0%に設定します。その結果、早すぎる切り込みが防止されます (振動軸 Z はまだ、反転領域 2 にありますが、切り込み軸 X で新規切り込みが可能です)。振動軸 Z の軸オーバライドの設定を 0% (2 番目のシンクロナイズドアクションの動作) から 100%に戻して、移動します。

->単独ブロックにプログラム指令してください。

3.揺動の開始

プログラムコード	コメント
OSCILL[Z]=(X) POSP[X]=(5,1,1)	; 軸を起動します 揺動軸 Z を切り込み軸として軸 X に割り当てます。 終了位置 5 まで、軸 X が 1 区間ずつ移動します。
M30	; プログラム終了

詳細情報

1. **揺動パラメータの定義**
   揺動のパラメータは、切り込み軸と揺動軸の割り当て、および切り込みの定義を含む移動ブロックの前に定義してください(「非同期揺動」を参照してください)。
2. **シンクロナイズドアクションの定義**
   次の同期制御条件を定義できます。
   切り込みを、揺動軸が反転領域内
   (ii1、ii2)、または反転点(U1、U2)に到達するまでマスクします。
   反転点での切り込みのときに**揺動動作**を**停止**します。
   区間切り込みの完了と同時に**揺動移動**を**再起動**します。
   次の区間切り込みの起動を定義します。
3. **揺動軸と切り込み軸**の他に、**区間切り込みと全体切り込み**を割り当てます。

揺動パラメータの定義

揺動軸と切り込み軸の割り当て OSCILL;ＯＳＣＩＬＬ

`OSCILL[<揺動軸>]=(<切り込み軸 1>,<切り込み軸 2>,<切り込み軸 3>)`

軸割り当てと揺動移動の起動は、「OSCILL」命令で定義します。

3 軸までの切り込み軸を、1 つの揺動軸に割り当てることができます。

注記

揺動を開始する前に、軸の動作の同期制御条件を定義してください。

切り込みの定義:POSP;ＰＯＳＰ

`POSP[<切り込み軸>]=(<終了位置>,<区間長>,<モード>)`

「POSP」命令で制御装置に以下を宣言します。

- 全体切り込み(終了位置を基準とする)
- 反転点または反転領域での区間切り込みの長さ
- 終了位置への到達時の区間切り込み動作(モードを基準として)

モード= 0	最後の 2 つの区間切り込みの終点までの残移動距離を、2 つの等しい区間に分割します(初期設定)。
モード= 1	すべての区間切り込みは、同じサイズです。これらは、全体切り込みから算出します。

シンクロナイズドアクションの定義

一般的な揺動には、次の一覧に示す、シンクロナイズドアクションを使用します。

個々の処理の解決方法の例は、ユーザー用の揺動移動を作成するモジュールとして使用できます。

注記
それぞれの場合について、異なる同期制御条件をプログラム指令できます。

キーワード

WHEN … DO …	…のときに、…を実行します
WHENEVER … DO	…のときは常に、…を実行します

機能

以下に説明する言語リソースで、次の機能を実行できます。

1. 反転点での切り込み。
2. 反転領域での切り込み。
3. 両方の反転点の切り込み。
4. 反転点で揺動移動の停止。
5. 揺動移動の再起動。
6. 早すぎる区間切り込みの起動の防止。

ここに示すシンクロナイズドアクションのすべての例は、以下を前提としています。

- 反転点 1 < 反転点 2
- Z =揺動軸
- X =切り込み軸

注記
詳しくは、「シンクロナイズドアクション」の章を参照してください。

揺動軸と切り込み軸、および区間切り込みと全体切り込みの割り当て

反転点範囲の切り込み

切り込み移動は、反転点に達する前に反転領域内で開始してください。

これらのシンクロナイズドアクションでは、揺動軸が反転領域内に入るまで、切り込み移動を禁止します。

前述の前提に従って、次の命令を使用します。

反転範囲 1:

```
WHENEVER $AA_IM[Z]>$SA_OSCILL_RESERVE_POS1[Z]+ii1 DO $AA_OVR[X] = 0
```

MCS の揺動軸の実位置が反転範囲の開始位置 1 より大きい場合は常に、切り込み軸の軸オーバライドを 0%に設定してください。

反転範囲 2:

```
WHENEVER $AA_IM[Z]<$SA_OSCILL_RESERVE_POS2[Z]+ii2 DO $AA_OVR[X] = 0
```

MCS での揺動軸の実位置が反転範囲の開始位置 2 より小さい場合は常に、切り込み軸の軸オーバライドを 0%に設定してください。

反転点での切り込み

揺動軸が反転点に達していない限り、切り込み軸は移動しません。

前述の前提に従って、次の命令を使用します。

反転範囲 1:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

MCS での揺動軸 Z の実位置が反転点 1 の位置より大きい、または小さい場合は常に、切り込み軸 X の軸オーバライドを 0%に、揺動軸 Z の軸オーバライドを 100%に設定してください。

反転範囲 2:

反転点 2 の場合:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

MCS での揺動軸 Z の実位置が反転点 2 の位置より大きい、または小さい場合は常に、切り込み軸 X の軸オーバライドを 0%に、揺動軸 Z の軸オーバライドを 100%に設定してください。

反転点で揺動移動を停止

揺動軸は反転点で停止し、同時に切り込み移動を開始します。揺動動作が続行され、切り込み移動が完了します。

まだ動作中の以前のシンクロナイズドアクションが切り込み移動を停止している場合は、同時に、このシンクロナイズドアクションを使用して切り込み移動を開始できます。

前述の前提に従って、次の命令を使用します。

反転範囲 1:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

MCS での揺動軸の実位置が反転位置 1 と同じである場合は常に、揺動軸の軸オーバライドを 0%に、切り込み軸の軸オーバライドを 100%に設定してください。

反転範囲 2:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

MCS での揺動軸 Z の実位置が反転位置 2 と同じである場合は常に、揺動軸 X の軸オーバライドを 0%に、切り込み軸の軸オーバライドを 100%に設定してください。

反転点のオンライン評価

比較演算の右側に$$で割り付られたメインラン変数がある場合は、2 個の変数を相互に、IPO サイクルで連続的に評価と比較をおこないます。

注記

詳細については、「シンクロナイズドアクション」の章を参照してください。

揺動移動の再開

このシンクロナイズドアクションの目的は、区間切り込み移動の完了時に揺動軸の移動を続けることです。

前述の前提に従って、次の命令を使用します。

```
WHENEVER $AA_DTEPW[X]==0 DO $AA_OVR[Z]= 100
```

WCS の切り込み軸 X での区間切り込みの残りの距離が 0 に等しい場合は常に、揺動軸の軸オーバライドを 100%に設定してください。

次の区間切り込み

切り込みの完了時には、次の区間切り込みの起動が早すぎないようにします。

このためには、チャネル別マーク($AC_MARKER[インデックス])を使用します。このマークは、区間切り込みの終了(区間残移動距離≡ 0)時に有効になり、軸が反転領域を出たときに解除されます。次の切り込み移動はシンクロナイズドアクションで防止します。

所定の前提に基づいて、以下の命令が反転点 1 に適用されます。

1.マークの設定:

```
WHENEVER $AA_DTEPW[X] == 0 DO $AC_MARKER[1]=1
```

WCS の切り込み軸 X での区間切り込みの残りの距離が 0 に等しい場合は常に、インデックス 1 のビットメモリを 1 に設定してください。

2.マークの削除

```
WHENEVER $AA_IM[Z]<> $SA_OSCILL_RESERVE_POS1[Z] DO $AC_MARKER[1] = 0
```

MCS の揺動軸 Z の実位置が反転点 1 の位置より大きい、または小さい場合は常に、ビットメモリ 1 を 0 に設定してください。

3.切り込みの禁止

```
WHENEVER $AC_MARKER[1]==1 DO $AA_OVR[X]=0
```

ビットメモリ 1 が同じである場合は常に、切り込み軸 X の軸オーバライドを 0%に設定してください。

パンチングとニブリング 16

16.1 起動と解除

16.1.1 パンチングとニブリングの起動または解除(SPOF、SON、PON、SONS、PONS、PDELAYON、PDELAYOF、PUNCHACC):

パンチングとニブリングの起動/解除

PON と SON を使用して、パンチングとニブリング機能を起動します。SPOF は、すべてのパンチングとニブリング専用の機能を終了します。モーダル命令 PON と SON は、同時には使用できません。つまり、PON は SON を解除し、SON は PON を解除します。

開始制御のあるパンチング/ニブリング

SONS と PONS 機能も、パンチングまたはニブリング機能を起動します。

SON/PON (補間レベルのストローク制御)とは異なり、これらの機能を使用すると、信号に対応したストローク開始制御をサーボレベルで実行します。これにより、ストローク回数が増加し、その結果、パンチング能力が向上します。

開始制御で信号を使用している間は、ニブリング軸またはパンチング軸の位置変更を伴うすべての機能(手動ハンドル移動、PLC によるフレームの変更、計測機能など)が禁止されます。

遅延のあるパンチング

PDELAYON は、パンチングストロークの出力を遅延します。このモーダルの有効命令には準備機能があり、このために、通常は PON の前に位置します。PDELAYOF の後は、通常のパンチングが再開されます。

注記

遅延時間は、セッティングデータ SD42400 $SC_PUNCH_DWELLTIME で設定します。

軌跡依存の加速

PUNCHACC を使用すると、穴の間隔に応じて異なる加速度を定義する加減速特性を指定できます。

2 番目のパンチングインタフェース

2 番目のパンチングインタフェース(もう 1 台のパンチングユニットまたは類似のもの)の使用が必要な機械の場合は、制御装置高速ディジタル入/出力部の 2 番目のペア(I/O ペア)へ、交互に切り替えることができます。どちらのインタフェースでも、パンチング/ニブリング機能をすべて使用できます。SPIF1 と SPIF2 命令を使用して、1 番目と 2 番目のパンチングインタフェースを切り替えます。

注記

必要条件:マシンデータで、2 番目の I/O ペアをパンチング機能に定義してください(→工作機械メーカの仕様書を参照してください)。

構文

```
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
```

意味

PON:	パンチングを起動します。
SON:	ニブリングの起動
PONS:	開始制御のあるパンチングを起動します。
SONS:	開始制御のあるニブリングを起動します。
SPOF:	パンチング/ニブリングを解除します。
PDELAYON:	遅延のあるパンチングを起動します。
PDELAYOF:	遅延のあるパンチングを解除します。

PUNCHACC:	移動に依存した加減速度を起動します。 パラメータ:	
	<Smin>	穴の間隔の最小値
	<Amin>	初期加減速度 <Amin>には、<Amax>より大きい値を指定できます。
	<Smax>	穴の間隔の最大値
	<Amax>	最終加速度 <Amax>には、<Amin>より大きい値を指定できます。
SPIF1:	**1 番目の**パンチングインタフェースを起動します。 高速 I/O の 1 番目のペアを使用してストロークを制御します。	
SPIF2:	**2 番目の**パンチングインタフェースを起動します。 高速 I/O の 2 番目のペアを使用してストロークを制御します。	
	注: 1 番目のパンチングインタフェースは常に、RESET またはコントロールシステムの電源投入の後に有効になります。1 つのパンチングインタフェースのみを使用する場合は、プログラム指令する必要はありません。	

例

例 1:ニブリングの起動

プログラムコード	コメント
...	
N70 X50 SPOF	; パンチングを開始せずに位置決めします。
N80 X100 SON	; ニブリングを起動し、移動 (X=50) の前、およびプログラム指令した移動の完了時 (X=100) にストロークを開始します。
...	

例 2:遅延のあるパンチング

プログラムコード	コメント
...	
N170 PDELAYON X100 SPOF	; パンチングを開始せずに位置決めし、遅延したパンチング開始を起動します。
N180 X800 PON	; パンチングを起動します。パンチのストロークは、終了位置に達すると、遅延されて出力されます。
N190 PDELAYOF X700	; 遅延のあるパンチングを解除し、プログラム指令した移動の完了時に通常のパンチングを開始します。
...	

例 3:2 つのパンチングインタフェースによるパンチング

プログラムコード	コメント
...	
N170 SPIF1 X100 PON	; ブロックの終点で、1 番目の高速出力でストロークを開始します。[ストローク有効]信号が、1 番目の入力で監視されます。
N180 X800 SPIF2	; 2 番目のストロークを、2 番目の高速出力で開始します。[ストローク有効]信号が、2 番目の入力で監視されます。
N190 SPIF1 X700	; その後のストロークはすべて、1 番目のインタフェースで制御されます。
...	

詳細情報

開始制御のあるパンチングとニブリング(PONS/SONS)

開始制御のあるパンチングとニブリングは、複数のチャネルで同時には実行できません。PONS または SONS は、一度に 1 つのチャネルでのみ起動できます。

軌跡依存の加速(PUNCHACC)

例:
PUNCHACC(2,50,10,100)

2 mm 未満の穴間隔:

軸は、最大加速度の 50%のレートで加速します。

2 ～ 10 mm の穴間隔:

加速度は間隔に比例して、100%に増加します。

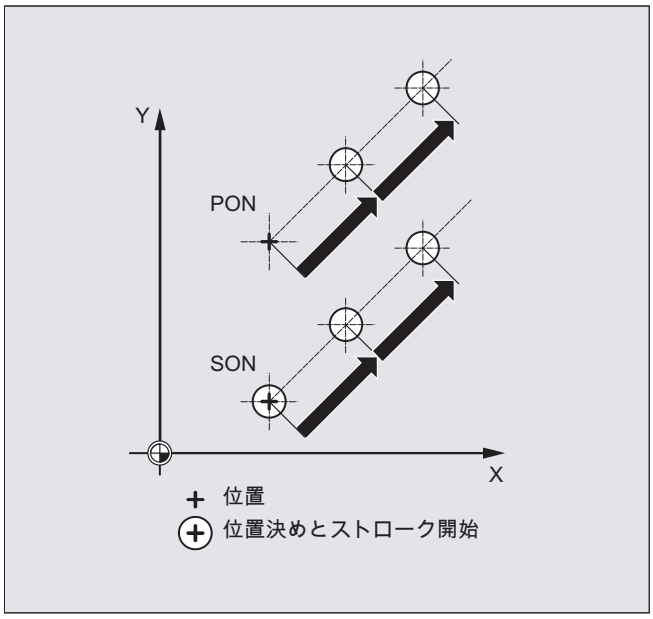
10 mm を超える穴間隔:

100%の加速度で移動します。

最初のストロークの開始

機能の起動後に最初のストロークを開始するタイミングは、ニブリングまたはパンチングの選択に応じて、次のように異なります。

- PON/PONS:
  - すべてのストローク(起動後の最初のブロックのストロークも含む)をブロックの終点で実行します。
- SON/SONS:
  - ニブリング機能の起動後の最初のストロークを、ブロックの始点で実行します。
  - それ以降の各ストロークは、ブロックの終点で開始します。



その時のパンチングとニブリング

パンチング軸またはニブリング軸(有効平面上の軸)の移動情報がブロックに含まれる場合にのみ、ストロークを開始します。

ただし、ストロークを同じ位置で開始するためには、パンチング軸/ニブリング軸の 1 つを移動軌跡 0 でプログラム指令できます。

旋回工具による加工

注記

旋回工具を、プログラム指令軌跡に接して位置決めする場合は、法線方向制御機能を使用してください。

M 命令の使用

以前のバージョンのように、マクロ指令により、言語命令の代わりに特殊な M 機能を使用できます(互換性)。以前のシステムで使用されていた M 機能と同等の言語命令は、次のとおりです。

```
M20、M23  ≙  SPOF
M22       ≙  SON
M25       ≙  PON
M26       ≙  PDELAYON
```

マクロファイルの例:

プログラムコード	コメント
DEFINE M25 AS PON	; パンチング ON
DEFINE M125 AS PONS	; 開始制御のあるパンチングのオン
DEFINE M22 AS SON	; ニブリング ON
DEFINE M122 AS SONS	; 開始制御のあるニブリングのオン
DEFINE M26 AS PDELAYON	; 遅延のあるパンチングのオン
DEFINE M20 AS SPOF	; パンチング、ニブリングのオフ
DEFINE M23 AS SPOF	; パンチング、ニブリングのオフ

プログラミング例:

プログラムコード	コメント
...	
N100 X100 M20	; パンチングを開始せずに位置決めします。
N110 X120 M22	; ニブリングを起動し、移動の前後でストロークを開始します。
N120 X150 Y150 M25	; パンチングを起動し、動作終了時にストロークを開始します。
...	

16.2 自動軌跡分割

軌跡セグメントへの分割

パンチングまたはニブリングを有効にすると、SPP と SPN の両方で、軌跡軸に対してプログラム指令した合計移動区間を、同じ長さの軌跡セグメント数に分割します(等間隔の軌跡分割)。内部的には、各軌跡セグメントがブロックに対応します。

ストローク数

パンチング時には、最初のストロークを最初の軌跡セグメントの終点で実行します。これに対し、ニブリング時には、最初の軌跡セグメントの始点で実行します。したがって、移動区間全体で、以下の数を取得します。

パンチング:ストローク数=軌跡セグメント数

ニブリング:ストローク数=軌跡セグメント数+1

補助機能

補助機能は、生成したブロックの最初で実行されます。

構文

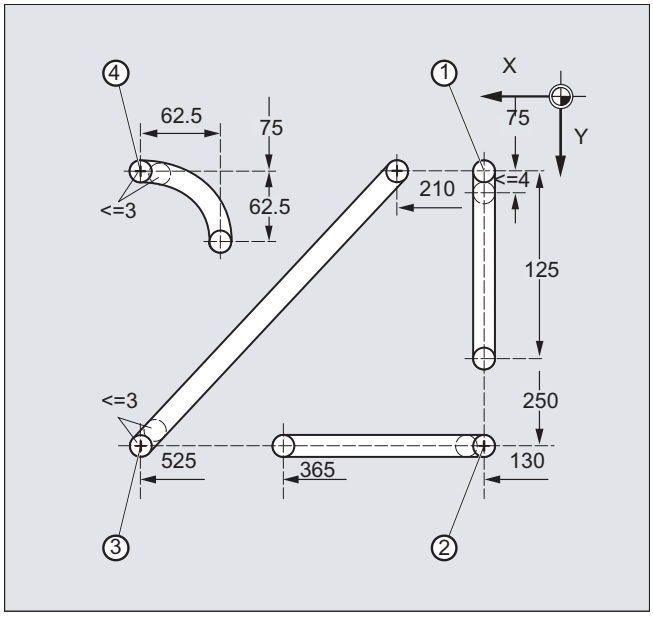
```
SPP=
SPN=
```

意味

SPP:	軌跡セグメントのサイズ(ストロークの最大間隔); モーダル
SPN:	ブロック毎の軌跡セグメント数; モーダルで有効

例 1

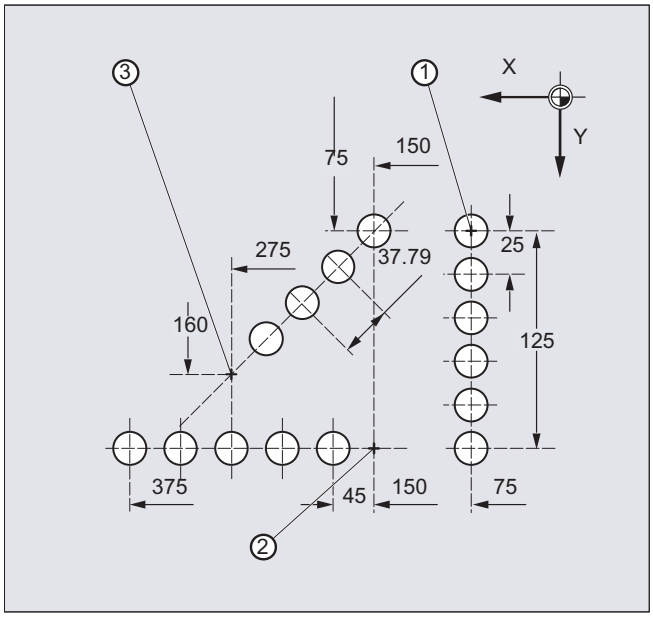
プログラム指令したニブリングセグメントを、自動的に軌跡セグメントに分割します。



プログラムコード	コメント
N100 G90 X130 Y75 F60 SPOF	; 始点 1 での位置決め
N110 G91 Y125 SPP=4 SON	; ニブリングのオン; 自動軌跡分割の最大軌跡セグメント長さ:4 mm
N120 G90 Y250 SPOF	; ニブリングのオフ; 始点 2 への位置決め
N130 X365 SON	; ニブリングのオン; 自動軌跡分割の最大軌跡セグメント長さ:4 mm
N140 X525 SPOF	; ニブリングのオフ; 始点 3 への位置決め
N150 X210 Y75 SPP=3 SON	; ニブリングのオン; 自動軌跡分割の最大軌跡セグメント長さ:3 mm
N160 X525 SPOF	; ニブリングのオフ; 始点 4 への位置決め
N170 G02 X-62.5 Y62.5 I J62.5 SPP=3 SON	; ニブリングのオン; 自動軌跡分割の最大軌跡セグメント長さ:3 mm
N180 G00 G90 Y300 SPOF	; ニブリングのオフ

例 2

自動軌跡分割を、それぞれ連続穴に対しておこないます。最大軌跡セグメント長さ(SPP値)を分割に対して指定します。



プログラムコード	コメント
N100 G90 X75 Y75 F60 PON	; 始点 1 への位置決め; 穴を 1 つあけます
N110 G91 Y125 SPP=25	; 自動軌跡分割の最大軌跡 セグメント長さ:25 mm
N120 G90 X150 SPOF	; パンチングのオフ; 始点 2 への 位置決め
N130 X375 SPP=45 PON	; パンチングのオン;自動軌跡分割の 最大軌跡セグメント長さ:45 mm
N140 X275 Y160 SPOF	; パンチングのオフ; 始点 3 への 位置決め
N150 X150 Y75 SPP=40 PON	; パンチングのオン、プログラム指令した 40 mm の軌跡セグメント長さ ではなく、算出した 37.79 mm の軌跡セグメント長さを使用します。
N160 G00 Y300 SPOF	; パンチングのオフ; 位置決め

16.2.1 軌跡軸の軌跡分割

SPP 軌跡セグメントの長さ

SPP を使用して、各ストロークの最大間隔を指定します。これにより、軌跡セグメントの最大長が指定され、合計移動距離がこの長さに分割されます。この命令は、SPOF または SPP=0 で解除します。

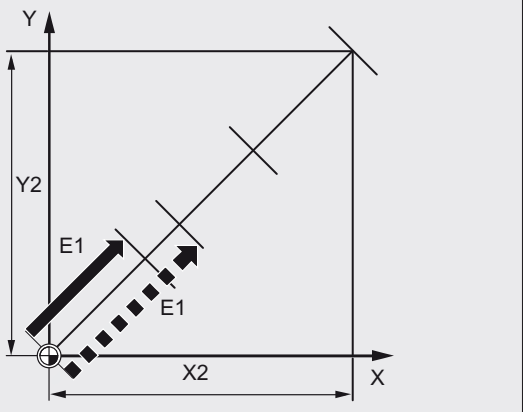
例:

```
N10 SON X0 Y0
N20 SPP=2 X10
```

合計移動距離 10 mm を、それぞれ 2 mm の 5 つの軌跡区間に分割します(SPP=2)。

注記

SPP による軌跡セグメントは常に等間隔です。つまり、すべてのセグメントは等しい長さです。言い換えると、プログラム指令した軌跡セグメントのサイズ(SPP の設定)は、合計移動距離と SPP 値の商が整数の場合にのみ有効です。これに該当しない場合は、商が整数になるように、軌跡セグメントのサイズを内部的に縮小します。



X2/Y2 プログラム指令移動軌跡(ニブリングブロックまたはパンチングブロック)
E1 プログラム指令軌跡セグメント長さ
E1' 自動的に整数化された軌跡セグメント長さ

例:

```
N10 G1 G91 SON X10 Y10
N20 SPP=3.5 X15 Y15
```

合計移動距離が 15 mm で軌跡セグメント長さが 3.5 mm の場合は、商が整数値ではありません(4.28)。この場合は、SPP 値を、次の整数の商に減らします。この例の結果は、3 mm の軌跡セグメント長さです。

SPN の軌跡セグメントの数

SPN は、合計移動距離から生成される軌跡セグメントの数を定義します。セグメントの長さは自動的に計算されます。SPN はノンモーダルであるため、パンチングまたはニブリングはそれぞれ、PON または SON で事前に有効にしておいてください。

16.2.2 単独軸の軌跡分割

軌跡軸の他に、単独軸をパンチング軸/ニブリング軸として定義した場合は、自動軌跡分割機能を単独軸に対して有効にできます。

SPP の単一軸の動作

プログラム指令した軌跡セグメント長さ(SPP)は基本的に、軌跡軸を基準とします。したがって、単一軸運動と SPP 値に加えて軌跡軸がプログラムされているブロックでは、SPP 値は無視されます。

単独軸と軌跡軸の両方をブロックにプログラムした場合、単独軸の動作は、当該のマシンデータの設定によって異なります。

1. 初期設定
   単独軸が移動する軌跡は、SPP で生成された中間ブロックの間で均等に配分されます。

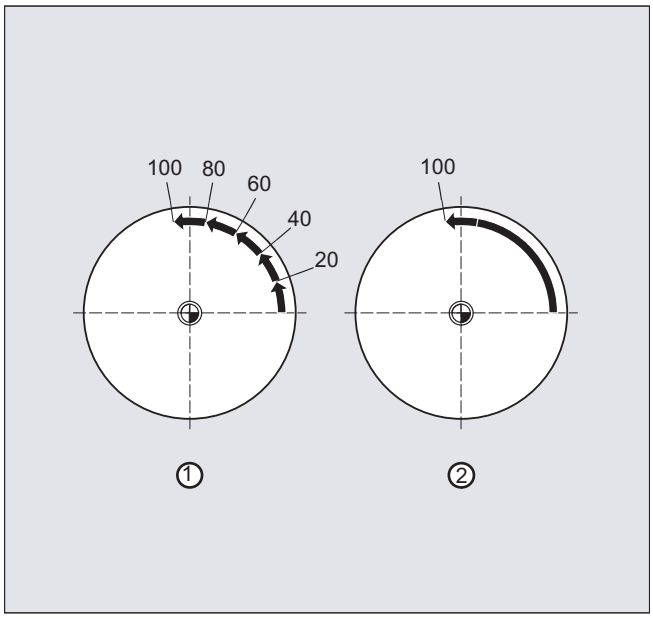
例:

```
N10 G1 SON X10 A0

N20 SPP=3 X25 A100
```

ストローク間隔を 3 mm にした結果として、X 軸(軌跡軸)の合計移動距離 15 mm に対して 5 つのブロックが生成されます。

このため、A 軸がすべてのブロックで 20°回転します。



1. 軌跡分割をおこなわない単独軸
   単独軸は、生成されたブロックのうち、最初のブロックで全距離を移動します。
2. 異なる軌跡分割
   単独軸の動作は、軌跡軸の補間に応じて、次のように異なります。

- 円弧補間:軌跡分割
- 直線補間:軌跡分割を行いません

SPN の動作

プログラムした軌跡セグメント数は、軌跡軸を同時にプログラムしていない場合でも適用されます。

必要条件単独軸をパンチング軸/ニブリング軸として定義します。

研削加工 17

17.1 研削用工具監視の有効化/無効化(TMON、TMOF)

予約手順 TMON(...)および TMOF(...)を使用して、研削用工具監視を有効化または無効化します(ジオメトリおよび速度監視)。

必要条件

工具固有のパラメータ$TC_TPG1 ～$TC_TPG9 を設定してください。

構文

```
TMON(<TNo>)
...
TMOF(<TNo>)
```

意味

TMON(...):	研削用工具監視の有効化 命令は、研削用工具監視が有効になるチャネルでプログラミングしてください。
TMOF(...):	研削用工具監視の無効化 命令は、研削用工具監視が無効になるチャネルでプログラミングしてください。
<TNo>:	T 番号 **注記:** 監視を現在使用されている有効なといしではなく、無効なといしに対してオンまたはオフにする場合にのみ必要です。
TMOF(0):	すべての工具の監視を解除します

その他の機能 18

18.1 軸機能(AXNAME、AX、SPI、AXTOSPI、ISAXIS、AXSTRING、MODAXVAL)

「AXNAME」は、たとえば軸の名称が不明な場合に、一般的に有効なサイクルを生成するために使用します。

「AX」は、ジオメトリ軸と同期軸を間接的にプログラム指令する場合に使用します。軸識別子は、AXIS タイプの変数で保存されるか、「AXNAME」や「SPI」などの命令により提供されます。

「SPI」は、主軸同期などの主軸に軸機能をプログラム指令している場合に使用します。

「AXTOSPI」を使用して、軸識別子を主軸インデックスへ変換します(SPI の逆の機能です)。

「AXSTRING」を使用して、軸識別子(データタイプ AXIS)を文字列に変換します(「AXNAME」の逆の機能です)。

「ISAXIS」を汎用サイクルで使用して、特定のジオメトリ軸が存在できるようにして、後続の$P_AXNX 呼び出しがエラーメッセージで中止されないようにします。

「MODAXVAL」は、モジュロ回転軸のモジュロ位置を特定するために使用します。

構文

```
AXNAME("文字列")
AX[AXNAME("文字列")]
SPI(n)

AXTOSPI(A)またはAXTOSPI(B)またはAXTOSPI(C)
AXSTRING( SPI(n) )
ISAXIS(<ジオメトリ軸番号>)
<モジュロ位置>=MODAXVAL(<軸>,<軸位置>)
```

意味

AXNAME:	入力文字列を軸識別子に変換します。入力文字列には、有効な軸名称を含めてください。
AX:	可変軸識別子

SPI:	主軸番号を軸識別子に変換します。転送パラメータには、有効な主軸番号を含めてください。
n:	主軸番号
AXTOSPI:	軸識別子を整数の主軸インデックスへ変換します。「AXTOSPI」は、SPI の逆の機能になります。
X, Y, Z:	変数または定数としての AXIS タイプの軸識別子
AXSTRING:	関連する主軸番号で文字列を出力します。
ISAXIS:	指定したジオメトリ軸が存在するかどうかを確認します。
MODAXVAL:	モジュロ回転軸に対して、モジュロ位置を特定します。これは、パラメータで設定したモジュロ範囲を基準とするモジュロ余りに対応します(初期設定では、0 ～ 360°です。モジュロ範囲の開始位置とサイズは、MD30340 MODULO_RANGE_START と MD30330 $MA_MODULO_RANGE で変更できます)。

注記

SPI の拡張

軸機能 SPI(n)を使用すると、フレーム成分の読み取りと書き込みも実行できます。つまり、$P_PFRAME[SPI(1),TR]=2.22 構文などでフレームを記述できます。

さらにアドレス AX[SPI(1)]=<軸位置>で軸の位置決めをプログラム指令して、軸を移動できます。必要条件は、主軸が位置決めモードまたは軸モードであることです。

例

例 1:AXNAME、AX、ISAXIS

プログラムコード	コメント
OVRA[AXNAME("径方向軸")]=10	; 径方向軸のオーバライド
AX[AXNAME("径方向軸")]=50.2	; 径方向軸の終了位置
OVRA[SPI(1)]=70	; 主軸 1 のオーバライド
AX[SPI(1)]=180	; 主軸 1 の終了位置
IF ISAXIS(1) == FALSE GOTOF CONTINUE	; 横軸が使用可能?
AX[$P_AXN1]=100	; 横軸を移動します
CONTINUE:	

例 2 :AXSTRING

AXSTRING[SPI(n)]でプログラム指令すると、主軸に割り当てた軸の軸インデックスが主軸番号として出力されなくなり、代わりに文字列「Sn」が出力されます。

プログラムコード	コメント
AXSTRING[SPI(2)]	; 文字列「S2」が出力されます。

例 3 :MODAXVAL

モジュロ回転軸 A のモジュロ位置を特定します。

軸位置 372.55 は、計算の開始値です。

パラメータ設定したモジュロ範囲は次のように 0 ～ 360°です。

MD30340 MODULO_RANGE_START = 0

MD30330 $MA_MODULO_RANGE = 360

プログラムコード	コメント
R10=MODAXVAL(A,372.55)	; 算出したモジュロ位置は R10 = 12.55 です。

例 4 :MODAXVAL

プログラム指令した軸識別子がモジュロ回転軸を表わしていない場合は、変換される値(<軸位置>)を、変換せずに返します。

プログラムコード	コメント
R11=MODAXVAL(X,372.55)	; X は直線軸です; R11 = 372.55

18.2 置換可能なジオメトリ軸(GEOAX)

「切り替え可能なジオメトリ軸」機能により、マシンデータによって設定されたジオメトリ軸を他のチャネル軸に置き換えることができます。

構文

```
GEOAX(<n>,<チャネル軸>,<n>,<チャネル軸>,<n>,<チャネル軸>)
GEOAX()
```

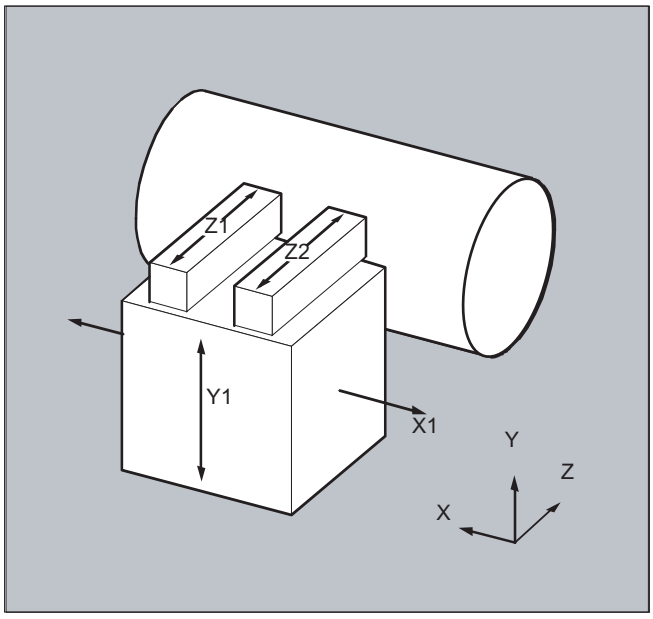
意味

GEOAX(...)	切り替え可能なジオメトリ軸の機能 **注**: パラメータ指定のない GEOAX() では、マシンデータでパラメータ設定されたジオメトリ軸の基本設定が再度有効になります。
<n>	指定されたチャネル軸と置換されるジオメトリ軸の番号 値の範囲: 0、1、2、3 **注**: 0:指定されたチャネル軸は、置換されずにジオメトリ軸グループから削除されます。 1:1 番目のジオメトリ軸 ≙ WCS の座標軸 X (横軸) 2:2 番目のジオメトリ軸 ≙ WCS の座標軸 Y (縦軸) 3:3 番目のジオメトリ軸 ≙ WCS の座標軸 Z (垂直軸)
<チャネル軸>	ジオメトリ軸グループに追加されるチャネル軸の名称

例

例 1:交互にジオメトリ軸となる 2 つの軸の切り替え

チャネル軸 X1、Y1、Z1、Z2 を使用して、次のように工具スライドを移動できます。



電源投入後にまず、3 番目のジオメトリ軸として Z1 が、「Z」というジオメトリ軸名称で有効になり、X1 と Y1 とともにジオメトリ軸グループを構成するように、ジオメトリ軸を設定します。

これで軸 Z1 と Z2 は、パートプログラムでジオメトリ軸 Z として交互に使用されるようになります。

プログラムコード	コメント
...	
N100 GEOAX(3,Z2)	; チャネル軸 Z2 が 3 番目のジオメトリ軸(Z)として機能します。
N110 G1 ...	
N120 GEOAX(3,Z1)	; チャネル軸 Z1 が 3 番目のジオメトリ軸(Z)として機能します。
...	

例 2:チャネル軸が 6 つの場合のジオメトリ軸の変更

機械には、名称が XX、YY、ZZ、U、V、W の 6 つのチャネル軸があります。

マシンデータによるジオメトリ軸構成の基本設定は、次のとおりです。

チャネル軸 XX =1 番目のジオメトリ軸(X 軸)

チャネル軸 YY =2 番目のジオメトリ軸(Y 軸)

チャネル軸 ZZ =3 番目のジオメトリ軸(Z 軸)

プログラムコード	コメント
N10 GEOAX()	; ジオメトリ軸の基本構成が有効です。
N20 G0 X0 Y0 Z0 U0 V0 W0	; すべての軸を位置 0 に早送りします。

プログラムコード	コメント
N30 GEOAX(1,U,2,V,3,W)	; チャネル軸 U が 1 番目の(X)、V が 2 番目の(Y)
	; W が 3 番目のジオメトリ軸(Z)になります。
N40 GEOAX(1,XX,3,ZZ)	; チャネル軸 XX が 1 番目の(X)、ZZ が 3 番目の
	; ジオメトリ軸(Z)になります。チャネル軸 V は 2 番目の
	; ジオメトリ軸(Y)のままです。
N50 G17 G2 X20 I10 F1000	; X/Y 平面上の一周円です。チャネル軸
	; XX と V が移動します。
N60 GEOAX(2,W)	; チャネル軸 W が 2 番目のジオメトリ軸(Y)になります。
N80 G17 G2 X20 I10 F1000	; X/Y 平面上の一周円です。チャネル軸
	; XX と W が移動します。
N90 GEOAX()	; 初期状態にリセットします。
N100 GEOAX(1,U,2,V,3,W)	; チャネル軸 U が 1 番目の(X)、V が 2 番目の
	; (Y)、W が 3 番目のジオメトリ軸(Z)になります。
N110 G1 X10 Y10 Z10 XX=25	; チャネル軸 U、V、W をそれぞれ
	; 位置 10 へ移動します。XX が付加軸として位置 25 に移動します。
N120 GEOAX(0,V)	; V をジオメトリ軸グループから解除します。
	; U は 1 番目のジオメトリ軸(X)、
	; W は 3 番目のジオメトリ軸(Z)のままです。
	; 2 番目のジオメトリ軸(Y)にはそのまま、何も割り当てられません。
N130 GEOAX(1,U,2,V,3,W)	; チャネル軸 U は 1 番目の(X)のまま、V は
	; 2 番目の(Y)になり、W は 3 番目のジオメトリ軸(Z)のままです。
N140 GEOAX(3,V)	; V は 3 番目のジオメトリ軸(Z)になります。
	; ここで、W は上書きされるため、ジオメトリ軸グループ
	; から解除されます。2 番目のジオメトリ軸(Y)は
	; 何も割り当てられないままです。

マシンデータ

軸構成

ジオメトリ軸、付加軸、機械軸のチャネル軸への割り当て:

- 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

リセット応答

変更されたジオメトリ軸割り当てのリセット応答:

- MD20110 $MC_RESET_MODE_MASK、ビット 12
- MD20118 $MC_GEOAX_CHANGE_RESET

NC スタートの動作

- MD20112 $MC_START_MODE_MASK、ビット 12

PLC ユーザープログラムへの通知

ジオメトリ軸が変更されたときに PLC インタフェースで出力される M 命令のパラメータ設定オプション。

- MD22532 $MC_GEOAX_CHANGE_M_CODE

必要条件

ジオメトリ軸の切り替えなし

- 次のいずれかの機能が動作中の場合、ジオメトリ軸の切り替えは実行できません。
  - 座標変換
  - スプライン補間
  - 工具径補正
  - 工具仕上げオフセット
- ジオメトリ軸ともう 1 つのチャネル軸が同じ名称の場合。
- ジオメトリ軸の切り替えに参加する軸のいずれかが、ブロック全体での位置決め軸や従動軸の軸連結などのブロックの境界を越えて実行される操作に関与する場合。

回転軸

回転軸はジオメトリ軸としてプログラムできません。

置換後の軸の状態

ジオメトリ軸グループ内で切り替えで置換した軸は、そのチャネル軸名称による切り替え操作後に、追加軸としてプログラム指令できます。

フレーム、プロテクションゾーン、作業領域リミット

すべてのフレーム、プロテクションゾーン、および作業領域リミットは、ジオメトリ軸の切り替え後に解除されます。

極座標

GEOAX でジオメトリ軸を置換すると、G17 ～ G19 による平面変更と同様に、モーダル極座標を 0 の値に設定します。

DRF、WO

ハンドルオフセット(DRF)または外部ゼロオフセット(WO)が使用可能で、切り替え後もそのまま有効です。

ジオメトリ軸の基本構成

GEOAX() 命令は、ジオメトリ軸グループの基本構成を呼び出します。

システムは、POWER ON 後、および「レファレンス点復帰」モードへの切り替え時に、自動的に基本構成に戻ります。

工具長補正

動作中の工具長補正は切り替え操作後も有効です。ただし、新しく追加した、または位置を置換したジオメトリ軸の場合は、移動が完了していないと見なされます。これらのジオメトリ軸の最初の動作命令では、それに対応して、得られる移動距離が、工具長補正とプログラム指令した移動距離の和になります。

置換操作後も軸グループの位置を保持するジオメトリ軸は、工具長補正に対する状態も保持します。

動作中の座標変換のジオメトリ軸構成

- 動作中の座標変換に対して座標変換マシンデータによってパラメータ設定されるジオメトリ軸構成は、「切り替え可能なジオメトリ軸」機能を使用して変更できません。
- 座標変換に対する異なるジオメトリ軸構成には、座標変換マシンデータで異なるデータセットをパラメータ設定してください。
- GEOAX で変更したジオメトリ軸構成は、座標変換を起動すると解除されます。
- ジオメトリ軸に関しては、動作中の座標変換の座標変換固有のジオメトリ軸パラメータ設定が、ジオメトリ軸の切り替えに関連するパラメータ設定よりも優先されます。
  例:座標変換が動作中です。マシンデータに従って、座標変換はチャネルリセットで保持されます。同時に、ジオメトリ軸の基本設定がチャネルリセットで復元されます。
  座標変換に対して指定されたジオメトリ軸構成が保持されます。
- 座標変換をオフにすると、パラメータ設定されたジオメトリ軸構成の基本設定が再度有効になります。

JOG モード、REF 機械機能

JOG モード、REF 機械機能(レファレンス点復帰)に切り替えると、マシンデータでパラメータ設定されたジオメトリ軸構成が有効になります。

18.3 軸コンテナ(AXCTSWE、AXCTSWED、AXCTSWEC)

「AXCTSWED」命令または「AXCTSWE」命令を使用して、指定した軸コンテナの回転を有効にします。

「AXCTSWEC」命令により、以前に設定された軸コンテナ回転のイネーブルはキャンセルされます。

構文

```
AXCTSWE(<ID>)
AXCTSWED(<ID>)
AXCTSWEC(<ID>)
```

意味

AXCTSWE:	軸コンテナの回転の有効化 プログラムの処理は「AXCTSWE」で停止しません。 回転は、軸コンテナ関連のすべてのチャネルが有効になると直ちに実行されます。
AXCTSWED:	軸コンテナ関連の他のチャネルを考慮せずに軸コンテナの回転を有効化 **注** ● パートプログラムまたはシンクロナイズドアクションのセットアップを簡素化する命令タイプ ● 軸コンテナ関連の他のチャネルに関する動作は、以下で指定できます。MD12760 $MN_ AXCT_FUNCTION_MASK、ビット 0
AXCTSWEC:	軸コンテナの回転の有効化の取り消し **注** 回転がまだ起動していない場合にのみ、軸コンテナの回転のイネーブルはキャンセルできます。 $AN_AXCTSWA[<軸コンテナ>] == 0 システム変数については、「軸コンテナ(AXCTSWE、AXCTSWED、AXCTSWEC) (ページ 769)」を参照してください。

<ID>:	軸コンテナまたはコンテナ軸の識別子	
	CT<番号>:	軸コンテナの初期設定の識別子: MD12750 $MN_AXCT_NAME_TAB 例:「CT1」
	<コンテナ>:	軸コンテナのユーザー用識別子 MD12750 $MN_AXCT_NAME_TAB 例:「CONTAINER_1」
	<軸>:	チャネル内の既知のコンテナ軸の識別子

注記

移動量

軸コンテナ回転の移動量は、次のセッティングデータで設定します。

SD41700 $SN_AXCT_SWWIDTH

詳細情報

診断

軸コンテナの現在の状態は、次のシステム変数で読み取ることができます。

システム変数	タイプ	説明
$AC_AXCTSWA[<名称>]	BOOL	軸コンテナのチャネル別の状態です。
$AN_AXCTSWA[<軸コンテナ>]	BOOL	軸コンテナの NCU 別の状態です。
$AN_AXCTSWE[<軸コンテナ>]	INT	軸コンテナの回転のスロット別の状態です。 システム変数で、軸コンテナスロット**ビット単位**の状態が提供されます。各ビットはスロットに対応しています。
$AN_AXCTAS[<軸コンテナ>]	INT	軸コンテナの切り替えに使用されるロケーション(スロット)の数

自動 GET/GETD による軸コンテナの回転

次のマシンデータを使用して、「AXCTSWE」命令での自動「GET/GETD」により、チャネルのすべてのコンテナ軸がチャネルに入るように設定できます。軸入れ替えは、コンテナ回転後にのみ再度可能になります。

MD10722 $MN_AXCHANGE_MASK、ビット 1 = 1

注記

自動「GET/GETD」による軸コンテナ回転は、「メイン軸」状態の軸(PLC 軸など)では実行**されません**。これは、軸コンテナ回転のためには軸がその状態を終了する必要があるためです。

18.4 有効な軸位置を待機します(WAITENC)

言語命令「WAITENC」を使用して、NC プログラムは、同期または同期解除された軸位置が、MD34800 $MA_WAIT_ENC_VALID = 1 で設定された軸で使用可能になるまで待機します。

ASUB を開始したり、運転を JOG モードに変更したりすると、待機状態が中断することがあります。プログラムが続行されると、該当する場合は、再び待機状態になります。

注記

操作画面では、保留状態の「検出を待機」を使用して、待機状態が表示されます。

構文

「WAITENC」は、任意の NC プログラムのプログラム区間でプログラム指令できます。

専用ブロックでプログラム指令してください。

```
...
WAITENC
...
```

例

「WAITENC」は、以下の用途例で示すように、イベント起動のユーザープログラム.../_N_CMA_DIR/_N_PROG_EVENT_SPF などで使用します。

用途例 :方向座標変換による電源切断後の工具の引き戻し

工具オリエンテーションによる加工が、停電により中断されました。
停電から復旧すると、イベント起動のユーザープログラム.../_N_CMA_DIR/_N_PROG_EVENT_SPF が呼び出されます。

イベント起動のユーザープログラムでは、システムが、「WAITENC」によって同期または同期解除された位置決め軸を待機し、その後に、ワークを工具方向に配置するフレームを計算できるようになります。

プログラムコード	コメント
...	
IF $P_PROG_EVENT == 4	; 起動しています。
IF $P_TRAFO <> 0	; 座標変換が選択されています。
WAITENC	; 旋回軸の有効な軸位置を待機します。

プログラムコード	コメント
TOROTZ	; WCS の Z 軸を工具軸の方向に回転します。
ENDIF	
M17	
ENDIF	
...	

その後、JOG モードで、工具軸の方へ後退移動によって工具を後退させることができます。

18.5 プログラム指令可能なパラメータセット切り替え(SCPARA)

SCPARA 命令により、軸の特定のパラメータセットへの切り替えを要求できます。

注記

ねじ切り時のパラメータセット切り替えなし

ねじ切り G33 およびタッピング G331/G332 時には、パラメータセットは制御装置によって選択されるため、切り替えできません。

パラメータセット切り替えの無効化

パラメータセット切り替えは、NC/PLC インタフェース経由で要求することもできます。切り替えの競合を避けるために、次のように NC/PLC インタフェース経由で NC (SCPARA)のパラメータセット切り替えを無効にすることができます。

DB31, ... DBX9.3 (NC によって無効にするパラメータセットの指定)

注記

NC/PLC インタフェース経由でパラメータセット切り替えが無効にされているときに SCPARA がパラメータセット切り替えを要求すると、切り替えはエラーメッセージを表示せずに拒否されます。

構文

```
SCPARA[<軸>]=<値>
```

意味

SCPARA:	命令:パラメータセットの切り替え	
<軸>:	軸識別子(チャネル軸)	
	タイプ:	AXIS
<値>:	パラメータセット番号:1、2、3、... 最大パラメータセット番号	

例

プログラムコード	コメント
...	
N110 SCPARA[X]= 3	; 選択:X 軸、3 番目のパラメータセット
...	

詳細情報

パラメータセット切り替えの有効化

軸のパラメータセット切り替えは明示的に有効にしてください。

MD35590 $MA_PARAMSET_CHANGE_ENABLE[<軸>]

パラメータセット番号の読み取り

選択したパラメータセット(指定したパラメータセット)の番号は、システム変数 $AA_SCPAR を介して読み取ることができます。

参照先

パラメータセットの詳細情報は、以下を参照してください。

機能マニュアル 基本機能、「速度、指令/フィードバック回路、閉ループ制御(G2)」>「閉ループ制御」>「位置制御のパラメータセット」の章

18.6 現在の NC 言語の適用範囲の確認(STRINGIS)

機能「STRINGIS(...)」を使用すると、指定した文字列が、実際の言語の適用範囲で NC プログラミング言語の要素として使用可能かどうかを確認できます。

定義

```
INT STRINGIS(STRING <名称>)
```

構文

```
STRINGIS(<名称>)
```

意味

STRINGIS	戻り値を伴う機能
<名称>:	確認する NC プログラミング言語要素の名称
戻り値:	戻り値の書式は yxx (10 進数)です。

NC プログラミング言語の要素

確認できる NC プログラミング言語の要素は以下のとおりです。

- 既存のすべての G 機能グループの G 命令、例:G0、INVCW、POLY、ROT、KONT、SOFT、CUT2D、CDON、RMBBL、SPATH
- DIN または NC のアドレス、ADIS、RNDM、SPN、SR、MEAS など
- 関数、例:TANG(...)または GETMDACT
- 処理、例:SBLOF
- キーワード、例:ACN、DEFINE、SETMS
- システムデータ(マシンデータ$M...、セッティングデータ$S...、オプションデータ$O... など)
- システム変数$A...、$V...、$P...
- 算術変数 R...
- 有効なサイクルのサイクル名称
- GUD 変数と LUD 変数

- マクロ名称
- ラベル名称

戻り値

戻り値の先頭の桁のみが有効です。戻り値の書式は yxx で、y =基本情報、および xx =詳細情報です。

戻り値	意味	
000	'名称'文字列は、このシステムでは不明な文字列です [1]	
100	'名称'文字列は、NC プログラミング言語の要素ですが、現在はプログラミングできません(オプション/機能が無効です)	
2xx	'名称'文字列は、NC プログラミング言語のプログラム指令可能な要素です(オプション/機能が有効です)。詳細情報 xx には、以下の要素タイプに関する詳細情報が含まれます。	
	xx	**意味**
	01	DIN アドレスまたは NC アドレス [2]
	02	G 命令(G04、INVCW など)
	03	戻り値を伴う機能
	04	戻り値を伴わない機能
	05	キーワード、例: DEFINE
	06	マシンデータ($M...)、セッティングデータ($S...)、またはオプションデータ($O...)
	07	システム変数($...)や算術変数(R...)などのシステムパラメータ
	08	サイクル(サイクルを NC にロードし、サイクルプログラムを起動してください [3])
	09	GUD 変数(GUD 変数を GUD 定義ファイルで定義して、GUD 変数を有効にしてください)
	10	マクロ名称(マクロをマクロ定義ファイルで定義して、マクロを起動してください) [4]
	11	現在のパートプログラムの LUD 変数
	12	ISO G 命令(ISO 言語モードを有効にしてください)
400	'名称'文字列は NC アドレスです。ただし、その名称は xx == 01 でも xx == 10 でもなく、また、G でも R でもありません [2]	

戻り値	意味
y00	何も割り当てることはできません

1)コントローラによっては、特定の状況の場合に、例えば SINUMERIK 802D sl などの場合、当社の NC 言語命令のサブセットのみが認識されます。このようなコントローラの場合は、主要な当社 NC 言語命令に対して、値 0 が返されます。この動作は、MD10711 $MN_NC_LANGUAGE_CONFIGURATION を使用して変更できます。MD10711 = 1 の場合は、当社 NC 言語命令に対して、常に値 100 が返されます。

2) NC アドレスに使用できる文字は、A、B、C、E、I、J、K、Q、U、V、W、X、Y、Z です。これらのNC アドレスは、アドレス拡張子を付けてプログラミングすることもできます。アドレス拡張子は、STRINGIS で確認するときに指定できます。例:201 == STRINGIS("A1").
文字 D、F、H、L、M、N、O、P、S、、T は NC アドレス、またはユーザー定義の補助機能です。これらに対しては、常に値 400 が返されます。例:400 == STRINGIS("D").STRINGIS で確認するときは、これらの NC アドレスにアドレス拡張子を付けて指定することはできません。
例:000 == STRINGIS("M02")、ただし、400 == STRINGIS("M").

3)サイクルパラメータの名称は、STRINGIS では確認できません。

4)マクロとして定義されたアドレス。G、H、M、L はマクロと見なされます。

例

以下の例では、文字列として指定した NC 言語要素(特に注記がない限り)を、原則としてコントローラにプログラミングできます。

1. 文字列「T」を補助機能として定義する場合:
   ```
   400 == STRINGIS("T")
   000 == STRINGIS("T3")
   ```
2. 文字列「X」を軸として定義する場合:
   ```
   201 == STRINGIS("X")
   201 == STRINGIS("X1")
   ```
3. 文字列「A2」を拡張子付きアドレスとして定義する場合:
   ```
   201 == STRINGIS("A")
   201 == STRINGIS("A2")
   ```
4. 文字列「INVCW」を名称付き G 命令として定義する場合:
   ```
   202 == STRINGIS("INVCW")
   ```
5. 文字列「$MC_GCODE_RESET_VALUES」をマシンデータとして定義する場合:
   ```
   206 == STRINGIS("$MC_GCODE_RESET_VALUES")
   ```
6. 文字列「GETMDACT」が NC 言語機能である場合:
   ```
   203 == STRINGIS("GETMDACT")
   ```
7. 文字列「DEFINE」がキーワードの場合:
   ```
   205 == STRINGIS("DEFINE")
   ```
8. 文字列「$TC_DP3」がシステムパラメータ(工具長成分)である場合:
   ```
   207 == STRINGIS("$TC_DP3")
   ```
9. 文字列「$TC_TP4」がシステムパラメータ(工具サイズ)である場合:
   ```
   207 == STRINGIS("$TC_TP4")
   ```

10. 文字列「$TC_MPP4」がシステムパラメータ(マガジンロケーションの状態)である場合:
   – 工具マガジン管理が動作中の場合:`207 == STRINGIS("$TC_MPP4") ;`
   – 工具マガジン管理が動作中でない場合:`000 == STRINGIS("$TC_MPP4")`
   また、下記の「工具マガジン管理」の節も参照してください。
11. 文字列「MACHINERY_NAME」を GUD 変数として定義する場合:
   `209 == STRINGIS("MACHINERY_NAME")`
12. 文字列「LONGMACRO」をマクロとして定義する場合:
   `210 == STRINGIS("LONGMACRO")`
13. 文字列「MYVAR」を LUD 変数として定義する場合:
   `211 == STRINGIS("MYVAR")`
14. 文字列「XYZ」が、NC では不明な命令、GUD 変数、マクロ名称、またはサイクル名称である場合:
   `000 == STRINGIS("XYZ")`

工具マガジン管理

工具マガジン管理機能が動作中でない場合は、以下のマシンデータとは無関係に、工具マガジン管理のシステムパラメータが STRINGIS で指定されます。

- MD10711 $MN_NC_LANGUAGE_CONFIGURATION

値は常に 000 となります。

ISO モード

「ISO モード」機能が動作中の場合は、次のようになります。

- MD18800 $MN_MM_EXTERN_LANGUAGE (適用、外部 NC 言語)
- MD10880 $MN_ MM_EXTERN_CNC_SYSTEM (設定するコントロールシステム)

STRINGIS は、指定した文字列を最初に SINUMERIK G 命令として確認します。この文字列が SINUMERIK G 命令でない場合は、次に、ISO G 命令として確認します。

プログラム指令切り換え(`G290` (SINUMERIK モード)、`G291` (ISO モード))は、STRINGIS には影響しません。

例

機能 STRINGIS(...)に有効なマシンデータの値は、次のとおりです。

- MD10711 $MN_NC_LANGUAGE_CONFIGURATION = 2 (オプションを設定した NC 言語命令のみを、認識します)
- MD19410 $ON_TRAFO_TYPE_MASK = 'H0' (オプション:座標変換)
- MD10700 $MN_PREPROCESSING_LEVEL='H43' (サイクルの先読みが有効)

次のサンプルプログラムが、エラーメッセージを表示せずに実行されます。

プログラムコード	コメント
N1 R1=STRINGIS("TRACYL")	; R1 == 0、座標変換機能がないため、TRACYL が「不明」と見なされるためです。オプション
N2 IF STRINGIS("TRACYL") ==204	
N3 TRACYL(1,2,3)	; N3 をスキップします
N4 ELSE	
N5 G00	; また、代わりに N5 を実行します
N6 ENDIF	
N7 M30	

18.7 対話形式によるパートプログラムからのウィンドウの呼び出し(MMC)

ユーザー用ダイアログは、予約サブプログラム MMC(...)を使用して、SINUMERIK Operate の操作画面の NC プログラムなどから表示できます。

次のタイプのダイアログを設定できます。

- Run MyScreens
- Easy XML
- User XML

参照先

- プログラミングマニュアル Run MyScreens
- プログラミングマニュアル Easy XML

構文

```
MMC("<ADDRESS>,<COMMAND>,<FILE>,<DIALOG>","<QUIT>")
```

意味

MMC(...):	サブプログラム識別子 パラメータは位置コードを使用して、命令文字列と応答文字列という 2 つのコンマ区切り文字列で指定されます。	
コマンド内のパラメータ:		
<ADDRESS>:	設定したユーザー対話ボックスを実行する操作エリア	
	機能	**操作エリア**
	「Run MyScreens」ユーザーダイアログ	CYCLES
	「Easy XML」ユーザーダイアログ	CYCLES
	User XML	XML
	ポップアップウィンドウ「Run MyScreens」	POPUPDLG
	ポップアップウィンドウ「Easy XML」	POPUPDLG

<COMMAND>:	実行する命令	
	機能	**命令**
	「Run MyScreens」ユーザーダイアログ	PICTURE_ON、PICTURE_OFF
	「Easy XML」ユーザーダイアログ	PICTURE_ON、PICTURE_OFF
	User XML	XML_ON、XML_OFF
	ポップアップウィンドウ「Run MyScreens」	PICTURE_ON、PICTURE_OFF
	ポップアップウィンドウ「Easy XML」	PICTURE_ON、PICTURE_OFF
<FILE>:	表示するダイアログをプログラムするファイルの名前	
	機能	**ファイル**
	「Run MyScreens」ユーザーダイアログ	<name>.com
	「Easy XML」ユーザーダイアログ	<name>.xml
	User XML	<name>.xml
	ポップアップウィンドウ「Run MyScreens」	<name>.com
	ポップアップウィンドウ「Easy XML」	<name>.xml
	NC プログラム内に直接設定するポップアップウィンドウ「Easy XML」(例 2 を参照)	xmldial_emb.xml
<DIALOG>:	表示する対話画面の名前	
	機能	**ダイアログの名前**
	NC プログラム内に直接設定するポップアップウィンドウ「Easy XML」を除くすべての機能	<FILE>ファイルで設定するダイアログの名前
	NC プログラム内に直接設定するポップアップウィンドウ「Easy XML」(例 3 を参照)	main
応答文字列内のパラメータ:		

<QUIT>:	確認タイプ	
	N:	応答なし 命令の送信が終わると、プログラムの実行を続行します。命令を正常に実行できない場合でも、フィードバックはありません。 **注** NC プログラム内に表示時間(ドウェル時間)がプログラムされている場合、応答タイプ「N」を使用してください(後述の例 2 を参照)
	A:	非同期応答 命令の発行後にプログラムの実行を続行します。戻り値はユーザー固有の応答変数(GUD 変数)に保存されます。これは、ダイアログ設定の範囲内で定義され、NC プログラム内に読み込むことができます。

例

例 1

NC プログラム内でのダイアログの表示とユーザー操作への応答。

```
; 応答変数 QUIT は、グローバルユーザー変数(GUD)として作成済みです
; ダイアログが設定されたときのタイプ STRING:
; DEF NCK STRING[20] QUIT
QUIT = "XXX"                                    ; 応答変数の初期化
G4 F5
MMC("CYCLES,PICTURE_ON,test.com,test1","A       ; ダイアログの表示
")                                              ;  - 操作エリア:CYCLES
                                                ;  - 画像ステータス:PICTURE_ON (表示)
                                                ;  - ダイアログ画面ファイル:test.com
                                                ;  - ダイアログ画面:test1
INPUT:                                          ; ユーザー入力を待機
   Stopre                                       ;    先読み停止
   IF MATCH (QUIT,"RUN") >= 0 GOTOF WORK        ;    ソフトキー「RUN」
   IF MATCH (QUIT,"CHK") >= 0 GOTOF CHECK       ;    ソフトキー「CHK」
GOTOB INPUT                                     ; => 待機

WORK:                                           ; ソフトキー「RUN」が押された
MSG("Continue with processing -> NC             ; 出力メッセージ
start")
```

プログラムコード	コメント
MMC("CYCLES,PICTURE_OFF","N")	; ダイアログを閉じる
M0	; NC スタートを待機
GOTOF END	; => プログラム終了
CHECK:	; ソフトキー「CHK」が押された
MSG("Approach position -> NC start")	; 出力メッセージ
MMC("CYCLES,PICTURE_OFF","N")	; ダイアログを閉じる
M0	; NC スタートを待機
GOTOF END	; => プログラム終了
END:	
...	

例 2

ダイアログの表示時間は、NC プログラムでドウェル時間などを使用して定義されます。

プログラムコード	コメント
F1000 G94	
...	
MMC("POPUPDLG,PICTURE_**ON**,xmldial_emb.xml,main","N")	; ダイアログの表示
X200	
Z40	
MMC("POPUPDLG,PICTURE_**OFF**","N")	; ダイアログを閉じる

例 3

NC プログラムへのポップアップスクリプトの埋め込みとその使用法。

プログラムコード

```
PROC POPUP_TEST
; ----------------------------- スクリプト -----------------------------
; <main_dialog entry="rpara_main">
;    <let name="xpos" />
;    <let name="ypos" />
;    <let name="field_name" type="string" />
;    <let name="num" />
;    <menu name="rpara_main">
;       <open_form name="rpara_form"/>
;       <softkey_back>
;          <close_form />
;       </softkey_back>
;    </menu>
;    <form name="rpara_form">
```

プログラムコード

```
;       <init>
;          <caption>NC パートプログラムからのマスク</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>
; ========================= プログラムセクション ============================
...
```

プログラムコード

```
G94 F100
MMC("POPUPDLG,PICTURE_ON,xmldial_emb.xml,main","N")
G4 F4
X200
MMC("POPUPDLG,PICTURE_OFF","N")
G4 F2
X0
...
```

必要条件

- ダイアログの定義ファイル*.com は、「proj」フォルダに保存してください。
- ダイアログの Easy XML 定義ファイル*.xml は、「appl」フォルダに保存してください。
  定義ファイルが別のディレクトリに保存されている場合、パスを「appl」ディレクトリから始まる相対パスで指定してください。
- ユーザー定義ダイアログを異なるチャネルから同時に表示することはできません。

18.8 プログラム実行時間/ワークカウンタ

工作機械のオペレータをサポートするために、プログラム実行時間およびワークカウンタに関する情報が提供されます。

この情報は、NC プログラムや PLC プログラムでシステム変数として処理できます。 この情報は、操作画面に表示することもできます。

18.8.1 プログラム実行時間

「プログラム実行時間」機能には、技術的な処理を監視する内部 NC タイマが装備されています。この技術的な処理は、NC 別とチャネル別システム変数を使用すれば、パートプログラムとシンクロナイズドアクション内に読み込むことができます。

実行時間計測の起動($AC_PROG_NET_TIME_TRIGGER)は、この機能で唯一の書き込み可能なシステム変数で、これを使用して、プログラム区間を選択して計測します。つまり、NC プログラムに$AC_PROG_NET_TIME_TRIGGER を書き込んで、時間計測を再度有効化および無効化できます。

システム変数	意味	動作
NC 別		
$AN_SETUP_TIME	初期値による、コントローラの最後の電源投入(「コールド再始動」)後の経過時間(分単位)です。 初期値により、コントローラの電源投入のたびに、自動的に「0」にリセットされます。	• 常に有効です
$AN_POWERON_TIME	コントローラの最後の通常の電源投入(「ウォーム再始動」)後の経過時間(分単位)です。 コントローラの通常の電源投入のたびに、自動的に「0」にリセットされます。	
チャネル別		

システム変数	意味	動作
$AC_OPERATING_TIME	自動モードの NC プログラムの合計実行時間(秒単位)です。 コントローラの電源投入のたびに、自動的に値が「0」にリセットされます。	• MD27860 により有効になります。 • 自動モードの場合のみ有効です。
$AC_CYCLE_TIME	選択した NC プログラムの実行時間(秒単位)です。 新しい NC プログラムを起動するたびに、自動的に値が「0」にリセットされます。	
$AC_CUTTING_TIME	処理時間(秒単位)です。 (1 軸以上が動作中)の軌跡軸の実行時間は、早送りが動作していないすべての NC プログラムで、NC スタートとエンドオブプログラム、または NC リセットの間に計測します。計測は、ドウェル時間が有効なときは中断されます。 初期値により、コントローラの電源投入のたびに、自動的に値が「0」にリセットされます。	
$AC_ACT_PROG_NET_TIME	現在の NC プログラムの実際の正味実行時間(秒単位)です。 新しい NC プログラムが起動すると、自動的に「0」にリセットされます。	• 常に有効です。 • 自動モードの場合のみ有効です。
$AC_OLD_PROG_NET_TIME	M30 で正常に終了した直後のプログラムの正味実行時間(秒単位)です。	
$AC_OLD_PROG_NET_TIME_COUNT	$AC_OLD_PROG_NET_TIME の変化 POWER ON 後の $AC_OLD_PROG_NET_TIME_COUNT は「0」です。 コントローラが$AC_OLD_PROG_NET_TIME に新しく書き込んだ場合は常に、$AC_OLD_PROG_NET_TIME_COUNT が増加します。	

システム変数	意味		動作
$AC_PROG_NET_TIME_TRIGGER	実行時間計測の起動:		● 自動モードの場合のみ有効です。
	0	中立状態 起動は無効です。	
	1	終了 測定を停止し、$AC_ACT_PROG_NET_TIME から $AC_OLD_PROG_NET_TIME に値をコピーします。$AC_ACT_PROG_NET_TIME は「0」に設定された後、引き続き実行されます。	
	2	開始 測定を開始し、その際に $AC_ACT_PROG_NET_TIME が「0」に設定されます。 $AC_OLD_PROG_NET_TIME は変更されません。	
	3	停止 計測を停止します。 $AC_OLD_PROG_NET_TIME を変更せず、$AC_ACT_PROG_NET_TIME を、再開するまで一定に保ちます。	
	4	再開 測定が再開されます。つまり、前に停止した測定が継続されます。 $AC_ACT_PROG_NET_TIME は続行されます。$AC_OLD_PROG_NET_TIME は変更されません。	
POWER ON の結果、すべてのシステム変数が「0」にリセットされます。			

注記

工作機械メーカ

マシンデータ MD27860 $MC_PROCESSTIMER_MODE を使用して、使用可能なタイマーを起動します。

一部の機能(GOTOS、オーバライド= 0%、動作中の試運転送り速度、プログラムテスト、ASUB、PROG_EVENT、…など)の有効時間の計測の特性は、マシンデータ MD27850 $MC_PROG_NET_TIMER_MODE と
MD27860 $MC_PROCESSTIMER_MODE を使用して設定します。

参照先:
総合機能説明書 基本機能、モードグループ、チャネル、プログラム運転,リセット応答(K1): プログラム実行時間の章

注記

ワーク加工の残り時間

次のタイマー値を使用して同じワークを連続で加工する場合、ワークの加工残り時間が特定できます。

- 最後に生産したワークの加工処理時間です($AC_OLD_PROG_NET_TIME を参照してください)。
- 現在の処理時間です($AC_ACT_PROG_NET_TIME を参照してください)。

操作画面には、現在の処理時間の他に、残り時間も表示されます。

注記

STOPRE の使用

システム変数$AC_OLD_PROG_NET_TIME および
$AC_OLD_PROG_NET_TIME_COUNT は、先読み停止処理を自動的には実行しません。これは、パートプログラムで使用する時に、前回のプログラム実行時のシステム変数の値を使用する場合は問題はありません。ただし、実行時間計測のトリガ ($AC_PROG_NET_TIME_TRIGGER)を書き込む頻度が非常に高く、その結果、$AC_OLD_PROG_NET_TIME が頻繁に変更される場合は、パートプログラムで STOPRE を明示的に使用してください。

必要条件

- **ブロックサーチ**
  プログラム実行時間は、ブロックサーチでは特定されません。
- **REPOS**
  REPOS の処理時間を、現在の処理時間に加算します($AC_ACT_PROG_NET_TIME)。

例

例 1:「mySubProgrammA」の時間測定

プログラムコード

```
...
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
```

プログラムが N80 行を処理したら、「mySubProgrammA」の正味実行時間が $AC_OLD_PROG_NET_TIME 内に記録されます。

$AC_OLD_PROG_NET_TIME の値は、次のようになります。

- M30 の後も保持されます。
- ループを通過するたびに更新されます。

例 2 :「mySubProgrammA」と「mySubProgrammC」の時間測定

プログラムコード

```
...
N10 DO $AC_PROG_NET_TIME_TRIGGER=2
N20 mySubProgrammA
N30 DO $AC_PROG_NET_TIME_TRIGGER=3
N40 mySubProgrammB
N50 DO $AC_PROG_NET_TIME_TRIGGER=4
N60 mySubProgrammC
N70 DO $AC_PROG_NET_TIME_TRIGGER=1
N80 mySubProgrammD
N90 M30
```

18.8.2 ワークカウンタ

「ワークカウンタ」機能により、制御装置内部でさまざまなカウンタを特別に使用して、ワークを数えることができます。

このカウンタは、0 ～ 999 999 999 の値の範囲で読み取りと書き込みのアクセスができる、チャネル別システム変数です。

システム変数	意味
$AC_REQUIRED_PARTS	生産ワーク数(ワーク数の指令値) このカウンタで定義したワーク数に達すると、実ワークカウント数($AC_ACTUAL_PARTS)を「0」にリセットします。
$AC_TOTAL_PARTS	完成したワークの総数(実際のワークの総数) このカウンタは、起動時から加工されるワークの総数を指定します。この値は、初期値により、制御装置の電源投入時にのみ、自動的に「0」にリセットされます。
$AC_ACTUAL_PARTS	完成したワーク数(実際のワークの総数) このカウンタは、起動時から加工されるワークの総数を登録します。このカウンタは、$AC_REQUIRED_PARTS > 0 の場合に限り、必要なワーク数($AC_REQUIRED_PARTS)に達すると、自動的に「0」にリセットします。
$AC_SPECIAL_PARTS	ユーザーが選択したワーク数 このカウンタで、ユーザー用のワークカウントを指定できます。指令ワーク数の値($AC_REQUIRED_PARTS)に達したときに出力されるアラームを定義できます。ユーザーが、ユーザ自身でカウンタをリセットしてください。

注記

初期値により、制御装置の電源投入時に、すべてのワークカウンタは「0」に設定され、その起動とは無関係に、読み取りと書き込みができます。

注記

チャネル別マシンデータを使用すると、カウンタの起動タイミング、カウンタのリセットのタイミング、およびカウントアルゴリズムを制御できます。

注記

ユーザー定義の M 命令によるワークのカウント

さまざまなワークカウンタのカウントパルスが、エンドオブプログラム(M2/M30)ではなく、ユーザー定義の M 命令で起動されるように、マシンデータを設定できます。

18.9 Process DataShare - 外部機器/ファイルへの出力(EXTOPEN、WRITE、EXTCLOSE):

パートプログラムから外部デバイス/ファイルへのデータの書き込みは、3 つのステップで実行されます。

1. 外部機器/ファイルを開きます。
   外部機器/ファイルが、EXTOPEN 命令を使用して書き込みをおこなうために、そのチャネルに対して開かれます。
2. データの書込み
   出力データは、NC 言語の文字列機能、たとえば SPRINT を使用して処理することができます。書き込みには WRITE 命令が使用されます。
3. 外部機器/ファイルを開きます。
   プログラムの終わりに達した場合(M30)、またはチャネルのリセット時に、EXTCLOSE 命令を使用して外部機器/ファイルがもう一度解放されます。

構文

```
DEF INT <結果>
DEF STRING[<n>] <出力>
…
EXTOPEN(<結果>,<ExtDev>,<SyncMode>,<AccessMode>,<WriteMode>)
…
<Output>="データ出力"
WRITE(<結果>,<ExtDev>,<出力>)
…
EXTCLOSE(<結果>,<ExtDev>)
```

意味

EXTOPEN:	外部機器/ファイルを開くための予約手順命令です		
<結果>:	**パラメータ 1:**結果変数 結果変数値を使用して、操作が正常におこなわれたかどうかをプログラムで評価し、処理を適切に続行することができます。		
	タイプ:	INT	
	値:	0	エラーなし
		1	外部機器を開けません。
		2	外部機器が設定されていません。
		3	無効なパスを持つ外部機器が設定されています。
		4	外部機器のアクセス権限がありません。
		5	使用モード:外部機器がすでに「占有」されています。
		6	使用モード:外部機器がすでに「共有」されています。
		7	フィールド長が LOCAL_DRIVE_MAX_FILESIZE を超えています。
		8	外部機器の最大数を超えています。
		9	LOCAL_DRIVE のオプションが設定されていません。
		11	V.24 インタフェースが Easy-Message 機能ですでに割り当てられています(828D のみ)。
		12	書き込みモード:データが extdev.ini と矛盾しています。
		16	無効な外部パスがプログラム指令されています。
		22	外部機器が搭載されていません。

<table>
<tr><td><ExtDev>:</td><td colspan="2">パラメータ 2:開こうとしている外部機器/ファイルのシンボルの識別子</td></tr>
<tr><td></td><td>タイプ:</td><td>STRING</td></tr>
<tr><td></td><td colspan="2">シンボル識別子は以下で構成されます。
1. 論理機器名称
2. 必要に応じて、その後にファイルパスが続きます(「/」を使用して付加します)。
以下の論理機器名称が定義されています。</td></tr>
<tr><td></td><td>"LOCAL_DRIVE":</td><td>ローカル CF カード(定義済み)</td></tr>
<tr><td></td><td>"CYC_DRIVE":</td><td>SIEMENS サイクルで使用するために予約されたドライブ名称(定義済み)</td></tr>
<tr><td></td><td>"/dev/ext/1",...
"/dev/ext/9":</td><td>使用可能なネットワークドライブ
注:
extdev.ini ファイルで設定する必要があります。</td></tr>
<tr><td></td><td>"/dev/cyc/1",
"/dev/cyc/2":</td><td>SIEMENS サイクルで使用するために予約されているドライブ名称
注:
extdev.ini ファイルで設定する必要があります。</td></tr>
<tr><td></td><td>"/dev/v24":</td><td>V.24 インタフェース
注:
extdev.ini ファイルで設定する必要があります。</td></tr>
<tr><td></td><td colspan="2">ファイルパス:
● "LOCAL_DRIVE"や"CYC_DRIVE"などに対してファイルパスを指定してください。
"LOCAL_DRIVE/my_dir/my_file.txt"
● 論理機器名称"/dev/ext/1...9"と"/dev/cyc/1...2"を以下のように設定できます。
– すでに参照されたファイルに対して設定する場合は、論理機器名称だけを指定できます。例:
"/dev/ext/4"
– ディレクトリの参照用に設定する場合は、ファイルパスを指定してください。例:
"/dev/ext/5/my_dir/my_file.txt"
● ファイルパスを"/dev/v24"に追加することはできません。</td></tr>
<tr><td></td><td colspan="2">注:
論理機器名称"/dev/ext/1...9"、"/dev/v24"、"/dev/cyc/1...2"の場合、大文字/小文字は無視されます。大文字/小文字は、ファイルへのパスを</td></tr>
</table>

	指定する場合に意味を持ちます。"LOCAL_DRIVE"と"CYC_DRIVE"では、大文字だけが使用できます。		
<SyncMode>:	**パラメータ 3:**当該機器/ファイルへの WRITE 命令の処理モード		
	タイプ:	STRING	
	値:	"SYN":	同期書き込み 書き込み操作が終了するまでプログラムの実行が停止します。 同期書き込み操作が正常に完了したかどうかは、WRITE 命令のエラー変数を評価することで確認できます。
		"ASYN":	非同期書き込み プログラムの実行は WRITE 命令で中断されません。 **注:** このモードでは、WRITE 命令の結果変数で情報は提供されず、結果変数は常に値 0(エラーなし)です。この特別なモードでは、WRITE 命令が正常に実行されたかどうかは確認できません。
<AccessMode>:	**パラメータ 4:**当該機器/ファイルの使用モード		
	タイプ:	STRING	
	値:	"SHARED":	機器/ファイルが「共有」モードで要求されます。このモードでは、他のチャネルも機器を使用することができます。つまり、機器を開くことができます。
		"EXCL":	機器/ファイルが当該チャネルだけで使用されます。他のチャネルは機器を使用できません。

<WriteMode>:	**パラメータ 5:**当該ファイル/機器への WRITE 命令の書き込みモード (任意選択)		
	タイプ:	STRING	
	値:	"APP":	付加 ファイルの内容は常に保持されます。書き込み呼び出しは最後に付加されます。
		「OVR」:	上書き ファイルの内容は削除され、その後の書き込み呼び出しを使用して再生成されます。
	注: extdev.ini ファイルで設定された書き込みモードは、このパラメータを使用して上書き できません。干渉が発生した場合、EXTOPEN 呼び出しがエラーで通知されます。		

WRITE:	出力データを書き込むための予約手順

EXTCLOSE:	開かれている外部機器/ファイルを閉じるための予約手順です。		
<結果>:	**パラメータ 1:**結果変数		
	タイプ:	INT	
	値:	0	エラーなし
		16	無効な外部パスがプログラム指令されています。
		21	外部機器を閉じるときにエラーが発生しました。
<ExtDev>:	**パラメータ 2:** 閉じようとしている外部機器/ファイルのシンボル識別子。EXTOPEN を参照してください。 **注:** 識別子は EXTOPEN 呼び出しで指定した識別子と同一のものにしてください。		

例

プログラムコード

```
N10       DEF INT RESULT
N20       DEF BOOL EXTDEVICE
N30       DEF STRING[80] OUTPUT
```

プログラムコード

```
N40        DEF INT PHASE
N50        EXTOPEN(RESULT,"LOCAL_DRIVE/my_file.txt","SYN","SHARED")
N60        IF RESULT > 0
N70           MSG("error for EXTOPEN:" << RESULT)
N80        ELSE
N90           EXTDEVICE=TRUE
N100       ENDIF
…
N200       PHASE=4
N210       IF EXTDEVICE
N220          OUTPUT=SPRINT("End phase: %D",PHASE)
N230          WRITE(RESULT,"LOCAL_DRIVE/my_file.txt",OUTPUT)
N240       ENDIF
…
```

下記も参照

文字列演算子 (ページ 97)

ファイルの書き込み(WRITE) (ページ 168)

18.10 アラーム(SETAL)

アラームは、NC プログラムで設定できます。アラームは、操作画面で個別のフィールドに表示されます。アラームは常に、アラームのカテゴリに従い、コントローラからの応答と連動します。

参照先:
アラーム応答について詳しくは、試運転マニュアルを参照してください。

構文

```
SETAL(<アラーム番号>[,<文字列>])
```

意味

SETAL:	アラームをプログラム指令するためのキーワードです。 SETAL は、個別の NC ブロックでプログラム指令してください。
<アラーム番号>:	INT タイプの変数です。アラーム番号が含まれます。 有効なアラーム番号の範囲は 60000 ～ 69999 です。このうち、60000 ～ 64999 は SIEMENS サイクル用に予約されているため、ユーザーが使用できる範囲は 65000 ～ 69999 です。
<文字列>:	さらに、ユーザーサイクルアラームのプログラム指令時には、4 個までのパラメータを含む文字列を指定できます。 これらのパラメータには可変ユーザーテキストを定義できます。 また、以下の予約パラメータを使用できます。

パラメータ	意味
%1	チャネル番号
%2	ブロック番号、ラベル
%3	サイクルアラームのテキストインデックス
%4	追加のアラームパラメータ

注記
アラームテキストは、操作画面で設定してください。

注記

操作画面で有効になっている言語でアラームを出力する場合、現在 HMI で設定されている言語についての情報が必要です。この情報は、システム変数$AN_LANGUAGE_ON_HMIを使用してパートプログラムとシンクロナイズドアクションで問い合わせることができます(「HMI で現在設定されている言語 (ページ 1141)」を参照してください)。

例

プログラムコード	コメント
...	
N100 SETAL (65000)	; アラーム番号 65000 設定
...	

18.11 停止延長と退避(ESR)

停止延長と退避機能(以降、ESR と呼びます)は、障害状況が発生した場合に、処理に応じてフレキシブルな対応ができるようにします。

- **停止延長**
  特定の障害状況で可能であれば、停止延長を有効にしたすべての軸を正しい手順で停止させます。
- **退避**
  現在使用中の工具をできるだけ素早くワークから退避させます。
- **ジェネレータ運転(SINAMICS ドライブ機能「Vdc 制御」)**
  電源電圧の故障などによって、パラメータ設定可能な DC リンク電圧値が降下した場合、この目的(ジェネレータ運転)のためにドライブの制動エネルギーを回生して退避に必要な電力エネルギーを生成します。

トリガソース

一般ソース(NC 外部/グローバルまたはモードグループ/チャネル別の):

- ディジタル入力(例: NCU モジュールまたは読み返しが可能な制御装置内部のディジタル出力イメージで)($A_IN、$A_OUT)
- チャネル状態($AC_STAT)
- VDI 信号($A_DBB)
- 多数のアラームを含むグループメッセージ($AC_ALARM_STAT)

軸ソース

- スレーブ軸の緊急退避のしきい値(電子連結の同期制御、$VA_EG_SYNCDIFF[<スレーブ軸>])
- ドライブ:DC リンクの警告レベル(不足電圧寸前)、$AA_ESR_STAT[<軸>]
- ドライブ:ジェネレータの最低速度レベル(使用可能な回生回転エネルギーがこれ以上ない)、$AA_ESR_STAT[<軸>]

静的シンクロナイズドアクションのゲート論理:ソース/応答リンク

静的シンクロナイズドアクションのフレキシブルなゲート機能を使用すると、ソースに応じた特定の応答を比較的素早く返すことができます。

静的シンクロナイズドアクションを使用するすべての関連ソースのリンクは、ユーザーの責任でおこないます。これにより、ソースのシステム変数を全体としてまたはビット

入力によって的確に評価し、必要な応答を返す論理演算をおこなうことができます。静的シンクロナイズドアクションは、すべての運転モードで有効です。

参照先:
機能マニュアル シンクロナイズドアクション

起動

機能の有効化

ジェネレータ運転、停止、退避の機能は、対応する制御信号$AA_ESR_ENABLE の設定によって解除されます。この制御信号は、シンクロナイズドアクションで変更できます。

機能の起動

システム変数$AC_ESR_TRIGGER を設定すると、有効なすべての軸でまとめて ESR が起動されます。

ジェネレータ運転は、DC リンクの不足電圧寸前の状態が検出されたときに、ドライブで「自動的に」開始されます。

ドライブに依存しない停止や退避は、通信障害(NC とドライブ間)が検出されてドライブで DC リンクの不足電圧が検出されたときに有効になります(設定と有効化が必要)。

また、ドライブに依存しない停止や退避は、適切な制御信号$AN_ESR_TRIGGER(すべてのドライブに対するブロードキャスト命令)を設定して NC によって起動することもできます。

参照先

ESR について詳しくは、以下を参照してください。

機能マニュアル 応用機能; 停止延長と退避(R3)

18.11.1 NC 制御 ESR

18.11.1.1 NC 制御による退避(POLF、POLFA、POLFMASK、POLFMLIN)

NC 制御による後退には、特定の初期条件が必要です(「NC 制御による退避(POLF、POLFA、POLFMASK、POLFMLIN) (ページ 802)」を参照してください)。この必要条件を満たしている場合は、システム変数$AC_ESR_TRIGGER(または一軸用の$AA_ESR_TRIGGER)を設定すると、チャネル内の 1 つまたは複数の後退軸に対して設定された速リトラクト(LIFTFAST)が起動されます。

構文

```
POLF(<軸>)=<位置>
POLFA(<軸>,<タイプ>,<位置>)
POLFMASK(<軸_1>,<軸_2>,...)
POLFMLIN(<軸_1>,<軸_2>,...)
```

```
POLFA には以下の省略形式を使用できます。
POLFA(<軸>,<タイプ>) ; 単独軸の後退の省略形
POLFA(軸,0/1/2): 簡単な無効化/有効化
POLFA(軸,0,$AA_POLFA[軸]): 先読み停止をおこなう
POLFA(軸,0): 先読み停止をおこなわない
```

意味

POLF:	後退軸の目標位置を指定するアドレス POLF はモーダルです。	
	<軸>:	後退するジオメトリ軸またはチャネル/機械軸の名称
	位置>:	後退位置
		タイプ: REAL
		ジオメトリ軸には WCS、その他の場合は MCS が使用されます。 ジオメトリ軸とチャネル/機械軸が同じ識別子を持つ場合は、後退は WCS でおこなわれます。

POLFA:	一軸の後退位置を指定する予約サブプログラム呼び出し			
	<軸>:	チャネル軸識別子		
	<タイプ>:	位置指定モード		
		タイプ:	INT	
		値:	0:	位置値を無効としてマーク
			1:	位置値はアブソリュート
			2:	位置値はインクレメンタル(距離)
		注: 軸が一軸でないか、タイプが欠落している、またはタイプ=0 の場合、対応するアラームが出力されます。		
	位置>:	後退位置(上記を参照してください)		
		注: 位置値はタイプ=0 でも確定できます。この値は無効としてマークするだけで、後退させるには位置値を再プログラム指令してください。		
POLFMASK:	高速リトラクトの始動後に**互いに無関係に**後退する軸を選択する予約サブプログラム呼び出し			
	<軸_1>,…:	高速リトラクト時に POLF で定義した位置に移動する軸の名称 指定する軸はすべて同じ座標系に属するものにしてください。		
	POLFMASK()で軸を指定しない場合は、互いに無関係に後退するすべての軸の高速リトラクトが解除されます。			
POLFMLIN:	高速リトラクトの始動後に**直線補間で**後退する軸を選択する予約サブプログラム呼び出し			
	<軸_1>,…:	上記を参照してください。		
	POLFMLIN()で軸を指定しない場合は、直線補間で後退するすべての軸の高速リトラクトが解除されます。			

注記

POLFMASK または POLFMLIN で固定位置への高速リトラクトを有効にする前に、選択した軸に対して POLF で位置をプログラム指令してください。

注記

軸を 1 つずつ POLFMASK、POLFMLIN または POLFMLIN、POLFMASK の順に有効にした場合、それぞれの軸に対して常に最後の定義が適用されます。

注記

POLF でプログラム指令した位置と、POLFMASK または POLFMLIN による有効化は、パートプログラムの起動時に解除されます。つまり、パートプログラム毎に POLF の値と POLFMASK または POLFMLIN で選択する軸を再プログラム指令してください。

注記

省略形式の POLFA を使用する場合は、タイプのみが変更されます。そのため、後退位置か後退軌跡のどちらかに実際の意味のある値が含まれていることを確認してください。特に、後退位置と後退軌跡は、電源投入後に再設定してください。

例

個々の軸の後退:

プログラムコード	コメント
MD37500 $MA_ESR_REACTION[AX1]=21	; NC 制御による後退
...	
$AA_ESR_ENABLE[AX1] = 1	
POLFA(AX1,1,20.0)	; AX1 に軸の後退位置 20.0(アブソリュート)を割り当てます。
$AA_ESR_TRIGGER[AX1]=1	; 後退はここから開始します。

詳細情報

NC 制御による退避の必要条件

- チャネルで NC 制御による退避のための後退軸を設定します。
  MD37500 $MA_ESR_REACTION = 21
- この軸に対して ESR を有効にしてください。
  $AA_ESR_ENABLE = 1
- 遅延時間を定義します。
  MD21380 $MC_ESR_DELAY_TIME1
  MD21381 $MC_ESR_DELAY_TIME2
- パートプログラムで POLF を使用して軸別の後退位置を設定しています。

- POLFMASK/POLFMLIN を使用して NC 制御による後退のための軸を選択します。
- 後退移動のための有効な信号を設定し、設定したままにしてください。

NC 制御による退避の有効化と起動

システム変数$AC_ESR_TRIGGER = 1 を設定し、このチャネルに後退軸を設定し(MD37500 $MA_ESR_REACTION = 21 など)、この軸に対して$AA_ESR_ENABLE = 1 を設定すると、このチャネルで高速リトラクト(LIFTFAST)が有効になります。

POLFMASK または POLFMLIN で選択した軸に対して、POLF(または POLFA)で設定した後退移動が、パートプログラムでこれらの軸に対して定義された軌跡移動の代わりに使用されます。

MD21380 $MC_ESR_DELAY_TIME1 と MD21381 $MC_ESR_DELAY_TIME2 の合計時間が後退に使用できる最大時間です。 この時間が経過すると、後退軸に対してフォローアップを伴う急減速も開始されます。

注記

拡張後退($AC_ESR_TRIGGER によって起動された LIFTFAST/LFPOS など)は**中断できず**、非常停止でのみ、すぐに終了できます。

注記

$AC_ESR_TRIGGER によって起動された後退は、複数の後退を防ぐためにロックされています。

一軸の後退

一軸の後退では、POLFA で一軸の後退位置をプログラム指令し、次の条件を満たしてください。

- $AA_ESR_ENABLE = 1
- 起動時($AAAA_ESR_TRIGGER = 1)には**<軸>**を一軸にしてください。
- **<タイプ>**は 1 または 2 にしてください。

高速リトラクト時の後退方向

高速リトラクトを開始したときに有効なフレームが考慮されます。

注記

回転有りのフレームも POLF によるリトラクトの方向に影響します。

軸入れ替え

後退軸は常に 1 つの NC チャネルのみに割り当てて、チャネル間で切り替えないでください。 後退軸を別のチャネルに入れ替えようとすると、アラームが出力されます。 $AA_ESR_ENABLE[AX] = 0 を使用してこの軸を無効にした後でのみ、新しいチャネルに入れ替えることができます。 軸の入れ替え後、$AA_ESR_ENABLE[AX] = 1 によって軸を再度有効にできます。

中立軸

中立軸は NC 制御による ESR を実行できません。

18.11.1.2 NC 制御による停止

NC 制御による停止は、チャネルでシステム変数$AC_ESR_TRIGGER(または一軸用の $AA_ESR_TRIGGER)によって設定した停止軸に対して適用されます。

必要条件

- チャネルで NC 制御による停止を適用する停止軸を設定します。
  MD37500 $MA_ESR_REACTION = 22
- この軸に対して ESR を有効にしてください。
  $AA_ESR_ENABLE = 1
- 遅延時間を定義します。
  MD21380 $MC_ESR_DELAY_TIME1(遅延時間、ESR 軸)
  MD21381 $MC_ESR_DELAY_TIME2 (補間減速のための ESR 時間)

実行

この軸は、MD21380 で設定した時間の間、プログラム指令に従って補間を続けます。 MD21380 で指定した遅延時間の経過後、制御減速(減速停止)が開始されます。MD21381 の時間は、補間制御減速に使用できる最大時間です。この時間の経過後、高速制動とその後の実速度追従が開始されます。

例

一軸の停止:

プログラムコード	コメント
`MD37500 $MC_ESR_REACTION[AX1] = 22`	`; NC 制御による停止`
`MD21380 $MC_ESR_DELAY_TIME1[AX1] = 0.3`	
`MD21381 $MC_ESR_DELAY_TIME2[AX1] = 0.06`	

プログラムコード	コメント
...	
$AA_ESR_ENABLE[AX1] = 1	
$AA_ESR_TRIGGER[AX1]=1	; ここから停止が開始します。

18.11.2 ドライブ統合 ESR

18.11.2.1 ドライブ内蔵の停止の設定(ESRS)

ドライブ内蔵の ESR 機能の「停止」のためのドライブパラメータは、ESRS(...)機能を使用して設定されます。

構文

ESRS(<アクセス_1>,<停止時間_1>[,...,<軸_n>,<停止時間_n>])

意味

ESRS(...):	ESR 機能「停止」のためにドライブパラメータに書き込む機能 この機能は: ● ブロック内では単独にします。 ● 先読み停止をおこないます。 ● シンクロナイズドアクションでは使用できません。	
<軸_1>, ..., <軸_n>:	ドライブ内蔵の停止を設定する軸 この軸に対して、ドライブパラメータ p0888 (設定)がドライブに書き込まれます。 p0888 = 1	
	タイプ:	AXIS
	値の範囲:	チャネル軸識別子

<停止時間_1>, ..., <停止時間_n>:	故障発生後にドライブが実速度指令で移動を続ける時間 指定された軸に対して、ドライブパラメータ p0892 (タイマ)がドライブに書き込まれます。 p0892 = <停止時間>	
	単位:	s
	タイプ:	REAL
	値の範囲:	0.00 - 20.00
ファンクションコールで最大 5 軸までプログラム指令できます; n = 5		

18.11.2.2 ドライブ内蔵の後退の設定(ESRS)

ドライブ内蔵の ESR 機能の「後退」のためのドライブパラメータは、ESRR(...)機能を使用して設定されます。

構文

```
ESRR(<軸_1>,<後退距離_1>,<後退速度_1>[,...,<軸_n>,<後退距離_n>,<後退速度_n>])
```

意味

ESRR(...):	ESR 機能「後退」のためにドライブパラメータに書き込む機能 この機能は: ● ブロック内では単独にします。 ● 先読み停止をおこないます。 ● シンクロナイズドアクションでは使用できません。	
<軸_1>, ..., <軸_n>:	ドライブ内蔵の後退を設定する軸 この軸に対して、ドライブパラメータ p0888 (設定)がドライブに書き込まれます。 p0888 = 2	
	タイプ:	AXIS
	値の範囲:	チャネル軸識別子

<後退量_1>, ..., <後退量_n>:	ドライブの場合、後退量が後退速度に変換されます。指定された軸の場合、値はドライブパラメータ p0893 (速度)に書き込まれます。 p0893 = (<後退量_n>後退速度に変換)	
	単位:	mm/min、inch/min、°/min(軸の単位による)
	タイプ:	REAL
	値の範囲:	MIN - MAX
<後退 速度_1>, ..., <後退 速度_n>:	ドライブの場合、後退速度が時間に変換されます。指定された軸の場合、値はドライブパラメータ p0892 (タイマ)に書き込まれます。 p0892 = <後退量_n> / <後退速度_n>	
	単位:	mm/min、inch/min、°/min(軸の単位による)
	タイプ:	REAL
	値の範囲:	0.00 - MAX
ファンクションコールで最大 5 軸までプログラム指令できます; n = 5		

18.12 Define blank (WORKPIECE)

グラフィカルシミュレーションでブランクの形状とサイズを表示するには、コントローラがそれらを認識している必要があります。したがって、ユーザーはユーザーインタフェース経由で、または NC プログラム内で直接、ブランクを定義できます。ブランクの定義は(プログラム終了/チャネル/モードグループ)のリセット後も維持されます。これらの定義は、コントローラの次回の電源投入時に自動的に削除されます。

構文

```
WORKPIECE("<WP>", "<RefP>", "<ZeroOffset>", "<Type>", <Par5>,
<Par6>, ..., <Par12>)
```

意味

WORKPIECE(...):		ブランクを定義するための予約処理		
		先読み停止:	あり	
		単独ブロック指令:	あり	
パラメータ:				
1	"<WP>":	ワークの名前(オプション)		
		データタイプ:	STRING	
		指定する必要があるのは、1 つのチャネルに複数のワークが存在する可能性がある場合のみです。指定しない場合、「WORKP<n>」で自動的に確定します。ここで、<n>は宣言するチャネルの数です。		
2	"<RefP>":	クランプ(オプション、フライス盤のみ)		
		データタイプ:	STRING	
		値の範囲:	"Table"	固定テーブルのクランプ
			"A"	回転軸 A にクランプ
			"B"	回転軸 B にクランプ
			"C"	回転軸 C にクランプ
		必要条件: ブランクのクランプ用マシンデータを使用して、テーブルまたは回転軸を有効にする必要があります。 (『SINUMERIK Operate 試運転マニュアル』を参照)。		

3	"<ZeroOffset>":	ブランクを位置決めするために設定可能なゼロオフセット(プログラム不可能) ブランクを位置決めするために設定可能なゼロオフセットの選択は、ユーザーインタフェースを使用したブランク入力でのみ可能です。パートプログラムでブランクを直接定義する場合、ブランクは常に現在有効なゼロオフセットに関連します。		
4	"<Type>":	素材形状		
		データタイプ:	STRING	
		値の範囲:	"CYLINDER":	円筒
			"PIPE":	風管
			"RECTANGLE":	中心のある直方体
			"BOX":	直方体
			"N_CORNER":	n 辺の多角形
5 ... 12	<Par5> ... <Par12>:	ブランクの形状を説明するためのパラメータ		
		データタイプ:	REAL	
		必要なパラメータ数とその意味は、対応するブランクの形状とビットパラメータの値によって異なります。 参照: ● 「ブランクの形状を説明するためのパラメータ」表 ● 「ビットパラメータ」表		
WORPIECE():		パラメータなしの WORKPIECE 呼び出しは、すべてのブランク定義を削除します。		
WORPIECE(<WP>):		ワーク名のみの WORKPIECE 呼び出しは、そのブランク定義のみを削除します。		

表 18-1　ブランクの形状を説明するためのパラメータ

素材形状	パラメータ							
	<Par5>	<Par6>	<Par7>	<Par8>	<Par9>	<Par10>	<Par11>	<Par12>
円筒	ビットパラメータ ビットコード整数値として解釈される実数値。ビットは次のパラメータの意味を定義します(「ビットパラメータ」表を参照)。	基準点 Z_0	長さ Z_1	加工寸法 Z_B	外径 d_0	-	回転軸を中心とする回転	-
風管		基準点 Z_0	長さ Z_1	加工寸法 Z_B	外径 d_0	壁厚(inc)/内径 d_1 (abs)	回転軸を中心とする回転	-
中心のある直方体		基準点 Z_0	長さ Z_1	加工寸法 Z_B	幅 W	長さ L	回転軸を中心とする回転	-
直方体		基準点 Z_0	長さ Z_1	加工寸法 Z_B	X_0	Y_0	X_1	Y_1
n 辺の多角形		基準点 Z_0	長さ Z_1	加工寸法 Z_B	頂点の数	二面幅	回転軸を中心とする回転	-

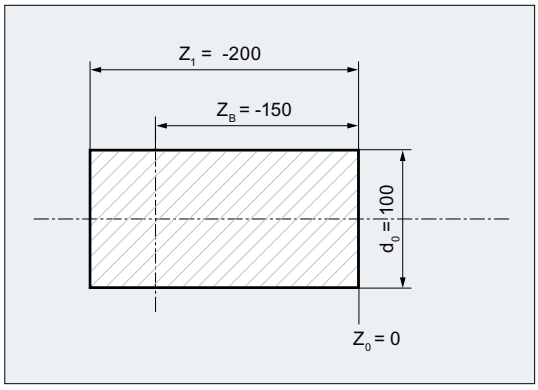
表 18-2　ビットパラメータ

ビット	意味	
4 (0x0010)	直方体:X_1	
	= 0	inc
	= 1	abs
5 (0x0020)	直方体:Y_1	
	= 0	inc
	= 1	abs
6 (0x0040)	長さ Z_1 (最終寸法)	
	= 0	inc
	= 1	abs
ビット 7 (0x0080)	加工寸法 Z_B	
	= 0	inc
	= 1	abs

ビット	意味	
ビット 8 (0x0100)	風管:壁厚/内径	
	= 0	inc
	= 1	abs
9 (0x0200)	n 辺の多角形	
	= 0	二面幅
	= 1	辺の長さ
12 (0x1000)	旋盤のクランプ	
	= 0	主軸
	= 1	対向主軸
13 (0x2000)	対向主軸	
	= 0	ミラーリング付き
	= 1	ミラーリングなし

例

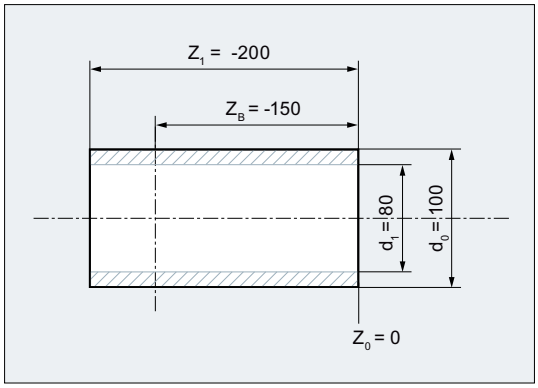
例 1:旋盤上の円筒状ブランク



プログラムコード	コメント
...	

プログラムコード	コメント
WORKPIECE(,,,"CYLINDER",0,0,-200,-150,100)	; **ブランク定義:** ; ブランク形状:円筒 ; ビットパラメータ=0(ビット設定なし) → 長さと加工寸法の値はインクレメンタル、主軸上のブランク ; 基準点(Z0)=0 ; 長さ(Z1)=-200 ; 加工寸法(ZB)=-150 ; 外径(d0)=100
...	

例 2 :旋盤上の風管状ブランク



プログラムコード	コメント
...	
WORKPIECE(,,,"PIPE",256,0,-200,-150,100,80)	; **ブランク定義:** ; ブランク形状:風管 ; ビットパラメータ=256(Bit8=1) →内径は絶対値; 長さと加工寸法はインクレメンタル、主軸上のブランク ; 基準点(Z0)=0 ; 長さ(Z1)=-200 ; 加工寸法(ZB)=-150 ; 外径(d0)=100 ; 内径(d1)=80
...	

18.13 言語切り替えモード(G290, G291)

このコントローラを使用すると、外部 CNC システムからパートプログラムを読み込んで処理できます。対応する NC 言語モード(ISO 系言語)がセットアップ時に定義されていることが前提条件です。

参照先:
『機能マニュアル G コード言語』

ISO 系言語モードは、チャネルごとに別々に有効化できます。たとえば、チャネル 1 を ISO 系言語モードで実行し、チャネル 2 を SINUMERIK モードで有効化することができます。

SINUMERIK モードと ISO 系言語モードの切り替えは、NC プログラムで G グループ 47 の命令を使用しておこなわれます。使用中の工具、工具補正、ワークオフセットはモード切り替えの影響を受けません。

構文

```
G291
...
G290
```

意味

G290:	SINUMERIK 言語モードの起動	
	単独ブロック指令:	あり
	効果:	モーダル
G291:	ISO 言語モードの起動	
	単独ブロック指令:	あり
	効果:	モーダル

条件

SINUMERIK モード

- G 命令の規定値はマシンデータを使用してチャネルごとに定義できます。
- ISO 系言語の言語指令を SINUMERIK モードでプログラムすることはできません。

ISO 系言語モード

- コントロールシステムの初期設定として、マシンデータで ISO 系言語モードを設定できます。初期設定では、その後、ISO 系言語モードでコントロールシステムが再起動されます。
- ISO 系言語の G 命令だけがプログラム指令できます。SINUMERIK G 機能は ISO 系言語モードではプログラム指令できません。
- ISO 系言語と SINUMERIK 言語を同一 NC ブロックで混在させることはできません。
- G 指令で ISO 系言語 M(フライス削り)と ISO 系言語 T(旋盤)の切り替えはできません。
- SINUMERIK モードでプログラムされたサブプログラムを呼び出すことができます。
- SINUMERIK 機能を使用する場合は、まず SINUMERIK モードに切り替える必要があります(例を参照)。

例

ISO 系言語モードでの直線ブロックの圧縮

プログラムコード	コメント
N5 G290	; SINUMERIK 言語モードの起動。
N10 COMPON	; COMPON は Siemens 言語のコマンドで、連続する直線ブロックを、可能なかぎり長い軌跡長の多項式ブロックで置き換えるコンプレッサ機能を有効化します。
N15 G291	; ISO 言語モードを起動します。
N20 G01 X100 Y100 F1000	; COMPON が SINUMERIK モードで有効化されているため、ISO 系言語モードの直線ブロックも、この機能を使用して圧縮できます。
...	

ユーザー荒削りプログラム 19

19.1 荒削りの機能のサポート

事前にプログラム指令された切削プログラムを切削に使用できます。それ以外に、以下の機能を使用して、ユーザー専用の切削プログラムを作成できます。

- 輪郭テーブルの生成(CONTPRON)
- コード化された輪郭テーブルの生成(CONTDCON)
- 輪郭解析の解除(EXECUTE)
- 2 個の輪郭要素の交点の特定(INTERSEC)
  (CONTPRON で生成したテーブルの場合のみ有効です)
- テーブルの輪郭要素をブロック毎に実行(EXECTAB)
  (CONTPRON で生成したテーブルの場合のみ有効です)
- 円弧データの計算(CALCDAT)

注記

これらの機能は、切削だけでなく、汎用的に使用できます。

必要条件

CONTPRON または CONTDCON 機能を呼び出す前に、以下をおこなってください。

- 衝突が発生しないで加工が可能な起点に移動してください。
- G40 で、工具径補正を解除してください。

19.2 輪郭テーブルの生成(CONTPRON)

CONTPRON は輪郭解析を起動します。その後に呼び出される NC ブロックは実行されず、個々の移動に分割されて、輪郭テーブルに格納されます。各輪郭要素は、輪郭テーブルの 2 次元配列の 1 行に対応します。レリーフカットの数が返されます。

構文

輪郭解析の起動:
CONTPRON(<輪郭テーブル>,<加工タイプ>,<レリーフカット>,
<加工方向>)

輪郭解析の解除と通常の実行モードへの復帰:
EXECUTE(<ERROR>)

「輪郭解析の解除(EXECUTE) (ページ 836)」を参照してください。

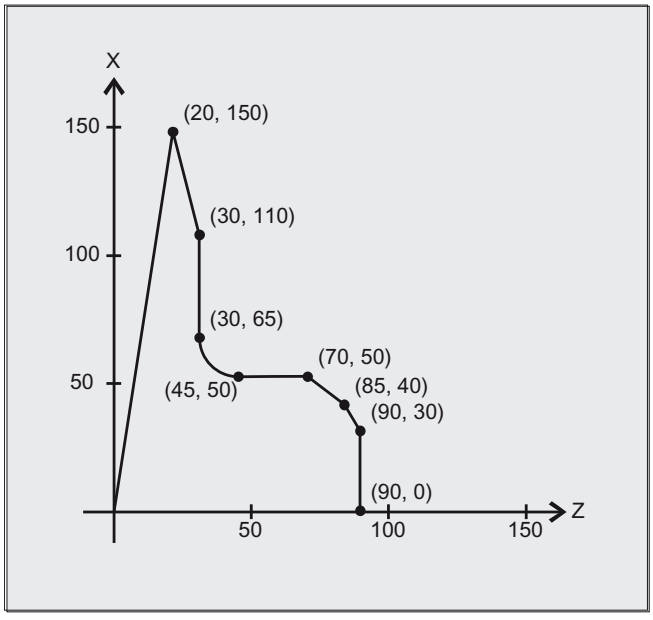
意味

CONTPRON:	輪郭解析を起動して輪郭テーブルを生成する予約処理		
<輪郭テーブル>:	輪郭テーブルの名称		
<加工タイプ>:	加工タイプのパラメータ		
	タイプ:	CHAR	
	値:	"G":	長手方向の旋削:内径加工
		"L":	長手方向の旋削:外側加工
		"N":	正面旋削:内径加工
		"P":	正面旋削:外側加工
<レリーフカット>:	発生するレリーフカット要素数として得られる変数		
	タイプ:	INT	
<加工方向>:	加工方向のパラメータ		
	タイプ:	INT	
	値:	0	輪郭解析、前方(初期値)
		1	両方向の輪郭解析

例 1

以下で輪郭テーブルを生成します:

- 名称「KTAB」
- 最大 30 個の輪郭要素(円弧、直線)
- 発生するレリーフカット要素数を表わす 1 個の変数
- エラーメッセージを表わす 1 個の変数



NC プログラム:

プログラムコード	コメント
N10 DEF REAL KTAB[30,11]	; 最大 30 個の輪郭要素を持つ、KTAB という名称の輪郭テーブル、パラメータ値 11 (テーブルの列数) は固定数です。
N20 DEF INT ANZHINT	; ANZHINT という名称のレリーフカット要素の数をを表わす変数。
N30 DEF INT ERROR	; エラーフィードバック信号を表わす変数(0=エラーなし、1=エラーあり)。
N40 G18	
N50 CONTPRON(KTAB,"G",ANZHINT)	; 輪郭解析の起動

プログラムコード	コメント
N60 G1 X150 Z20	; N60 ～ N120:輪郭の記述
N70 X110 Z30	
N80 X50 RND=15	
N90 Z70	
N100 X40 Z85	
N110 X30 Z90	
N120 X0	
N130 EXECUTE(ERROR)	; 輪郭テーブルへの書き込みを終了し、通常のプログラム\モードに切り替えます。
N140 …	; テーブルの処理を続行します。

輪郭テーブル KTAB

インデックス 行	列									
(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

列の内容の説明

(0) 次の輪郭要素(その列の行番号)へのポインタ

(1) 前の輪郭要素へのポインタ

(2) 輪郭モードの動作コード

X の指令値 = 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) 輪郭要素の始点

(3) =横座標、(4) =現在の平面上の縦座標

(5), (6) 輪郭要素の始点

(5) =横座標、(6) =現在の平面上の縦座標

(7) 最大/最小値:輪郭の部分の最大値と最小値を示します

(8) 輪郭要素と、横座標(長手方向の加工の場合)、または縦座標(正面旋削の場合)の間の最大値。角度は、プログラム指令した加工のタイプによって異なります。

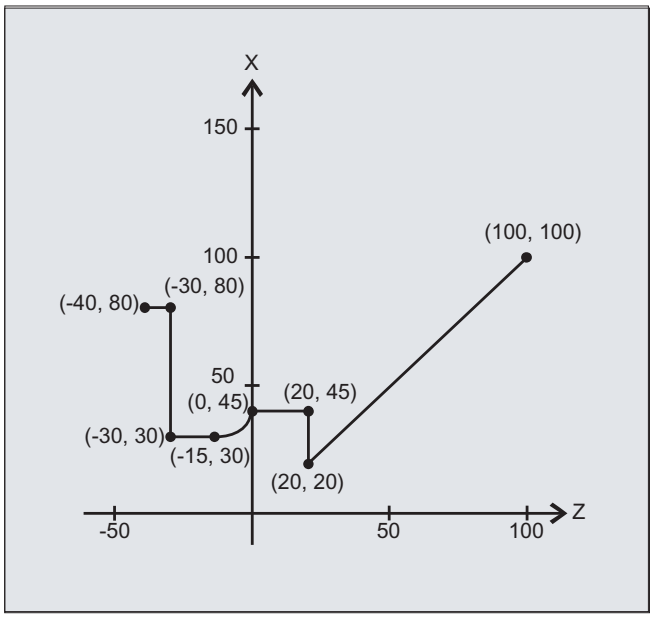
(9), (10) 円弧ブロックの場合の、輪郭要素の中心点座標です。

(9) =横座標、(10) =縦座標

例 2

以下のような輪郭テーブルを生成します。

- 名称「KTAB」
- 最大 92 個の輪郭要素(円弧、直線)
- 運転モード:長手方向の旋削、外側加工
- 解析、前方と後方



NC プログラム

プログラムコード	コメント
N10 DEF REAL KTAB[92,11]	; 最大 92 個の輪郭要素を持つ、KTAB という名称の輪郭テーブルです、パラメータ値 11 は固定数です。
N20 DEF CHAR BT="L"	; CONTPRON のモード:長手方向の旋削、外側加工
N30 DEF INT HE=0	; レリーフカット要素の数=0
N40 DEF INT MODE=1	; 解析、前方と後方
N50 DEF INT ERR=0	; エラーフィードバック信号
...	
N100 G18 X100 Z100 F1000	
N105 CONTPRON(KTAB,BT,HE,MODE)	; 輪郭解析の起動
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)	; 輪郭テーブルへの書き込みを終了し、通常のプログラム\モードに切り替えます。
...	

輪郭テーブル KTAB

輪郭解析が終了した後は、輪郭は両方向に有効です。

インデックス	列										
行	(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

各列の内容の説明と 0、1、6、8、83、85、および 91 行のコメント

例 1 の列の説明が適用されます。

テーブル 0 行では常に:

1) 先行ポインタ:n 行には輪郭終点が含まれます(前方へ)

2) 後続ポインタ:n 行は輪郭テーブルの終点です(前方へ)

輪郭要素内でそれぞれ 1 回前方へ:

3) 先行ポインタ:輪郭始点(前方へ)

4) 後続ポインタ:輪郭終点(前方へ)

常に輪郭テーブル終点(前方へ) +1 行:

5) 先行ポインタ:レリーフカットの数(前方へ)

6) 後続ポインタ:レリーフカットの数(後方へ)

輪郭要素内でそれぞれ 1 回後方へ:

7) 後続ポインタ:輪郭終点(後方へ)

8) 先行ポインタ:輪郭始点(後方へ)

常にテーブルの最終行:

9) 先行ポインタ:n 行は輪郭テーブルの始点です(後方へ)

10) 後続ポインタ:n 行には輪郭始点が含まれます(後方へ)

詳細情報

使用可能な移動指令、座標系

輪郭のプログラミングには、次の G 命令を使用できます。

- G グループ 1:G0、G1、G2、G3

さらに、以下を実行できます。

- 丸み付けと面取り
- CIP と CT による円弧プログラミング

スプライン、多項式、およびねじの各機能を実行すると、エラーが発生します。

CONTPRON と EXECUTE の間でフレームを起動して、座標系を変更することはできません。これは、G70 と G71、または G700 と G710 の間での変更でも同様です。

輪郭テーブルの解析中に、GEOAX でジオメトリ軸を置換すると、アラームが発生します。

レリーフカット要素

サブプログラムまたは個々のブロックで、個々のレリーフカット要素について輪郭を記述できます。

プログラム指令した輪郭方向に影響されない荒削り

CONTPRON による輪郭解析は、プログラム指令方向にかかわらず、呼び出し後に輪郭テーブルを使用できるように拡張されました。

19.3 コード化された輪郭テーブルの生成(CONTDCON)

CONTDCON で起動した輪郭解析の場合は、メモリの使用を最適化するために、呼び出した以下の NC ブロックを、6 列の輪郭テーブルにコード化形式で保存します。各輪郭要素は、輪郭テーブルの 1 行に対応します。以下に示すコード化の規則を熟知しているなどの場合は、テーブル行のサイクルに DIN コードプログラムを組み合わせることもできます。開始点のデータは、番号 0 のテーブル行に保存されます。

構文

輪郭解析の起動:
CONTDCON(<輪郭テーブル>,<加工方向>)

輪郭解析の解除と通常の実行モードへの復帰:
EXECUTE(<ERROR>)

「輪郭解析の解除(EXECUTE) (ページ 836)」を参照してください。

意味

CONTDCON:	輪郭解析を起動して、コード化された輪郭テーブルを生成する予約処理		
<輪郭テーブル>:	輪郭テーブルの名称		
<加工方向>:	加工方向のパラメータ		
	タイプ:	INT	
	値:	0	輪郭ブロックの順序に従った輪郭解析(初期値)
		1	不可

注記

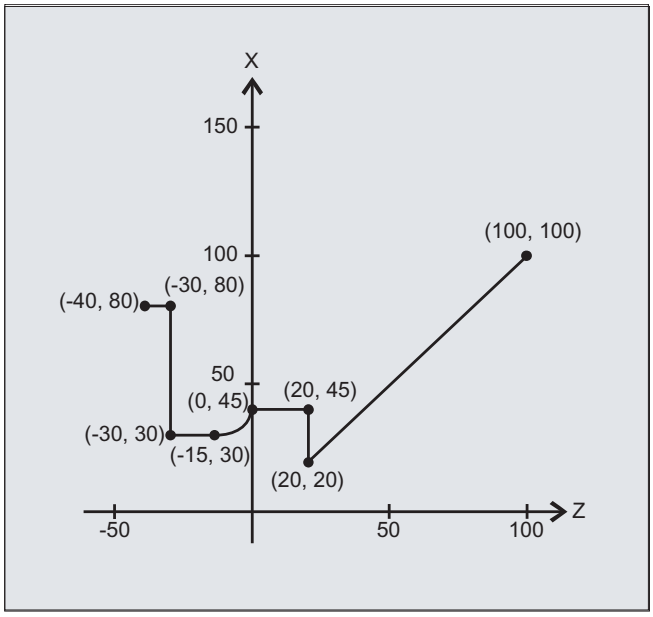
テーブルに含まれるプログラム区間の CONTDCON に使用できる G 命令は、CONTPRON に使用できるものより広範囲です。さらに、送り速度と送り速度タイプを、輪郭区間毎に保存します。

例

以下で輪郭テーブルを生成します:

- 名称「KTAB」
- 輪郭要素(円弧、直線)

- 運転モード:旋削
- 加工方向:前方



NC プログラム:

プログラムコード	コメント
N10 DEF REAL KTAB[9,6]	; 名称が KTAB で、テーブルのセルが 9 個の輪郭テーブルです。これには、輪郭セットを 8 個格納できます。パラメータ値 6 (テーブルの列番号)は固定サイズです。
N20 DEF INT MODE = 0	; 加工方向の変数です。標準値 0:輪郭のプログラム指令方向にのみ有効です。
N30 DEF INT ERROR = 0	; エラーフィードバック信号を示す変数です。
...	
N100 G18 G64 G90 G94 G710	
N101 G1 Z100 X100 F1000	
N105 CONTDCON (KTAB, MODE)	; 輪郭解析を呼び出します(MODE は省略可能です)。
N110 G1 Z20 X20 F200	; 輪郭を記述します。
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)	; 輪郭テーブルへの書き込みを終了し、通常のプログラム\モードに切り替えます。
...	

輪郭テーブル KTAB:

	列インデックス					
	0	1	2	3	4	5
行インデックス	輪郭モード	終点の横座標	終点の縦座標	中心点の横座標	中心点の縦座標	送り速度
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

列の内容の説明:

始点の 0 行目のコード:

0 列目: 10^0(1 の位):G0 = 0
10^1(10 の位):G70 = 0、G71 = 1、G700 = 2、G710 = 3
1 列目: 起点の横座標
2 列目: 起点の縦座標
3 ～ 4 列目: 0
5 列目: テーブルの最後の輪郭部分の行インデックス

1 ～ n 行:**輪郭部分の入力値**

0 列目: 10^0(1 の位):G0 = 0、G1 = 1、G2 = 2、G3 = 3
10^1(10 の位):G70 = 0、G71 = 1、G700 = 2、G710 = 3
10^2 (100 の桁):G90 = 0、G91 = 1
10^3 (1000 の桁):G93 = 0、G94 = 1、G95 = 2、G96 = 3
10^4 (10000 の桁):G60 = 0、G44 = 1、G641 = 2、G642 = 3
10^5 (100000 の桁):G9 = 1
1 列目: 終点の横座標
2 列目: 終点の縦座標

3 列目: 円弧補間の中心点の横座標
4 列目: 円弧補間の中心点の縦座標
5 列目: 送り速度

詳細情報

使用可能な移動指令、座標系

輪郭のプログラミングには、次の G グループと G 命令を使用できます。

G グループ 1: G0、G1、G2、G3
G グループ 10: G60、G64、G641、G642
G グループ 11: G9
G グループ 13: G70、G71、G700、G710
G グループ 14: G90、G91
G グループ 15: G93、G94、G95、G96G961

さらに、以下を実行できます。

- 丸み付けと面取り
- CIP と CT による円弧プログラミング

スプライン、多項式、およびねじの各機能を実行すると、エラーが発生します。

CONTDCON と EXECUTE の間でフレームを起動して、座標系を変更することはできません。これは、G70 と G71、または G700 と G710 の間での変更でも同様です。

輪郭テーブルの解析中に、GEOAX でジオメトリ軸を置換すると、アラームが発生します。

加工方向

CONTDCON で生成した輪郭テーブルを使用して、輪郭のプログラム指令方向に荒削りをおこないます。

19.4 2 個の輪郭要素の交点の特定(INTERSEC)

INTERSEC は、CONTPRON で生成した輪郭テーブルの標準化した 2 個の輪郭要素の交点を特定します。

構文

```
<状態>=INTERSEC(<輪郭テーブル_1>[<輪郭要素_1>],
<輪郭テーブル_2>[<輪郭要素_2>],<交点>,<加工タイプ>)
```

意味

INTERSEC:	CONTPRON で生成した輪郭テーブルから取得した 2 個の輪郭要素の交点を特定する予約機能		
<状態>:	交点状態を表わす変数		
	タイプ:	BOOL	
	値:	TRUE	交点が見つかりました
		FALSE	交点が見つかりません
<輪郭テーブル_1>:	1 番目の輪郭テーブルの名称		
<輪郭要素_1>:	1 番目の輪郭テーブルの輪郭要素の番号		
<輪郭テーブル_2>:	2 番目の輪郭テーブルの名称		
<輪郭要素_2>:	2 番目の輪郭テーブルの輪郭要素の番号		
<交点>:	有効平面(G17 / G18 / G19)上の交点の座標		
	タイプ:	REAL	
<加工タイプ>:	加工タイプのパラメータ		
	タイプ:	INT	
	値:	0	有効平面上のパラメータ 2 による交点演算(標準値)
		1	平面移動に影響されない交点演算

注記

変数は、使用する前に定義してください。

CONTPRON で定義した値を維持したまま、輪郭を移動してください。

パラメータ	意味
2	移動のための輪郭モードのコード
3	輪郭の始点の横座標
4	輪郭の始点の縦座標
5	輪郭の終点の横座標
6	輪郭の終点の縦座標
9	横座標の中心点座標(円弧輪郭の場合のみ)
10	縦座標の中心点座標(円弧輪郭の場合のみ)

例

テーブル TABNAME1 の輪郭要素 3 とテーブル TABNAME2 の輪郭要素 7 の交点演算です。有効平面上の交点座標は変数 ISCOORD に格納されます(1 番目の要素=横座標、2 番目の要素=縦座標)。交点がない場合は、プログラムが NOCUT にジャンプします(交点がみつかりません)。

プログラムコード	コメント
DEF REAL TABNAME1[12,11]	; 輪郭テーブル 1
DEF REAL TABNAME2[10,11]	; 輪郭テーブル 2
DEF REAL ISCOORD [2]	; 交点座標を表わす変数です。
DEF BOOL ISPOINT	; 交点状態を表わす変数です。
DEF INT MODE	; 加工タイプを表わす変数です。
…	
モード= 1	; 演算は有効平面とは無関係におこなわれます。
N10 ISPOINT=INTERSEC(TABNAME1[3],TABNAME2[7], ISCOORD,MODE)	; 輪郭要素の交点を呼び出します。
N20 IF ISPOINT==FALSE GOTOF NOCUT	; NOCUT へジャンプします
…	

19.5 テーブルの輪郭要素をブロック毎に実行(EXECTAB)

EXECTAB を使用して、CONTPRON などで生成したテーブルの輪郭要素をブロック毎に実行できます。

構文

EXECTAB(<輪郭テーブル>[<輪郭要素>])

意味

EXECTAB:	輪郭要素を実行する予約機能
<輪郭テーブル>:	輪郭テーブルの名称
<輪郭要素>:	輪郭要素の番号

例

テーブル KTAB の輪郭要素 0 ～ 2 をブロック毎に実行します。

プログラムコード	コメント
N10 EXECTAB(KTAB[0])	; テーブル KTAB の要素 0 を移動します。
N20 EXECTAB(KTAB[1])	; テーブル KTAB の要素 1 を移動します。
N30 EXECTAB(KTAB[2])	; テーブル KTAB の要素 2 を移動します。

19.6 円弧データの計算(CALCDAT)

CALCDAT を使用して、円弧上の既知の 3 ～ 4 個の点から、円弧の半径と中心点の座標を計算できます。それぞれ異なる点を指定してください。

円弧上に 4 点が直接に存在していない場合は、円弧の中心点と半径に対する平均値が計算されます。

注記

平均の計算規則

円弧の計算は、次の 4 回実行されます。

1. 円弧上の点 1、2、3 で
2. 円弧上の点 1、2、4 で
3. 円弧上の点 1、3、4 で
4. 円弧上の点 2、3、4 で

円弧の中心点の横座標と縦座標の値は、4 つの円弧計算の横座標と縦座標の値を合計して 4 で割って算出されます。

半径は、円弧計算で得られた 4 つの半径の合計の平方根を求め、その結果に 0.5 を乗算して算出されます。

構文

```
<状態>=CALCDAT(<円弧上の点>[<数>,<タイプ>],<数>,<結果>)
```

意味

CALCDAT:	円弧の半径と中心点座標を 3 ～ 4 点から計算する予約機能		
<状態>:	円弧の計算状態を示す変数		
	タイプ:	BOOL	
	値:	TRUE	指定した点が円弧上にあります。
		FALSE	指定した点が円弧上に**ありません**。
<円弧上の点>[]:	円弧上の点を パラメータで指定するための変数		
	<番号>:	円弧上の点の数(3 または 4)	
	<タイプ>:	座標データのタイプ 例: 2 点座標の場合は 2	
<番号>:	計算に使用する点の数を示すパラメータ(3 または 4)		

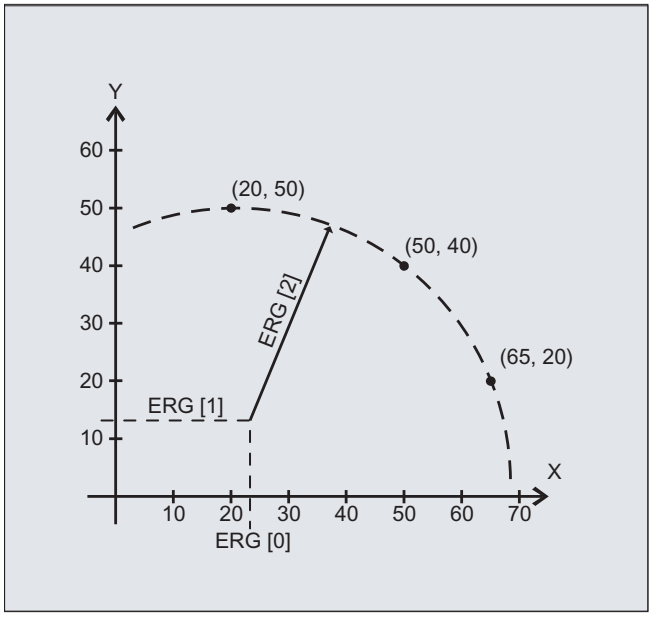
<結果>[3]:	結果を示す変数: 円弧の中心点座標と半径	
	0	円弧の中心点座標:横座標値
	1	円弧の中心点座標:縦座標値
	2	半径

注記

変数は、使用する前に定義してください。

例

3 点を使用して、それらが、1 つの円弧セグメント上にあるかどうかを特定します。



プログラムコード	コメント
N10 DEF REAL PT[3,2]=(20,50,50,40,65,20)	; 円弧の点を指定するための変数
N20 DEF REAL RES[3]	; 結果を示す変数
N30 DEF BOOL STATUS	; 状態を示す変数
N40 STATUS=CALCDAT(PKT,3,ERG)	; 特定した円弧データを呼び出します。
N50 IF STATUS == FALSE GOTOF ERROR	; エラーへジャンプします。

19.7 輪郭解析の解除(EXECUTE)

EXECUTE は輪郭解析を解除します。これと同時に、システムは通常の実行モードに戻ります。

構文

```
EXECUTE(<ERROR>)
```

意味

EXECUTE:	輪郭解析を終了する予約機能	
<ERROR>:	; エラーフィードバック信号を示す変数	
	タイプ:	INT
	変数の値は、エラーが発生することなく、輪郭が解析可能であったかどうかを表わします。	
	0	異常
	1	エラーなし

例

プログラムコード

```
...
N30 CONTPRON(...)
N40 G1 X...Z...
...
N100 EXECUTE(...)
...
```

サイクルを外部的にプログラミング 20

20.1 テクノロジサイクル

20.1.1 はじめに

目次

本章では、旋削、フライス削り、および研削のサイクルについて説明します。

構成

サイクルの説明は、以下のように構成されています。

- **構文**
  転送パラメータのサイクル名と呼び出し手順です。
- **パラメータ**
  個々のパラメータを説明するためのテーブルです。

パラメータリスト

パラメータのテーブルには、名称、説明、値の範囲、および他のパラメータとの依存関係が含まれています。

この画面のパラメータの参照欄は、外部で生成されたサイクル呼び出しの再コンパイル時に、制御装置でプログラム指令された値をより簡単に特定するために提供されています。

「インタフェース専用」パラメータ

特定のパラメータはテーブル内で「インタフェース専用」の印が付いています。このようなパラメータは、サイクルの動作には関係ありません。サイクル呼び出しを完全に再コンパイルするためにだけ必要です。このパラメータがプログラム指令されていなくても、サイクルは再コンパイルが可能です。このフィールドは色で識別され、画面内で完結します。

「予備」パラメータ

「予備」として記述されているパラメータは、以下の呼び出しパラメータが内部サイクルパラメータと一致するように値 0 またはカンマを使ってプログラム指令してください。
例外:値""またはカンマを使用した文字列パラメータ

位置決めパターンでのサイクルの繰り返し

穴あけサイクルとミリングサイクルは、位置決めパターン(モーダル呼び出し)で繰り返すことができます。このような場合、同一の行で MCALL をサイクルの前に書き込んでください(例:MCALL CYCLE83(...))。

注記

特定の転送パラメータ(<_VARI>, <_GMODE>, <_DMODE>, <_AMODE>など)がパラメータとして間接的にプログラム指令されている場合は、再コンパイル時に入力画面が開かれますが、定義された選択欄への一義的な割り当てがないため、保存できません。

20.1.2 テクノロジ別の概要

以下の一覧表に、使用可能な外部でプログラム可能なすべてのテクノロジサイクルと、それぞれのサイクルに割り当てられたテクノロジをリストします。

機能	テクノロジサイクル
穴あけ	• CYCLE81 - 穴あけ、センタリング (ページ 895) • CYCLE82 - 穴あけ、座ぐり (ページ 896) • CYCLE85 - リーマ加工 (ページ 909) • CYCLE86 - ボーリング (ページ 910) • CYCLE83 - 深穴あけ (ページ 899) • CYCLE830 - 深穴ドリル 2 (ページ 943) • CYCLE84 - フローティングチャックなしのタッピング (ページ 903) • CYCLE840 - フローティングチャックによるタッピング (ページ 955) • CYCLE78 - ドリルねじフライス削り (ページ 888) • CYCLE802 - 任意の位置 (ページ 939) • HOLES1 - 直線位置決めパターン (ページ 841) • CYCLE801 - 格子またはフレーム (ページ 937) • HOLES2 - 円弧位置決めパターン (ページ 842)
旋削	• CYCLE951 - 荒削り (ページ 969) • CYCLE930 - 溝削り (ページ 962) • CYCLE940 - アンダーカット形状 (ページ 965) • CYCLE99 - ねじの旋削 (ページ 923) • CYCLE98 - 連続ねじ (ページ 917) • CYCLE92 - 突切り (ページ 912)
輪郭旋削	• CYCLE62 - 輪郭の呼び出し (ページ 869) • CYCLE952 - 輪郭の溝削り (ページ 973)

機能	テクノロジサイクル
フライス加工	• CYCLE61 - 正面削り (ページ 866) • POCKET3 - 長方形ポケットのフライス削り (ページ 845) • POCKET4 - 円形ポケットのフライス削り (ページ 849) • CYCLE76 - 長方形スピゴットのフライス削り (ページ 882) • CYCLE77 - 円形スピゴットのフライス削り (ページ 885) • CYCLE79 - 多角形 (ページ 892) • SLOT1- 直線溝 (ページ 853) • SLOT2 - 円周溝 (ページ 857) • CYCLE899 - オープン溝のフライス削り (ページ 959) • LONGHOLE - 長穴 (ページ 860) • CYCLE70 - ねじフライス削り (ページ 876) • CYCLE60 - 彫刻サイクル (ページ 863)
輪郭切削	• CYCLE62 - 輪郭の呼び出し (ページ 869) • CYCLE72 - 輪郭フライス削り (ページ 878) • CYCLE63 - 輪郭ポケットのフライス削り (ページ 870) • CYCLE64 - 輪郭ポケットの前加工 (ページ 874)
グラインダー	• CYCLE495 - 成形研削 (ページ 929) • CYCLE435 - ドレッサー座標系の設定 (ページ 928) • CYCLE4071 - 反転点での切り込みによる長手方向の研削 (ページ 980) • CYCLE4072 - 反転点での切り込みによる長手方向の研削とキャンセル信号 (ページ 982) • CYCLE4073 - 連続切り込みによる長手方向の研削 (ページ 985) • CYCLE4074 - 連続切り込みによる長手方向の研削とキャンセル信号 (ページ 987) • CYCLE4075 - 反転点での切り込みによる平面研削 (ページ 991) • CYCLE4077 - 反転点での切り込みによる平面研削とキャンセル信号 (ページ 993) • CYCLE4078 - 連続切り込みによる平面研削 (ページ 997) • CYCLE4079 - 間欠切り込みによる平面研削 (ページ 999)

機能	テクノロジサイクル
他の	• CYCLE800 - 旋回 (ページ 931) • CYCLE832 - 高速設定 (ページ 951)
すべて	• GROUP_BEGIN - プログラムブロック開始 (ページ 1001) • GROUP_END - プログラムブロック終了 (ページ 1002) • GROUP_ADDEND - 試し切り追加終了 (ページ 1003)

20.1.3 HOLES1 - 直線位置決めパターン

構文

```
HOLES1(<SPCA>, <SPCO>, <STA1>, <FDIS>, <DBH>, <NUM>, <_VARI>,
<_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	X0	<SPCA>	REAL	1 番目の軸の直線位置決めパターンの基準点(abs)
2	Y0	<SPCO>	REAL	2 番目の軸の直線位置決めパターンの基準点(abs)
3	α0	<STA1>	REAL	基本回転角度(1 番目の軸に対する角度)
4	L0	<FDIS>	REAL	1 番目の穴から基準点までの距離
5	L	<DBH>	REAL	穴間の間隔
6	n	<NUM>	INT	穴の数
7		<_VARI>	INT	予備
8		<_UMODE>	INT	予備
9		<_HIDE>	STRING [200]	非表示位置 • 最大 198 文字 • 連続する位置番号の指定。例:「1,3」(位置 1 と 3 は実行されません)
10		<_NSP>	INT	予備

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
11		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/18/19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)

20.1.4 HOLES2 - 円弧位置決めパターン

構文

```
HOLES2(<CPA>, <CPO>, <RAD>, <STA1>, <INDA>, <NUM>, <_VARI>,
<_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)
```

パラメータ

No.	マスクパラメータ	パラメータ 内部	データタイプ	意味	
1	X0	<CPA>	REAL	1 番目の軸の円弧位置決めパターンの中心点(abs) 1 番目の軸の基準点	(XY の場合) (XA、YB、ZC の場合)
2	Y0	<CPO>	REAL	2 番目の軸の円弧位置決めパターンの中心点(abs) 2 番目の軸の基準点	(XY の場合) (XA、YB、ZC の場合)
3	R	<RAD>	REAL	円弧位置決めパターンの半径	(XY の場合)
4	α0	<STA1>	REAL	開始角度 または 1 番目の回転軸の位置	(XY の場合) (XA、YB、ZC の場合)

No.	マスクパラメータ	パラメータ 内部	データタイプ	意味		
5	α1	<INDA>	REAL	進み角度 - (円弧のみ)		(XY、XA、YB、ZC の場合)
					< 0 =	右回り
					> 0 =	反時計回り
6	n	<NUM>	INT	位置の数		

No.	マスクパラメータ	パラメータ 内部	データタイプ	意味		
7		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	予備	
				TENS(十の位):	位置決めタイプ	
					0 =	アプローチ位置 - 直線
					1 =	アプローチ位置 - 円弧軌跡
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):	円弧パターン	
					0 =	互換性モード、INDA = の場合は一周円、INDA <> 0 の場合は円弧
					1 =	一周円
					2 =	円弧
				TEN THOUSANDS(万の位):	回転軸ありの位置決めパターン	
					0 =	XY (回転軸なし) (XY の場合)
					1 =	XA (X 軸と X を中心とした回転軸) (XA の場合のみ)
					2 =	YB (Y 軸と Y を中心とした回転軸) (YB の場合のみ)
					3 =	ZC (Z 軸と C を中心とした回転軸) (ZC の場合のみ)
				TEN MILLIONS + ONE MILLION(百万+ 十万の位):	オフセット(同じ軸を中心とした複数の回転軸用、インデックスが大きすぎる場合は、1 番目の軸)	
					00 =	1 番目の A、B、または C 軸
					01 =	2 番目の A、B、または C 軸
					...	
					10 =	20 番目の A、B、または C 軸
8		<_UMODE>	INT	予備		

No.	マスクパラメータ	パラメータ 内部	データタイプ	意味		
9		<_HIDE>	STRING [200]	予備		
10		<_NSP>	INT	予備		
11		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/18/19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)

20.1.5 POCKET3 - 長方形ポケットのフライス削り

構文

```
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>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<_RTP>	REAL	イニシャル点(abs)
2	Z0	<_RFP>	REAL	工具軸の基準点(abs)
3	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<_DP>	REAL	ポケットの深さ(abs/inc)、<_AMODE>を参照
5	L	<_LENG>	REAL	ポケットの長さ(inc、符号なしで入力)
6	W	<_WID>	REAL	ポケットの幅(inc、符号なしで入力)
7	R	<_CRAD>	REAL	ポケットのコーナ半径
8	X0	<_PA>	REAL	基準点、1 番目の軸(abs)

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
9	YO	<_PO>	REAL	基準点、2 番目の軸(abs)		
10	α0	<_STA>	REAL	回転角度、長手軸(L)と 1 番目の軸との間の角度		
11	DZ	<_MID>	REAL	最大切り込み深さ		
12	UXY	<_FAL>	REAL	平面の仕上げ代		
13	UZ	<_FALD>	REAL	深さの仕上げ代		
14	F	<_FFP1>	REAL	平面の送り速度		
15	FZ	<_FFD>	REAL	深さ切り込み速度		
16		<_CDIR>	INT	加工方向	0 =	ダウンカット
					1 =	アップカット
17		<_VARI>	INT	加工タイプ		
				UNITS(一の位):		
					1 =	荒削り
					2 =	仕上げ
					4 =	端面の仕上げ
					5 =	面取り
				TENS(十の位):		
					0 =	予備穴あけ、G0 による切り込み
					1 =	縦方向、G1 による切り込み
					2 =	ヘリカル
					3 =	ポケットの長手軸の揺動
				HUNDREDS(百の位):	予備	
18	DXY	<_MIDA>	REAL	最大平面切り込み、単位については、<_AMODE>を参照		
19	L1	<_AP1>	REAL	予備加工の長さ(inc)		
20	W1	<_AP2>	REAL	予備加工の幅(inc)		
21	AZ	<_AD>	REAL	予備加工の深さ(inc)		
22	ER	<_RAD1>	REAL	ヘリカル切込みのヘリカル軌跡の半径		
	EW			揺動時の最大切り込み角度		
23	EP	<_DP1>	REAL	ヘリカル切込み時のヘリカルピッチ		
24		<_UMODE>	INT	予備		
25	FS	<_FS>	REAL	面取り幅(inc)		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
26	ZFS	<_ZFS>	REAL	面取りの切り込み深さ(工具先端) (abs/inc)、<_AMODE>を参照		
27		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	予備	
				TENS(十の位):	予備	
				HUNDREDS(百の位):	加工または開始点の計算のみのどちらかを選択します。	
					0 =	互換性モード
					1 =	通常の加工
				THOUSANDS(千の位):	中心/コーナによる位置指令方法	
					0 =	互換性モード
					1 =	中心による位置指令方法
					2 =	コーナ点による位置指令方法、ポケット位置 +LENG/+WID
					3 =	コーナ点による位置指令方法、ポケット位置 -LENG/+WID
					4 =	コーナ点による位置指令方法、ポケット位置 +LENG/-WID
					5 =	コーナ点による位置指令方法、ポケット位置 -LENG/-WID
				TEN THOUSANDS(万の位):	全て加工/再加工	
					0 =	互換性モード(<_AP1>, <_AP2>と<_AD> を以前と同様に処理)
					1 =	全て加工
					2 =	後加工

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
28		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	送り速度タイプ:面と深さの送り速度の G グループ(G94/G95)	
					0 =	互換性モード
					1 =	サイクル呼び出し前と同じ G 命令。面と深さの送り速度と同じ G94/G95
				HUNDREDS(百の位):	---	予備
				THOUSANDS(千の位):	---	予備
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1004)	
					0 =	入力:全て
					1 =	入力:簡易
29		<_AMODE>	INT	代替モード		
				UNITS(一の位):	ポケットの深さ(Z1)	
					0 =	アブソリュート(互換性モード)
					1 =	インクレメンタル
				TENS(十の位):	平面切り込みの単位(DXY)	
					0 =	mm
					1 =	工具直径の%
				HUNDREDS(百の位):	面取りの切り込み深さ(ZFS)	
					0 =	アブソリュート
					1 =	インクレメンタル

20.1.6 POCKET4 - 円形ポケットのフライス削り

構文

```
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>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1	RP	<_RTP>	REAL	イニシャル点(abs)		
2	Z0	<_RFP>	REAL	工具軸の基準点(abs)		
3	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)		
4	Z1	<_DP>	REAL	ポケットの深さ(abs/inc)、<_AMODE>を参照		
5	∅	<_CDIAM>	REAL	ポケットの直径または半径、 <_DMODE>を参照		
6	X0	<_PA>	REAL	基準点、1 番目の軸(abs)		
7	Y0	<_PO>	REAL	基準点、2 番目の軸(abs)		
8	DZ	<_MID>	REAL	最大深さ設定、<_VARI> = 平面毎を参照 最大ヘリカル設定、 <_VARI> = ヘリカルを参照		
9	UXY	<_FAL>	REAL	平面の仕上げ代		
10	UZ	<_FALD>	REAL	深さの仕上げ代		
11	F	<_FFP1>	REAL	面加工時の送り速度		
12	FZ	<_FFD>	REAL	深さ切り込み速度		
13		<_CDIR>	INT	加工方向	0 =	ダウンカット
					1 =	アップカット

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
14		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	加工	
					1 =	荒削り
					2 =	仕上げ
					4 =	端面の仕上げ
					5 =	面取り
				TENS(十の位):	切り込みタイプ(荒削りと仕上げ)	
					0 =	予備穴あけ、G0 による切り込み(ポケットは前加工済み)
					1 =	縦方向、G1 による切り込み
					2 =	ヘリカル
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):		
					0 =	平面毎
					1 =	ヘリカル
15	DXY	<_MIDA>	REAL	最大平面切り込み、<_AMODE>を参照、 0 = 0.8 x 工具直径		
16	∅	<_AP1>	REAL	予備加工の直径/半径(inc)		
17	AZ	<_AD>	REAL	予備加工の深さ(inc)		
18	ER	<_RAD1>	REAL	ヘリカル切込みのヘリカル軌跡の半径		
19	EP	<_DP1>	REAL	ヘリカル軌跡上の切り込みのヘリカルピッチ		
20		<_UMODE>	INT	予備		
21	FS	<_FS>	REAL	面取り幅(inc)		
22	ZFS	<_ZFS>	REAL	面取りの切り込み深さ(工具先端) (abs/inc)、 <_AMODE>を参照		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
23		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	予備	
				TENS(十の位):	予備	
				HUNDREDS(百の位):	加工/開始点の計算	
					0 =	互換性モード
					1 =	通常の加工
				THOUSANDS(千の位):	予備	
				TEN THOUSANDS(万の位):	全て加工/再加工	
					0 =	互換性モード(<_AP1>と <_AD> を以前と同様に処理)
					1 =	全て加工
					2 =	後加工

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
24		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/18/19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	送り速度タイプ:面と深さの送り速度の G グループ(G94/G95)	
					0 =	互換性モード
					1 =	サイクル呼び出し前と同じ G 命令。面と深さの送り速度と同じ G94/G95
				HUNDREDS(百の位):		
					0 =	互換性モード(半径として<_CDIAM>/<_AP1>を入力)
					1 =	直径として<_CDIAM>/<_AP1> を入力
				THOUSANDS(千の位):	---	予備
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1004)	
					0 =	入力:全て
					1 =	入力:簡易

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
25		<_AMODE>	INT	代替モード		
				UNITS(一の位):	ポケットの深さ(Z1)	
					0 =	アブソリュート(互換性モード)
					1 =	インクレメンタル
				TENS(十の位):	切り込み幅の単位(DXY)	
					0 =	mm
					1 =	工具直径の%
				HUNDREDS(百の位):	面取りの切り込み深さ(ZFS)	
					0 =	アブソリュート
					1 =	インクレメンタル

20.1.7 SLOT1- 直線溝

構文

```
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>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<RTP>	REAL	イニシャル点(abs)
2	Z0	<RFP>	REAL	工具軸の基準点(abs)
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<_DP>	REAL	溝の深さ(abs)
5		<_DPR>	REAL	Z0 を基準にした溝の深さ(inc)(符号なしで入力)
6		<NUM>	INT	溝の数 = 1
7	L	<LENG>	REAL	溝の長さ
8	W	<WID>	REAL	溝幅

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
9	X0	<_CPA>	REAL	平面の 1 番目の軸の基準点		
10	Y0	<_CPO>	REAL	平面の 2 番目の軸の基準点		
11		<RAD>	REAL	予備		
12	α	<STA1>	REAL	回転角度		
13		<INDA>	REAL	予備		
14	FZ	<FFD>	REAL	深さ切り込み速度		
15	F	<FFP1>	REAL	送り速度		
16	DZ	<_MID>	REAL	最大切り込み深さ		
17		<CDIR>	INT	加工方向	0 =	ダウンカット
					1 =	アップカット
18	UXY	<_FAL>	REAL	平面または溝端面の仕上げ代		
19		<VARI>	INT	加工タイプ		
				UNITS(一の位):		
					0 =	予備
					1 =	荒削り
					2 =	仕上げ
					4 =	端面の仕上げ(端面のみを加工)
					5 =	面取り
				TENS(十の位):	アプローチ	
					0 =	予備穴あけ、G0 による切り込み(溝は前加工済み)
					1 =	縦方向、G1 による切り込み
					2 =	ヘリカル
					3 =	揺動
				HUNDREDS(百の位):	予備	
20	DZF	<_MIDF>	REAL	予備		
21	FF	<FFP2>	REAL	予備		
22	SF	<SSF>	REAL	予備		
23	UZ	<_FALD>	REAL	深さの仕上げ代		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
24	ER	<_STA2>	REAL	ヘリカル切り込みでのヘリカル軌跡の半径		
	EW			揺動時の最大切り込み角度		
25	EP	<_DP1>	REAL	ヘリカル時の 1 回転あたりの切り込み深さ		
26		<_UMODE>	INT	予備		
27	FS	<_FS>	REAL	面取りの面取り幅(inc)		
28	ZFS	<_ZFS>	REAL	面取りの切り込み深さ(工具先端) (abs または inc)、<_AMODE>を参照		
29		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	予備	
				TENS(十の位):	予備	
				HUNDREDS(百の位):	加工または開始点の計算のみのどちらかを選択します。	
					1 =	通常の加工
				THOUSANDS(千の位):	溝の長さの基準点の位置指令方法	
					0 =	中央
					1 =	内部左側 +L
					2 =	内部右側 -L
					3 =	左端 +L
					4 =	右端 -L

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
30		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/18/19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	予備	
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):	ソフトウェアバージョンの識別	
					1 =	機能の拡張 SLOT1
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1004)	
					0 =	入力:全て
					1 =	入力:簡易
31		<_AMODE>	INT	代替モード		
				UNITS(一の位):	最終深さ Z1 (abs/inc)	
					0 =	互換性
					1 =	Z1 (inc)
					2 =	Z1 (abs)
				TENS(十の位):	予備	
				HUNDREDS(百の位):	面取りの切り込み深さ(ZFS)	
					0 =	ZFS (abs)
					1 =	ZFS (inc)

注記

このサイクルには、以前のソフトウェアバージョンにはなかった新しい機能が備わっています。その結果、入力画面(<NUM>, <RAD>, <INDA>)の特定のパラメータが表示されなくなりました。1 つの位置決めパターンの複数の溝を、「MCALL」を使用して希望する位置決めパターン、たとえば HOLES2 などを呼び出すことでプログラム指令できます。

20.1.8 SLOT2 - 円周溝

構文

```
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>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1	RP	<RTP>	REAL	イニシャル点(abs)		
2	Z0	<RFP>	REAL	工具軸の基準点(abs)		
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)		
4	Z1	<_DP>	REAL	溝の深さ(abs)		
5		<_DPR>	REAL	Z0 を基準にした溝の深さ(inc)(符号なしで入力)		
6	N	<NUM>	INT	溝の数		
7	α1	<AFSL>	REAL	溝の開口部角度		
8	W	<WID>	REAL	溝幅		
9	X0	<_CPA>	REAL	基準点 = 円弧の中心点、平面の 1 番目の軸		
10	Y0	<_CPO>	REAL	基準点 = 円弧の中心点、平面の 2 番目の軸		
11	R	<RAD>	REAL	円弧の半径		
12	α0	<STA1>	REAL	開始角度		
13	α2	<INDA>	REAL	増分角度		
14	FZ	<FFD>	REAL	深さ切り込み速度		
15	F	<FFP1>	REAL	送り速度		
16	DZ	<_MID>	REAL	最大切り込み深さ		
17		<CDIR>	INT	加工方向	0 =	ダウンカット
					1 =	アップカット
18	UXY	<_FAL>	REAL	平面または溝端面の仕上げ代		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
19		<VARI>	INT	加工タイプ		
				UNITS(一の位):		
					0 =	全て加工
					1 =	荒削り
					2 =	仕上げ
					3 =	端面の仕上げ
					5 =	面取り
				TENS(十の位):		
					0 =	G0 ラインによる中間位置決め
					1 =	円弧軌跡上での中間位置決め
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):		
					0 =	互換性モード、 <INDA> = 0 の場合は一周円、<INDA> <> 0 の場合は円弧
					1 =	一周円
					2 =	円弧
20	DZF	<_MIDF>	REAL	予備		
21		<FFP2>	REAL	予備		
22		<SSF>	REAL	予備		
23	FF	<_FFCP>	REAL	予備		
24		<_UMODE>	INT	予備		
25	FS	<_FS>	REAL	面取り幅(inc)		
26	ZFS	<_ZFS>	REAL	面取りの切り込み深さ(工具先端) (abs または inc)、<_AMODE>を参照		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
27		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	予備	
				TENS(十の位):	予備	
				HUNDREDS(百の位):	加工または開始点の計算のみのどちらかを選択します。	
					0 =	互換性モード
					1 =	通常の加工
28		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/18/19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	予備	
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):	ソフトウェアバージョンの識別	
					1 =	ソフトウェアバージョン 2.5 以降の SLOT2 機能
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1004)	
					0 =	入力:全て
					1 =	入力:簡易

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
29		<_AMODE>	INT	代替モード		
				UNITS(一の位):	最終深さ Z1 (abs/inc)	
					0 =	互換性
					1 =	Z1 (inc)
					2 =	Z1 (abs)
				TENS(十の位):	予備	
				HUNDREDS(百の位):	面取りの切り込み深さ(ZFS)	
					0 =	ZFS (abs)
					1 =	ZFS (inc)

20.1.9 LONGHOLE - 長穴

構文

```
LONGHOLE(<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <LENG>,
<_CPA>, <_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <MID>,
<_VARI>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<RTP>	REAL	イニシャル点(abs)
2	Z0	<_RFP>	REAL	工具軸の基準点(abs)
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<_DP>	REAL	長穴の深さ(abs)
5		<_DPR>	REAL	Z0 を基準にした長穴の深さ(inc)(符号なしで入力)
6		<NUM>	INT	長穴の数 = 1
7	L	<LENG>	REAL	長穴の長さ
8	X0	<_CPA>	REAL	平面における 1 番目の軸の基準点
9	Y0	<_CPO>	REAL	平面における 2 番目の軸の基準点
10		<RAD>	REAL	予備

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
11	α0	<STA1>	REAL	回転角度		
12		<INDA>	REAL	予備		
13	FZ	<FFD>	REAL	深さ切り込み速度		
14	F	<FFP1>	REAL	送り速度		
15	DZ	<MID>	REAL	最大切り込み深さ		
16		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	切り込みタイプ	
					1 =	G1 で縦方向
					3 =	揺動
				HUNDREDS(百の位):	予備	
17		<_UMODE>	INT	予備		
18		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	予備	
				TENS(十の位):	予備	
				HUNDREDS(百の位):	加工または開始点の計算のみのどちらかを選択します。	
					0 =	互換性モード
					1 =	通常の加工
				THOUSANDS(千の位):	溝の長さの基準点の位置指令方法	
					0 =	中央
					1 =	内部左側 +L
					2 =	内部右側 -L
					3 =	左端 +L
					4 =	右端 -L

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
19		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/18/19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	送り速度タイプ:面と深さの送り速度の G グループ(G94/G95)	
					0 =	互換性モード
					1 =	サイクル呼び出し前と同じ G 命令。面と深さの送り速度と同じ G94/G95
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):	ソフトウェアバージョンの識別	
					1 =	機能の拡張 LONGHOLE (基準点の位置指令方法)
20		<_AMODE>	INT	代替モード		
				UNITS(一の位):	最終深さ Z1 (abs/inc)	
					0 =	互換性
					1 =	Z1 (inc)
					2 =	Z1 (abs)

注記

このサイクルには、以前のソフトウェアバージョンにはなかった新しい機能が備わっています。その結果、入力画面(<NUM>, <RAD>, <INDA>)の特定のパラメータが表示されなくなりました。1 つの位置決めパターンの複数の溝を、「MCALL」を使用して希望する位置決めパターン、たとえば HOLES2 などを呼び出すことでプログラム指令できます。

20.1.10 CYCLE60 - 彫刻サイクル

構文

```
CYCLE60(<_TEXT>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_PA>,
<_PO>, <_STA>, <_CP1>, <_CP2>, <_WID>, <_DF>, <_FFD>, <_FFP1>,
<_VARI>, <_CODEP>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1		<_TEXT>	STRING [200]	彫刻するテキスト(最大 100 文字)
2	RP	<_RTP>	REAL	イニシャル点(abs)
3	Z0	<_RFP>	REAL	工具軸の基準点(abs)
4	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
5	Z1	<_DP>	REAL	深さ(abs)、<_AMODE>を参照
6	Z1	<_DPR>	REAL	深さ(inc)、<_AMODE>を参照
7	X0	<_PA>	REAL	平面の 1 番目の軸の基準点(abs) - 直交座標、<_VARI>を参照
	R			基準点、長さ(半径) - 極座標、<_VARI>を参照
8	Y0	<_PO>	REAL	平面の 2 番目の軸の基準点(abs) - 直交座標、<_VARI>を参照
	α0			基準点、1 番目の軸を基準にした角度 - 極座標、<_VARI>を参照
9	α1	<_STA>	REAL	テキストの方向、1 番目の軸を基準にしたテキスト行の角度、<_VARI>を参照
10	XM	<_CP1>	REAL	テキストの円弧の中心点、平面の 1 番目の軸(abs) - 直交座標、<_VARI> を参照
	LM			テキストの円弧の中心点、WNP を基準にした長さ(半径) - 極座標、 <_VARI> を参照
11	YM	<_CP2>	REAL	テキストの円弧の中心点、平面の 2 番目の軸(abs) - 直交座標、 <_VARI>を参照
	αM			テキストの円弧の中心点、1 番目の軸を基準にした角度 - 極座標、<_VARI>を参照

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
12	W	<_WID>	REAL	文字の高さ(符号なしで入力)
13	DX1 DX2	<_DF>	REAL	文字間の距離/全体の幅、 <_VARI>を参照
	α2			開口角度、 <_VARI>を参照
14	FZ	<_FFD>	REAL	深さ切り込み速度、<_DMODE> を参照
15	F	<_FFP1>	REAL	面加工時の送り速度

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
16		<_VARI>	INT	加工(彫刻したテキストの配置と基準点)		
				UNITS(一の位):	基準点	
					0 =	直交座標
					1 =	極座標
				TENS(十の位):	テキストの配置	
					0 =	1 つのライン上に配置
					1 =	上向きの円弧に配置
					2 =	下向きの円弧に配置
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):	テキストの基準点、横方向	
					0 =	左側
					1 =	中央
					2 =	右側
				TEN THOUSANDS(万の位):	テキストの基準点、縦方向	
					0 =	底面
					1 =	中央
					2 =	上面
				HUNDRED THOUSANDS(十万の位):	テキスト長	
					0 =	文字の間隔
					1 =	テキスト全体の幅(直線テキストのみ)
					2 =	開口角度(円形テキストのみ)
				ONE MILLION(百万の位):	円弧の中心点	
					0 =	直交(直交座標)
					1 =	極座標
				TEN MILLION(千万の位):	ミラー書き込み	
					0 =	互換性
					1 =	ミラー書き込み ON
					2 =	ミラー書き込み OFF
17		<_CODEP>	INT	書き込みのためのコードページ番号(現在は 1252 のみ)		
18		<_UMODE>	INT	予備		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
19		_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	予備	
				TENS(十の位):	予備	
				HUNDREDS(百の位):	加工または開始点の計算のみのどちらかを選択します。	
					0 =	互換性モード
					1 =	通常の加工
20		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/18/19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	送り速度タイプ:面と深さの送り速度の G グループ(G94/G95)	
					0 =	互換性モード
					1 =	サイクル呼び出し前と同じ G 命令。面と深さの送り速度と同じ G94/G95
21		<_AMODE>	INT	代替モード		
				UNITS(一の位):	最終深さ(<_DP>, <_DPR>)	
					0 =	互換性
					1 =	インクレメンタル(<_DPR>)
					2 =	アブソリュート(<_DP>)

20.1.11 CYCLE61 - 正面削り

構文

```
CYCLE61(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_PA>, <_PO>, <_LENG>,
<_WID>, <_MID>, <_MIDA>, <_FALD>, <_FFP1>, <_VARI>, <_LIM>,
<_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1	RP	<_RTP>	REAL	イニシャル点(abs)		
2	Z0	<_RFP>	REAL	工具軸の基準点、素材の高さ(abs)		
3	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)		
4	Z1	<_DP>	REAL	仕上げ部の高さ(abs/inc)、<_AMODE>を参照		
5	X0	<_PA>	REAL	1 番目の軸のコーナ点 1(abs)		
6	Y0	<_PO>	REAL	2 番目の軸のコーナ点 1(abs)		
7	X1	<_LENG>	REAL	1 番目の軸のコーナ点 2(abs または inc)、<_AMODE>を参照		
8	Y1	<_WID>	REAL	2 番目の軸のコーナ点 2(abs または inc)、<_AMODE>を参照		
9	DZ	<_MID>	REAL	最大切り込み深さ		
10	DXY	<_MIDA>	REAL	最大平面切り込み(単位については、<_AMODE>を参照)		
11	UZ	<_FALD>	REAL	深さの仕上げ代		
12	F	<_FFP1>	REAL	加工送り速度		
13		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	加工タイプ	
					1 =	荒削り
					2 =	仕上げ
				TENS(十の位):	加工方向	
					1 =	1 番目の軸に平行、一方向
					2 =	2 番目の軸に平行、一方向
					3 =	1 番目の軸に平行、可変方向
					4 =	2 番目の軸に平行、可変方向

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
14		<_LIM>	INT	制限		
				UNITS(一の位):	1 番目の軸の制限値、負	
					0 =	なし
					1 =	あり
				TENS(十の位):	1 番目の軸の制限値、正	
					0 =	なし
					1 =	あり
				HUNDREDS(百の位):	2 番目の軸の制限値、負	
					0 =	なし
					1 =	あり
				THOUSANDS(千の位):	2 番目の軸の制限値、正	
					0 =	なし
					1 =	あり
15		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/18/19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
16		<_AMODE>	INT	代替モード		
				UNITS(一の位):	最終深さ(<_DP>)	
					0 =	アブソリュート
					1 =	インクレメンタル
				TENS(十の位):	平面切り込みの単位(<_MIDA>)	
					0 =	mm
					1 =	工具直径の%
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):	面の長さ	
					0 =	インクレメンタル
					1 =	アブソリュート
				TEN THOUSANDS(万の位):	面の幅	
					0 =	インクレメンタル
					1 =	アブソリュート

20.1.12 CYCLE62 - 輪郭の呼び出し

構文

```
CYCLE62(<_KNAME>,<_TYPE>, <_LAB1>, <_LAB2>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1	PRG/CON	<_KNAME>	STRING [140]	輪郭名称またはサブプログラム名称は、_TYPE = 2 でプログラム指令しないでください。		
2		<_TYPE>	INT	輪郭入力の特定		
					0 =	サブプログラム
					1 =	輪郭名称
					2 =	ラベル
					3 =	サブプログラム内のラベル

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
3	LAB1	<_LAB1>	STRING[32]	ラベル 1、輪郭の開始
4	LAB2	<_LAB2>	STRING[32]	ラベル 2、輪郭の終了

20.1.13 CYCLE63 - 輪郭ポケットのフライス削り

構文

```
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>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1	PRG	<_PRG>	STRING [100]	切削プログラムの名称		
2		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	加工プロセス	
					1 =	荒削り
					3 =	底面の仕上げ
					4 =	端面の仕上げ
					5 =	面取り
				TENS(十の位):	切り込みタイプ	
					0 =	中央切り込み
					1 =	ヘリカル切り込み
					2 =	揺動切込み
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):	戻しモード	
					0 =	イニシャル点まで戻します
					1 =	基準点 + 安全距離まで戻します
				TEN THOUSANDS(万の位):	底面の荒削りと仕上げの開始点	
					0 =	自動
					1 =	手動
3	RP	<_RP>	REAL	イニシャル点(abs)		
4	Z0	<_Z0>	REAL	工具軸の基準点(abs)		
5	SC	<_SC>	REAL	安全距離(基準点に追加、符号なしで入力)		
6	Z1	<_Z1>	REAL	最終深さ(<_AMODE> UNITS を参照)		
7	F	<_F>	REAL	荒削り/仕上げ時の平面の送り速度		
8	FZ	<_FZ>	REAL	深さ切り込み速度		
9	DXY	<_DXY>	REAL	平面切り込み - 単位(<_AMODE> TENS を参照)		
10	DZ	<_DZ>	REAL	深さ切り込み		
11	UXY	<_UXY>	REAL	平面の仕上げ代		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
12	UZ	<_UZ>	REAL	深さの仕上げ代		
13		<_CDIR>	INT	加工方向	0 =	ダウンカット
					1 =	アップカット
14	XS	<_XS>	REAL	開始点 X、アブソリュート		
15	YS	<_YS>	REAL	開始点 Y、アブソリュート		
16	ER	<_ER>	REAL	ヘリカル切込み:半径		
17	EP	<_EP>	REAL	ヘリカル切込み:ピッチ		
18	EW	<_EW>	REAL	揺動切り込み:最大切り込み角度		
19	FS	<_FS>	REAL	面取りの面取り幅(inc)		
20	ZFS	<_ZFS>	REAL	面取り時の工具先端の切り込み深さ(<_AMODE> HUNDREDS を参照)		
21	TR	<_TR>	STRING[32]	削り残し除去加工時の基準工具の名称		
22	DR	<_DR>	INT	削り残し除去加工時の基準工具の D 番号		
23		<_UMODE>	INT	予備		
24		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	予備	
				TENS(十の位):	予備	
				HUNDREDS(百の位):	加工または開始点の計算のみのどちらかを選択します。	
					0 =	通常の加工(互換性モードは不要)
					1 =	通常の加工
					2 =	予備

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
25		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	予備	
				HUNDREDS(百の位):	テクノロジモード	
					1 =	ポケット
					2 =	スピゴット
				THOUSANDS(千の位):	削り残し除去加工	
					0 =	なし
					1 =	あり
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1004)	
					0 =	入力:全て
					1 =	入力:簡易
26		<_AMODE>	INT	代替モード		
				UNITS(一の位):	最終深さ(Z1)	
					0 =	アブソリュート(互換性モード)
					1 =	インクレメンタル
				TENS(十の位):	平面切り込みの単位(DXY)	
					0 =	mm
					1 =	工具直径の%
				HUNDREDS(百の位):	面取りの切り込み深さ(ZFS)	
					0 =	アブソリュート
					1 =	インクレメンタル
				THOUSANDS(千の位):	---	予備

20.1.14 CYCLE64 - 輪郭ポケットの前加工

構文

```
CYCLE64(<_PRG>, <_VARI>, <_RP>, <_Z0>, <_SC>, <_Z1>, <_F>,
<_DXY>, <_UXY>, <_UZ>, <_CDIR>, <_TR>, <_DR>, <_UMODE>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1	PRG	<_PRG>	STRING [100]	穴あけ/センタリングプログラムの名称		
2		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	予備	
				TENS(十の位):	予備	
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):	戻しモード	
					0 =	イニシャル点まで戻します
					1 =	基準点 + 安全距離まで戻します
3	RP	<_RP>	REAL	イニシャル点(abs)		
4	Z0	<_Z0>	REAL	基準点(abs)		
5	SC	<_SC>	REAL	安全距離(基準点に追加、符号なしで入力)		
6	Z1	<_Z1>	REAL	穴あけ/センタリング深さ(<_AMODE> UNITS を参照)		
7	F	<_F>	REAL	穴あけ/センタリング送り速度		
8	DXY	<_DXY>	REAL	平面切り込み - 単位(<_AMODE> TENS を参照)		
9	UXY	<_UXY>	REAL	平面の仕上げ代		
10	UZ	<_UZ>	REAL	深さの仕上げ代		
11		<_CDIR>	INT	加工方向	0 =	ダウンカット
					1 =	アップカット
12	TR	<_TR>	STRING[20]	基準工具名称		
13	DR	<_DR>	INT	基準工具 D 番号		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
14		<_UMODE>	INT	予備		
15		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	予備	
				TENS(十の位):	予備	
				HUNDREDS(百の位):	加工または開始点の計算のみのどちらかを選択します。	
					0 =	通常の加工(互換性モードは不要)
					1 =	通常の加工
					2 =	予備
25		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	テクノロジモード	
					1 =	予備穴あけ
					2 =	センタリング
26		<_AMODE>	INT	代替モード		
				UNITS(一の位):	穴あけ/センタリングの深さ Z1	
					0 =	アブソリュート(互換性モード)
					1 =	インクレメンタル
				TENS(十の位):	平面切り込みの単位(DXY)	
					0 =	mm
					1 =	工具直径の%

20.1.15 CYCLE70 - ねじフライス削り

構文

```
CYCLE70(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DIATH>, <_H1>, <_FAL>,
<_PIT>, <_NT>, <_MID>, <_FFR>, <_TYPTH>, <_PA>, <_PO>, <_NSP>,
<_VARI>, <_PITA>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1	RP	<_RTP>	REAL	イニシャル点(abs)		
2	Z0	<_RFP>	REAL	工具軸の基準点(abs)		
3	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)		
4	Z1	<_DP>	REAL	ねじの長さ(abs、inc)、<_AMODE>を参照 穴の底面でのねじ切り上げを考慮します(最低でも半ピッチ)。		
5	∅	<_DIATH>	REAL	ねじの公称直径		
6	H1	<_H1>	REAL	ねじ山深さ		
7	U	<_FAL>	REAL	仕上げ代		
8	P	<_PIT>	REAL	ピッチ(<_PITA>を選択:mm、inch、MODULE、ねじ山数/inch)		
9	NT	<_NT>	INT	工具先端の刃数 工具長は常に一番下の刃を基準にしています。		
10	DXY	<_MID>	REAL	一回切削当たりの最大切り込み <_MID> > <_H1>:一回切削ですべて		
11	F	<_FFR>	REAL	フライス削り送り速度		
12		<_TYPTH>	INT	ねじタイプ	0 =	めねじ
					1 =	おねじ
13	X0	<_PA>	REAL	1 番目の軸の円弧の中心点(abs)		
14	Y0	<_PO>	REAL	2 番目の軸の円弧の中心点(abs)		
15	αS	<_NSP>	REAL	開始角度(多条ねじ)		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
16		<_VARI>	INT	加工タイプ		
				UNITS(一の位):		
					1 =	荒削り
					2 =	仕上げ
				TENS(十の位):		
					1 =	上から下へ
					2 =	下から上へ
				HUNDREDS(百の位):		
					0 =	右ねじ
					1 =	左ねじ
17		<_PITA>	INT	ねじピッチの評価		
					0 =	互換性モード
					1 =	ピッチ(mm)
					2 =	ピッチ(ねじ山数/inch)(TPI)
					3 =	ピッチ(inch)
					4 =	MODULE で表したピッチ
18		<_PITM>	STRING[15]	ピッチ入力のマーカとなる文字列(インタフェースのみ)		
19		<_PTAB>	STRING[20]	ねじテーブルの文字列("、「ISO」、「BSW」、「BSP」、「UNC」) (インタフェースのみ)		
20		<_PTABA>	STRING[20]	ねじテーブルからの選択のための文字列(例:「M 10」、「M 12」、...) (インタフェースのみ)		
21		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	予備	
				TENS(十の位):	予備	
				HUNDREDS(百の位):	加工/開始点の計算	
					0 =	互換性モード
					1 =	通常の加工

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
22		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
23		<_AMODE>	INT	代替モード		
				UNITS(一の位):	ねじ長さ(<_DP>)	
					0 =	アブソリュート
					1 =	インクレメンタル

20.1.16 CYCLE72 - 輪郭フライス削り

構文

```
CYCLE72(<_KNAME>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_MID>, <_FAL>,
<_FALD>, <_FFP1>, <_FFD>, <_VARI>, <_RL>, <_AS1>, <_LP1>, <_FF3>,
<_AS2>, <_LP2>,<_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1		<_KNAME>	STRING [141]	輪郭サブプログラムの名称
2	RP	<_RTP>	REAL	イニシャル点(abs)
3	Z0	<_RFP>	REAL	工具軸の基準点(abs)
4	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
5	Z1	<_DP>	REAL	終点、最終深さ(abs/inc)、<_AMODE>を参照
6	DZ	<_MID>	REAL	最大切り込み深さ(inc; 符号なしで入力)
7	UXY	<_FAL>	REAL	平面の仕上げ代(inc)、輪郭端面の削り代

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
8	UZ	<_FALD>	REAL	仕上げ代の深さ(inc)、底面の削り代(符号なしで入力)		
9	FX	<_FFP1>	REAL	輪郭の送り速度		
10	FZ	<_FFD>	REAL	深さ切り込み(または空間切り込み)の送り速度		
11		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	加工タイプ	
					1 =	荒削り
					2 =	仕上げ
					5 =	面取り
				TENS(十の位):		
					0 =	G0 による中間軌跡
					1 =	G1 による中間軌跡
				HUNDREDS(百の位):	輪郭の最後で後退	
					0 =	輪郭の最後で基準点まで後退
					1 =	輪郭の最後で基準点 + <_SDIS>まで後退
					2 =	輪郭の最後で<_SDIS>だけ後退
					3 =	輪郭の最後で後退せず、輪郭送り速度で次の開始点まで移動
				THOUSANDS(千の位):	予備	
				TEN THOUSANDS(万の位):	輪郭を加工	
					0 =	輪郭を前方に加工
					1 =	輪郭を後方に加工 後方に加工時の制約: • 最大で 170 個の輪郭要素(面取りまたは丸み付けを含む)。 • (X/Y)と F 平面の値だけが評価されます。

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
12		<_RL>	INT	加工方向		
					40 =	輪郭の中心(G40、アプローチと後退:直線または垂直)
					41 =	輪郭の左側(G41、アプローチと後退:直線または円弧)
					42 =	輪郭の右側(G42、アプローチと後退:直線または円弧)
13		<_AS1>	INT	輪郭のアプローチ動作		
				UNITS(一の位):		
					1 =	直線
					2 =	四分円
					3 =	半円
					4 =	垂直方向にアプローチおよび後退
				TENS(十の位):		
					0 =	平面での最後の動作
					1 =	空間での最後の動作
14	L1	<_LP1>	REAL	移動軌跡または移動半径(inc、符号なしで入力)		
15	FZ	<_FF3>	REAL	中間軌跡の送り速度(輪郭の場合と同様に G94/G95)		
16		<_AS2>	INT	輪郭の移動動作(垂直移動/後退以外)		
				UNITS(一の位):		
					1 =	直線
					2 =	四分円
					3 =	半円
				TENS(十の位):		
					0 =	平面での最後の動作
					1 =	空間での最後の動作
17	L2	<_LP2>	REAL	後退軌跡または後退半径(inc、符号なしで入力)		
18		<_UMODE>	INT	予備		
19	FS	<_FS>	REAL	面取り幅(inc)		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
20	ZFS	<_ZFS>	REAL	面取りの切り込み深さ(工具先端) (abs または inc)、<_AMODE>を参照		
21		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	予備	
				TENS(十の位):	予備	
				HUNDREDS(百の位):	加工または開始点の計算のみのどちらかを選択します。	
					0 =	互換性モード
					1 =	通常の加工
22		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/18/19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	送り速度タイプ:面と深さの送り速度の G グループ(G94/G95)	
					0 =	互換性モード
					1 =	サイクル呼び出し前と同じ G 命令。面と深さの送り速度と同じ G94/G95
				THOUSANDS(千の位):		
					0 =	互換性モード:輪郭名称は <_KNAME>にあります。
					1 =	輪郭名称は CYCLE62 でプログラム指令され、_SC_CONT_NAME に転送されます。

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
23		<_AMODE>	INT	代替モード		
				UNITS(一の位):	終点 Z1 (<_DP>)	
					0 =	アブソリュート(互換性モード)
					1 =	インクレメンタル
				TENS(十の位):	平面切り込みの単位	
					0 =	mm、inch
					1 =	予備
				HUNDREDS(百の位):	面取りの切り込み深さ(<_ZFS>)	
					0 =	アブソリュート
					1 =	インクレメンタル

注記

次の転送パラメータが間接的に(パラメータとして)プログラム指令されている場合、入力画面はリセットされません。

<_VARI>, <_RL>, <_AS1>, <_AS2>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>

20.1.17 CYCLE76 - 長方形スピゴットのフライス削り

構文

```
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>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<_RTP>	REAL	イニシャル点(abs)
2	Z0	<_RFP>	REAL	工具軸の基準点(abs)

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
3	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)		
4	Z1	<_DP>	REAL	スピゴットの深さ(abs)		
5		<_DPR>	REAL	Z0 を基準にしたスピゴットの深さ(inc)(符号なしで入力)		
6	L	<_LENG>	REAL	スピゴットの長さ、<_GMODE>を参照(符号なしで入力)		
7	W	<_WID>	REAL	スピゴットの幅、<_GMODE>を参照(符号なしで入力)		
8	R	<_CRAD>	REAL	スピゴットのコーナ半径(符号なしで入力)		
9	X0	<_PA>	REAL	平面の 1 番目の軸のスピゴットの基準点(abs)		
10	Y0	<_PO>	REAL	平面の 2 番目の軸のスピゴットの基準点(abs)		
11	α0	<_STA>	REAL	回転角度、長手軸(L)と平面の 1 番目の軸との間の角度		
12	DZ	<_MID>	REAL	最大切り込み深さ(inc; 符号なしで入力)		
13	UXY	<_FAL>	REAL	平面の仕上げ代(inc)、輪郭端面の削り代		
14	UZ	<_FALD>	REAL	仕上げ代の深さ(inc)、底面の削り代(符号なしで入力)		
15	FX	<_FFP1>	REAL	輪郭の送り速度		
16	FZ	<_FFD>	REAL	深さ切り込み速度		
17		<_CDIR>	INT	加工方向(符号なしで入力)		
				UNITS(一の位):		
					0 =	ダウンカット
					1 =	アップカット
18		<_VARI>	INT	加工タイプ		
				UNITS(一の位):		
					1 =	荒削り
					2 =	仕上げ
					5 =	面取り
19	L1	<_AP1>	REAL	スピゴット素材の長さ		
20	W1	<_AP2>	REAL	スピゴット素材の幅		
21	FS	<_FS>	REAL	面取り幅(inc)		
22	ZFS	<_ZFS>	REAL	面取りの切り込み深さ(工具先端) (abs または inc)、<_AMODE>を参照		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
23		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	予備	
				TENS(十の位):	予備	
				HUNDREDS(百の位):	加工または開始点の計算のみのどちらかを選択します。	
					0 =	互換性モード
					1 =	通常の加工
				THOUSANDS(千の位):	中心またはコーナ基準のスピゴットの位置指令方法	
					0 =	互換性モード
					1 =	中心による位置指令方法
					2 =	コーナ点の位置指令方法、スピゴット +L +W
					3 =	コーナ点の位置指令方法、スピゴット -L +W
					4 =	コーナ点の位置指令方法、スピゴット +L -W
					5 =	コーナ点の位置指令方法、スピゴット -L -W
				TEN THOUSANDS(万の位):	全て加工または再加工	
					0 =	互換性モード
					1 =	全て加工
					2 =	後加工

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
24		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/18/19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	---	予備
				HUNDREDS(百の位):	---	予備
				THOUSANDS(千の位):	---	予備
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1004)	
					0 =	入力:全て
					1 =	入力:簡易
25		<_AMODE>	INT	代替モード		
				UNITS(一の位):	最終深さ Z1 (DP)	
					0 =	互換性
					1 =	インクレメンタル
					2 =	アブソリュート
				TENS(十の位):	予備	
				HUNDREDS(百の位):	面取りの切り込み深さ(ZFS)	
					0 =	アブソリュート
					1 =	インクレメンタル

20.1.18 CYCLE77 - 円形スピゴットのフライス削り

構文

```
CYCLE77(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_CDIAM>, <_PA>,
<_PO>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>,
<_VARI>, <_AP1>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1	RP	<_RTP>	REAL	イニシャル点(abs)		
2	Z0	<_RFP>	REAL	工具軸の基準点(abs)		
3	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)		
4	Z1	<_DP>	REAL	スピゴットの深さ(abs)		
5		<_DPR>	REAL	Z0 を基準にしたスピゴットの深さ(inc)(符号なしで入力)		
6	∅	<_CDIAM>	REAL	スピゴットの直径(符号なしで入力)		
7	X0	<_PA>	REAL	平面の 1 番目の軸のスピゴットの基準点(abs)		
8	Y0	<_PO>	REAL	平面の 2 番目の軸のスピゴットの基準点(abs)		
9	DZ	<_MID>	REAL	最大切り込み深さ(inc; 符号なしで入力)		
10	UXY	<_FAL>	REAL	平面の仕上げ代(inc)、輪郭端面の削り代		
11	UZ	<_FALD>	REAL	仕上げ代の深さ(inc)、底面の削り代(符号なしで入力)		
12	FX	<_FFP1>	REAL	輪郭の送り速度		
13	FZ	<_FFD>	REAL	深さ切り込み速度		
14		<_CDIR>	INT	加工方向(符号なしで入力)		
				UNITS(一の位):		
					0 =	ダウンカット
					1 =	アップカット
15		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	加工タイプ	
					1 =	最終加工代まで荒削り
					2 =	仕上げ(仕上げ代 X/Y/Z=0)
					5 =	面取り
16	∅1	<_AP1>	REAL	スピゴット素材の直径		
17	FS	<_FS>	REAL	面取り幅(inc)		
18	ZFS	<_ZFS>	REAL	面取りの切り込み深さ(工具先端) (abs または inc)、<_AMODE>を参照		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
19		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	予備	
				TENS(十の位):	予備	
				HUNDREDS(百の位):	加工または開始点の計算のみのどちらかを選択します。	
					0 =	互換性モード
					1 =	通常の加工
				THOUSANDS(千の位):	予備	
				TEN THOUSANDS(万の位):	全て加工/再加工	
					0 =	互換性モード(以前と同様に<_AP1>を処理)
					1 =	全て加工
					2 =	後加工
20		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/18/19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	---	予備
				HUNDREDS(百の位):	---	予備
				THOUSANDS(千の位):	---	予備
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1004)	
					0 =	入力:全て
					1 =	入力:簡易

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
21		<_AMODE>	INT	代替モード		
				UNITS(一の位):	最終深さ Z1 (DP)	
					0 =	アブソリュート(互換性モード)
					1 =	インクレメンタル
					2 =	アブソリュート
				TENS(十の位):	予備	
				HUNDREDS(百の位):	面取りの切り込み深さ(ZFS)	
					0 =	アブソリュート
					1 =	インクレメンタル

20.1.19 CYCLE78 - ドリルねじフライス削り

構文

```
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>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<_RTP>	REAL	イニシャル点(abs)
2	Z0	<_RFP>	REAL	工具軸の基準点(abs)
3	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<_DP>	REAL	最終穴あけ深さ(abs/inc)、<_AMODE>を参照
5		<_ADPR>	REAL	減速された穴あけ送り速度の予備穴あけ深さ(inc)、<_VARI> TEN THOUSANDS(万の位)で有効

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
6	D	<_FDPR>	REAL	最大切り込み深さ(inc) D ≥ Z1 ⇒ 最終穴あけ深さまで一回で切り込み D < Z1 ⇒ 複数回の切り込みによる深穴サイクルと切り屑除去		
7	ZR	<_LDPR>	REAL	FR 送り速度での貫通穴あけ時の残りの穴あけ深さ(inc)		
8	∅	<_DIAM>	REAL	ねじの公称直径		
9	P	<_PIT>	REAL	数値で表したピッチ		
10		<_PITA>	INT	ねじピッチ P の評価		
					1 =	ピッチ(mm/rev)
					2 =	ピッチ(ねじ山数/inch)
					3 =	ピッチ(inch/rev)
					4 =	MODULE で表したピッチ
11	DF	<_DAM>	REAL	各々の追加切り込みの絶対値/パーセント値(逓減量)、<_AMODE>を参照		
12	V1	<_MDEP>	REAL	最小切り込み(inc)、逓減パーセンテージ値の場合のみ有効		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
13		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	予備	
				TENS(十の位):	ねじ切り加工前の切り屑除去	
					0 =	ねじフライス削り前の切り屑除去なし(最終穴あけ深さでのみ有効)
					1 =	ねじフライス削り前の切り屑除去(最終穴あけ深さでのみ有効)
				HUNDREDS(百の位):	右/左ねじ	
					0 =	右ねじ
					1 =	左ねじ
				THOUSANDS(千の位):	穴あけ送り速度での残りの穴あけ深さ	
					0 =	穴あけ送り速度 FR での残りの穴あけ深さなし
					1 =	穴あけ送り速度 FR での残りの穴あけ深さ
				TEN THOUSANDS(万の位):	減速された送り速度での予備穴あけ	
					0 =	減速された送り速度での予備穴あけなし
					1 =	減速された送り速度での予備穴あけ 予備穴あけ送り速度 = 0.3 F1(F1 < 0.15 mm/rev の場合) 予備穴あけ送り速度 = 0.1 mm/rev(F1 ≥ 0.15 mm/rev の場合)

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
14		<_CDIR>	INT	加工方向	0 =	ダウンカット
					1 =	アップカット
					4 =	アップカット + ダウンカット(荒削りと仕上げの組み合わせ)
15	Z2	<_GE>	REAL	ねじフライス削り前の後退距離(inc)		
16	F1	<_FFD>	REAL	穴あけ送り速度(mm/min、in/min、または mm/rev)		
17	FR	<_FRDP>	REAL	残りの穴あけ深さに対応した穴あけ送り速度(mm/min または mm/rev)		
18	F2	<_FFR>	REAL	ねじフライス削り時の送り速度(mm/min または mm/tooth)		
19	FS	<_FFP2>	REAL	<_CDIR> =4 の仕上げ送り速度(mm/min または mm/刃)		
20		<_FFA>	INT	送り速度の評価		
				UNITS(一の位):	穴あけ送り速度 F1	
				TENS(十の位):	残りの穴あけ深さに対応した穴あけ送り速度 FR	
				HUNDREDS(百の位):	ねじフライス削り時の送り速度 F2	
				THOUSANDS(千の位):	仕上げ送り速度 FS	
21		<_PITM>	STRING[15]	ピッチ入力のマーカとなる文字列(インタフェースのみ)[1)]		
22		<_PTAB>	STRING[20]	ねじテーブルの文字列(""、「ISO」、「BSW」、「BSP」、「UNC」)(インタフェースのみ)[1)]		
23		<_PTABA>	STRING[20]	ねじテーブルからの選択のための文字列(例:「M 10」、「M 12」、...) (インタフェースのみ)[1)]		
24		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)、予約済み		
25		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/18/19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
26		<_AMODE>	INT	代替モード		
				UNITS(一の位):	穴あけ深さ = 最終穴あけ深さ Z1 (abs/inc)	
					0 =	アブソリュート
					1 =	インクレメンタル
				TENS(十の位):	各々の追加切り込みの絶対値/パーセント値(逓減量)	
					0 =	絶対値
					1 =	パーセント値(0.001~100%)

注記

[1] パラメータ 21、22、23 は、入力画面ねじテーブルでのねじの選択にのみ使用されます。ねじテーブルは、サイクル運転時にサイクル定義を介してアクセスすることはできません。

20.1.20 CYCLE79 - 多角形

構文

```
CYCLE79(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_NUM>, <_SWL>, <_PA>,
<_PO>, <_STA>, <_RC>, <_AP1>, <_MIDA>, <_MID>, <_FAL>, <_FALD>,
<_FFP1>, <_CDIR>, <_VARI>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<_RTP>	REAL	イニシャル点(abs)
2	Z0	<_RFP>	REAL	工具軸の基準点(abs)
3	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<_DP>	REAL	多角形の深さ(abs/inc)、<_AMODE>を参照
5	N	<_NUM>	INT	端面の数(1...n)

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
6	SW/L	<_SWL>	REAL	二面幅または辺の長さ(<_VARI>による) (「SW」は二面幅、「L」は辺の長さ) 偶数の辺と単数の辺の場合は、二面幅のみ		
7	X0	<_PA>	REAL	スピゴットの基準点、1 番目の軸(abs)		
8	Y0	<_PO>	REAL	スピゴットの基準点、2 番目の軸(abs)		
9	α0	<_STA>	REAL	回転角度、1 番目の軸(X 軸)に対する端面の中心		
10	R1/FS1	<_RC>	REAL	<_NUM>によるコーナ丸み付け > 2 (半径/面取り、<_AMODE>を参照) (inc、符号なしで入力) (「R1」は丸み付け、「FS1」は面取り)		
11	∅	<_AP1>	REAL	スピゴット素材の直径		
12	DXY	<_MIDA>	REAL	最大切り込み幅(単位については、<_AMODE>を参照)		
13	DZ	<_MID>	REAL	最大切り込み深さ		
14	UXY	<_FAL>	REAL	平面の仕上げ代		
15	UZ	<_FALD>	REAL	深さの仕上げ代		
16	F	<_FFP1>	REAL	加工送り速度		
17		<_CDIR>	INT	加工方向	0 =	ダウンカット
					1 =	アップカット
18		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	加工タイプ	
					1 =	荒削り
					2 =	仕上げ
					3 =	端面の仕上げ
					5 =	面取り
				TENS(十の位):	二面幅または辺の長さ	
					0 =	二面幅
					1 =	辺の長さ
19	FS	<_FS>	REAL	面取り幅(inc)		
20	ZFS	<_ZFS>	REAL	面取りの切り込み深さ(工具先端) (abs または inc)、<_AMODE>を参照		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
21		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	予備	
				TENS(十の位):	予備	
				HUNDREDS(百の位):	加工または開始点の計算のみのどちらかを選択します。	
					1 =	通常の加工
22		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/18/19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	---	予備
				HUNDREDS(百の位):	---	予備
				THOUSANDS(千の位):	---	予備
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1004)	
					0 =	入力:全て
					1 =	入力:簡易

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
23		<_AMODE>	INT	代替モード		
				UNITS(一の位):	最終深さ(<_DP>)	
					0 =	アブソリュート
					1 =	インクレメンタル
				TENS(十の位):	平面切り込みの単位(<_MIDA>)	
					0 =	mm
					1 =	工具直径の%
				HUNDREDS(百の位):	面取りの切り込み深さ(<_ZFS>)	
					0 =	アブソリュート
					1 =	インクレメンタル
				THOUSANDS(千の位):	コーナ丸み付け(<_RC>)	
					0 =	半径
					1 =	面取り

20.1.21 CYCLE81 - 穴あけ、センタリング

構文

```
CYCLE81(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<RTP>	REAL	イニシャル点(abs)
2	Z0	<RFP>	REAL	基準点(abs)
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1/∅	<DP>	REAL	穴あけ深さ(abs)/センタリング径(abs)、<_GMODE>を参照
5	Z1	<DPR>	REAL	穴あけ深さ(inc)
6	DT	<DTB>	REAL	最終穴あけ深さでのドウェル時間、<_AMODE>を参照

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
7		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	予備	
				TENS(十の位):	深さ/直径を基準にしたセンタリング	
					0 =	互換性、深さ
					1 =	Diameter
8		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
9		<_AMODE>	INT	代替モード		
				UNITS(一の位):	穴あけ深さ Z1 (abs/inc)	
					0 =	互換性、DP/DPR のプログラム指令から
					1 =	インクレメンタル
					2 =	アブソリュート
				TENS(十の位):	最終穴あけ深さ DT でのドウェル時間(秒(s)/回転数(rev)の単位)	
					0 =	互換性、DTB の符号から(> 0 は秒(s)、< 0 は回転数(rev))
					1 =	秒(s)
					2 =	回転数(rev)

20.1.22 CYCLE82 - 穴あけ、座ぐり

構文

```
CYCLE82(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>,
<_DMODE>, <_AMODE>, <_VARI>, <S_ZA>, <S_FA>, <S_ZD>, <S_FD>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1	RP	<RTP>	REAL	イニシャル点(abs)		
2	Z0	<RFP>	REAL	基準点(abs)		
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)		
4	Z1	<DP>	REAL	穴あけ深さ(abs)、<_AMODE>を参照		
5	Z1	<DPR>	REAL	穴あけ深さ(inc)、<_AMODE>を参照		
6	DT	<DTB>	REAL	最終穴あけ深さでのドウェル時間、<_AMODE>を参照		
7		<_GMODE>	INT	ジオメトリックモード(プログラム指令されたジオメトリックデータの評価)		
				UNITS(一の位):	予備	
				TENS(十の位):	先端/シャンクを基準にした穴あけ深さ	
					0 =	互換性、先端
					1 =	シャンク
8		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	予備	
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):	予備	
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1004)	
					0 =	入力:全て
					1 =	入力:ベーシック

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
9		<_AMODE>	INT	代替モード		
				UNITS(一の位):	穴あけ深さ Z1 (abs/inc)	
					0 =	互換性、DP/DPR のプログラム指令から
					1 =	インクレメンタル
					2 =	アブソリュート
				TENS(十の位):	最終穴あけ深さでのドウェル時間 DT(秒(s)/回転数(rev)の単位)	
					0 =	互換性、DT の符号から(> 0 は秒(s)、< 0 は回転数(rev))
					1 =	秒(s)
					2 =	回転数(rev)
				HUNDREDS(百の位):	穴あけ深さ DZA abs/inc	
					0 =	インクレメンタル
					1 =	アブソリュート
				THOUSANDS(千の位):	予備穴あけ送り速度の評価	
					0 =	穴あけ送り速度の%として
					1 =	F/min
					2 =	F/rev
				TEN THOUSANDS(万の位):	残りの穴あけ深さ ZD abs/inc	
					0 =	インクレメンタル
					1 =	アブソリュート
				HUNDRED THOUSANDS(十万の位):	残りの穴あけ送り速度の評価	
					0 =	穴あけ送り速度の%として
					1 =	F/min
					2 =	F/rev

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
10		<_VARI>	INT	予備穴あけ/貫通穴あけ加工タイプ		
				UNITS(一の位):	予備	
				TENS(十の位):	予備	
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):	貫通穴あけ	
					0 =	貫通穴あけ「なし」
					1 =	貫通穴あけ「あり」
				TEN THOUSANDS(万の位):	前加工	
					0 =	予備穴あけ「なし」
					1 =	予備穴あけ「あり」
11	ZA	<S_ZA>	REAL	レファレンス点またはアブソリュートを基準にしたインクレメンタル予備穴あけ深さ(<_AMODE> HUNDREDS を参照)		
12	FA	<S_FA>	REAL	予備穴あけ送り速度(値としてまたは%単位で)(<_AMODE> THOUSANDS と組み合わせて)		
13	ZD	<S_ZD>	REAL	最終穴あけ深さまたはアブソリュートを基準にしたインクレメンタルの残りの穴あけ深さ(<_AMODE> TEN THOUSANDS)を参照)		
14	FD	<S_FD>	REAL	残りの穴あけ送り速度(値としてまたは%単位で)(<_AMODE> HUNDRED THOUSANDS と組み合わせて)		

20.1.23 CYCLE83 - 深穴あけ

構文

```
CYCLE83(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <FDEP>, <FDPR>,
<_DAM>, <DTB>, <DTS>, <FRF>, <VARI>, <_AXN>, <_MDEP>, <_VRT>,
<_DTD>, <_DIS1>, <_GMODE>, <_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1	RP	<RTP>	REAL	イニシャル点(abs)		
2	Z0	<RFP>	REAL	基準点(abs)		
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)		
4	Z1	<DP>	REAL	最終穴あけ深さ(abs)、<_AMODE>を参照		
5	Z1	<DPR>	REAL	最終穴あけ深さ(inc)、<_AMODE> を参照		
6	D	<FDEP>	REAL	1 番目の穴あけ深さ(abs)、<_AMODE>を参照		
7	D	<FDPR>	REAL	1 番目の穴あけ深さ(inc)、<_AMODE>を参照		
8	DF	<_DAM>	REAL	各々の追加切り込みの逓減値/パーセント値。<_AMODE> を参照。		
9	DTB	<DTB>	REAL	穴あけ深さでのドウェル時間、 <_AMODE> を参照		
10	DTS	<DTS>	REAL	開始点でのドウェル時間(切り屑除去のみ)、<_AMODE>を参照		
11	FD1	<FRF>	REAL	最初の切り込み送り速度のパーセント値、<_AMODE>を参照		
12		<VARI>	INT	加工タイプ		
				UNITS(一の位):	切り屑破断/切り屑除去	
					0 =	切り屑切断
					1 =	切り屑除去
13		<_AXN>	INT	工具軸		
					0 =	3 番目のジオメトリ軸
					1 =	1 番目のジオメトリ軸
					2 =	2 番目のジオメトリ軸
					> 2	3 番目のジオメトリ軸
14	V1	<_MDEP>	REAL	最小切り込み(逓減パーセント値のみ)		
15	V2	<_VRT>	REAL	各加工ステップ終了後の後退量(切り屑破断の場合のみ)		
					> 0	可変 後退距離
					0 =	初期値 1 mm
16	DT	<_DTD>	REAL	最終穴あけ深さでのドウェル時間、<_AMODE>を参照		
17	V3	<_DIS1>	REAL	リミット距離(切り屑除去の場合のみ)、<_AMODE>を参照		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
18		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	予備	
				TENS(十の位):	先端/シャンクを基準にした穴あけ深さ	
					0 =	先端
					1 =	シャンク
19		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	---	予備
				HUNDREDS(百の位):	---	予備
				THOUSANDS(千の位):	---	予備
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1004)	
					0 =	入力:全て
					1 =	入力:簡易

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
20		<_AMODE>	INT	代替モード		
				UNITS(一の位):	穴あけ深さ = 最終穴あけ深さ Z1 (abs/inc)	
					0 =	互換性、 <DP>/<DPR>のプログラム指令から
					1 =	インクレメンタル
					2 =	アブソリュート
				TENS(十の位):	穴あけ深さでのドウェル時間 DTB(秒(s)/回転数(rev))	
					0 =	互換性、DTB の符号から(> 0 は秒(s)、< 0 は回転数(rev))
					1 =	秒(s)
					2 =	回転数(rev)
				HUNDREDS(百の位):	DTS の開始点でのドウェル時間(秒(s)/回転(rev))	
					0 =	互換性、DTS の符号から(> 0 は秒(s)、< 0 は回転数(rev))
					1 =	秒(s)
					2 =	回転数(rev)
				THOUSANDS(千の位):	最終穴あけ深さでのドウェル時間 DTD(秒(s)/回転数(rev)の単位)	
					0 =	互換性、DTD の符号から(> 0 は秒(s)、< 0 は回転数(rev))
					1 =	秒(s)
					2 =	回転数(rev)
				TEN THOUSANDS(万の位):	1 番目の穴あけ深さ D (abs/inc)	
					0 =	互換性、<FDEPF>/<DPR>のプログラム指令から
					1 =	インクレメンタル
					2 =	アブソリュート

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
				HUNDRED THOUSANDS(十万の位):	各々の追加切り込みの逓減値/パーセント値<_DAM>	
					0 =	互換性、<_DAM>の符号から(> 0 は逓減値、< 0 は係数 0.001~1.0)
					1 =	逓減値
					2 =	パーセント値(0.001~100%)
				ONE MILLION(百万の位):	リミット距離 V3 自動/手動	
					0 =	互換性、<_DIS1> の符号から(= 0 は自動、> 0 は手動)
					1 =	自動(サイクルで計算)
					2 =	手動(プログラム指令値)
				TEN MILLION(千万の位):	係数/パーセント値としての最初の切り込み送り速度係数 <FRF>	
					0 =	互換性、係数として(0.001 ～ 1.0、FRF = 0 は 100%)
					1 =	パーセント値 (0.001~999.999%)

20.1.24 CYCLE84 - フローティングチャックなしのタッピング

構文

```
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>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<RTP>	REAL	イニシャル点(abs)
2	Z0	<RFP>	REAL	基準点(abs)

No.	マスクパラメータ	内部パラメータ	データタイプ	意味			
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)			
4	Z1	<DP>	REAL	穴あけ深さ = 最終穴あけ深さ(abs)、 <_AMODE>を参照			
5	Z1	<DPR>	REAL	穴あけ深さ = 最終穴あけ深さ(inc)、<_AMODE>を参照			
6	DT	<DTB>	REAL	穴あけ深さでのドウェル時間(s)			
7	SDE	<SDAC>	INT	サイクル終了後の回転方向			
8		<MPIT>	REAL	「ISO メトリック」のねじサイズのみ(ピッチは運転時に内部で計算されます)			
9	P	<PIT>	REAL	数値で表したピッチ、単位については <_PITA>を参照			
10	αS[1)]	<POSS>	REAL	主軸オリエンテーションの主軸位置			
11	S	<SST>	REAL	タッピングの主軸速度			
12	SR	<SST1>	REAL	後退時の主軸速度			
13		<_AXN>	INT	穴あけ軸	0 =	3 番目のジオメトリ軸	
					1 =	1 番目のジオメトリ軸	
					2 =	2 番目のジオメトリ軸	
					≥ 3 =	3 番目のジオメトリ軸	
14		<_PITA>	INT	ピッチ単位(<PIT>と<MPIT>の評価)			
					0 =	ピッチ(mm)	- 評価<MPIT>/<PIT>
					1 =	ピッチ(mm)	- 評価<PIT>
					2 =	ピッチ(TPI)	- <PIT>の評価(ねじ山数/inch)
					3 =	ピッチ(inch)	- 評価<PIT>
					4 =	MODULUS	- 評価<PIT>

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
15		<_TECHNO>	INT	用途 [1]		
				UNITS(一の位):	イグザクトストップ動作	
					0 =	サイクル呼び出し前と同じイグザクトストップ動作が有効
					1 =	イグザクトストップ G601
					2 =	イグザクトストップ G602
					3 =	イグザクトストップ G603
				TENS(十の位):	フィードフォワード制御	
					0 =	サイクル呼び出し前と同じフィードフォワード制御あり/なしが有効
					1 =	フィードフォワード制御あり FFWON
					2 =	フィードフォワード制御なし FFWOF
				HUNDREDS(百の位):	加減速	
					0 =	サイクル呼び出し前と同じ SOFT/BRISK/DRIVE が有効
					1 =	加々速度制限あり SOFT
					2 =	加々速度制限なし BRISK
					3 =	加速度てい減 DRIVE
				THOUSANDS(千の位):	MCALL 主軸モード	
					0 =	MCALL の主軸運転を再起動
					1 =	MCALL で位置制御を維持
16		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	0 =	1 回切削
					1 =	切り屑破断(深穴タッピング)
					2 =	切り屑除去(深穴タッピング)
				THOUSANDS(千の位):	ISO/シーメンスモードは入力画面と無関係	
					0 =	ISO 互換からの呼び出し
					1 =	シーメンスコンテキストからの呼び出し

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
17	d	<_DAM>	REAL	最大切り込み深さ(切り屑除去/切り屑破断のみ)	
18	V2	<_VRT>	REAL	各加工ステップ終了後の後退量(切り屑破断の場合のみ)、<_AMODE>を参照	
19		<_PITM>	STRING[15]	ピッチ入力のマーカとなる文字列 2)	
20		<_PTAB>	STRING[5]	ねじテーブルの文字列(""、「ISO」、「BSW」、「BSP」、「UNC」)2)	
21		<_PTABA>	STRING[20]	ねじテーブルからの選択のための文字列(例:「M 10」、「M 12」、...)2)	
22		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)	
				UNITS(一の位):	予備
				TENS(十の位):	予備

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
23		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	予備	
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):	互換性モード(再コンパイル入力画面のみ)、 MD 52216 ビット 0 = 1 の場合 [1])	
					0 =	テクノロジパラメータが表示されます(互換性):TECHNO パラメータが有効
					1 =	テクノロジパラメータは表示されません:「サイクル呼び出し前と同じ」プログラム値が有効
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1004)	
					0 =	入力:全て
					1 =	入力:簡易

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
24		<_AMODE>	INT	代替モード		
				UNITS(一の位):	穴あけ深さ = 最終穴あけ深さ Z1 (abs/inc)	
					0 =	互換性、 <DP>/<DPR>のプログラム指令から
					1 =	インクレメンタル
					2 =	アブソリュート
				TENS(十の位):	予備	
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):	ねじの回転方向、右/左	
					0 =	互換性、PIT/MPTI の符号から
					1 =	右側
					2 =	左側
				TEN THOUSANDS(万の位):	予備	
				HUNDRED THOUSANDS(十万の位):	予備	
				ONE MILLION(百万の位):	各加工ステップ終了後の後退量 V2、手動/自動	
					0 =	互換性、<_VRT> のプログラム指令から(> 0 は可変値、≤ 0 は標準値 1 mm/0.0394 inch)
					1 =	自動(標準値 1 mm / 0.0394 inch)
					2 =	手動(V2 以下となるようプログラム指令)

[1] テクノロジフィールドは、SD52216 $MCS_FUNCTION_MASK_DRILL の設定に応じて非表示になる場合があります。

[2] パラメータ 19、20、21 は、入力画面ねじテーブルでのねじの選択にのみ使用されます。ねじテーブルは、サイクル運転時にサイクル定義を介してアクセスすることはできません。

20.1.25 CYCLE85 - リーマ加工

構文

```
CYCLE85(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <FFR>, <RFF>,
<_GMODE>, <_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1	RP	<RTP>	REAL	イニシャル点(abs)		
2	Z0	<RFP>	REAL	基準点(abs)		
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)		
4	Z1	<DP>	REAL	穴あけ深さ(abs)、<_AMODE>を参照		
5	Z1	<DPR>	REAL	穴あけ深さ(inc)、<_AMODE>を参照		
6	DT	<DTB>	REAL	最終穴あけ深さでのドウェル時間、<_AMODE>を参照		
7	F	<FFR>	REAL	送り速度		
8	FR	<RFF>	REAL	後退時の送り速度		
9		<_GMODE>	INT	予備		
10		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
11		<_AMODE>	INT	代替モード(穴あけ)		
				UNITS(一の位):	穴あけ深さ Z1 (abs/inc)	
					0 =	互換性、DP/DPR のプログラム指令から
					1 =	インクレメンタル
					2 =	アブソリュート
				TENS(十の位):	最終穴あけ深さでのドウェル時間 DT(秒(s)/回転数(rev))	
					0 =	互換性、DT の符号から(> 0 は秒(s)、< 0 は回転数(rev))
					1 =	秒(s)
					2 =	回転数(rev)

20.1.26 CYCLE86 - ボーリング

構文

```
CYCLE86(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDIR>, <RPA>,
<RPO>, <RPAP>, <POSS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1	RP	<RTP>	REAL	イニシャル点(abs)		
2	Z0	<RFP>	REAL	基準点(abs)		
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)		
4	Z1	<DP>	REAL	穴あけ深さ(abs)、 <_AMODE>を参照		
5	Z1	<DPR>	REAL	穴あけ深さ(inc)、 <_AMODE>を参照		
6	DT	<DTB>	REAL	最終穴あけ深さでのドウェル時間、<_AMODE>を参照		
7	DIR	<SDIR>	INT	主軸回転方向	3 =	M3
					4 =	M4
8	Dx	<RPA>	REAL	X 方向の戻し距離		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
9	DY	<RPO>	REAL	Y 方向の戻し距離		
10	DZ	<RPAP>	REAL	Z 方向の戻し距離		
11	SPOS	<POSS>	REAL	戻し時の主軸位置(主軸オリエンテーションの場合、° 単位)		
12		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	戻しモード	
					0 =	戻し、互換性
					1 =	輪郭から戻しをおこなわない
13		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
14		<_AMODE>	INT	代替モード		
				UNITS(一の位):	穴あけ深さ Z1 (abs/inc)	
					0 =	互換性、<DP>/<DPR> のプログラム指令から
					1 =	インクレメンタル
					2 =	アブソリュート
				TENS(十の位):	最終穴あけ深さ DT でのドウェル時間(秒(s)/回転数(rev))	
					0 =	互換性、DT の符号から(> 0 は秒(s)、< 0 は回転数(rev))
					1 =	秒(s)
					2 =	回転数(rev)

20.1.27 CYCLE92 - 突切り

構文

```
CYCLE92(<_SPD>, <_SPL>, <_DIAG1>, <_DIAG2>, <_RC>, <_SDIS>,
<_SV1>, <_SV2>, <_SDAC>, <_FF1>, <_FF2>, <_SS2>, <_DIAGM>,
<_VARI>, <_DN>, <_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1	X0	<_SPD>	REAL	基準点(abs、常に直径)		
2	Y0	<_SPL>	REAL	基準点(abs)		
3	X1	<_DIAG1>	REAL	減速時の深さ、<_AMODE> (UNITS)を参照		
4	X2	<_DIAG2>	REAL	最終深さ、<_AMODE> (TENS)を参照		
5	R/FS	<_RC>	REAL	丸み付け状態または面取り幅、<_AMODE> (THOUSANDS)を参照		
6	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)		
7	S	<_SV1>	REAL	一定主軸速度、<_AMODE> (TEN THOUSANDS)を参照		
	V			定切削速度		
8	SV	<_SV2>	REAL	定切削速度時の最大速度		
9	DIR	<_SDAC>	INT	主軸回転方向	3 =	M3 の場合
					4 =	M4 の場合
10	F	<_FF1>	REAL	減速時の深さまで切り込み		
11	FR	<_FF2>	REAL	最終深さまでの減速切り込み		
12	SR	<_SS2>	REAL	最終深さまでの減速速度		
13	XM	<_DIAGM>	REAL	部品固定具引き抜き深さ(abs、常に直径)		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
14		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	後退	
					0 =	<_SPD> + <_SDIS>へ後退
					1 =	最後での後退なし
				TENS(十の位):	部品固定具	
					0 =	なし、M 命令を実行しない
					1 =	あり、CUST_TECHCYC(101)から呼び出し- ドロアを開く、CUST_TECHCYC(102)- ドロアを閉じる
15		<_DN>	INT	工具の 2 番目の刃先の D 番号。プログラム指令されていない場合 ⇒ D+1		
20		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
21		<_AMODE>	INT	代替モード		
				UNITS(一の位):	減速時の深さ(<_DIAG1>)	
					0 =	アブソリュート、径方向軸の直径の値
					1 =	インクレメンタル、径方向軸の半径の値
				TENS(十の位):	最終深さ(<_DIAG2>)	
					0 =	アブソリュート、径方向軸の直径の値
					1 =	インクレメンタル、径方向軸の半径の値
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):	丸み付け/面取り(<_RC>)	
					0 =	半径
					1 =	面取り
				TEN THOUSANDS(万の位):	主軸速度/切削速度(<_SV1>)	
					0 =	一定主軸速度
					1 =	定切削速度

20.1.28 CYCLE95 - 輪郭切削

構文

```
CYCLE95(<NPP>, <MID>, <FALZ>, <FALX>, <FAL>, <FF1>, <FF2>, <FF3>,
<_VARI>, <DT>, <DAM>, <_VRT>, <_GMODE>, <_DMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	CON	<NPP>	STRING [140]	輪郭名称
2	d	<MID>	REAL	荒削り時の最大切り込み深さ、 <_GMODE>を参照

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
3	UZ	<FALZ>	REAL	Z の仕上げ代		
4	UX	<FALX>	REAL	X の仕上げ代		
5	U	<FAL>	REAL	輪郭に平行の仕上げ代(両方の軸で有効)		
6	F	<FF1>	REAL	荒削りの送り速度		
7	FY	<FF2>	REAL	レリーフカット切り込み送り速度		
8	FS	<FF3>	REAL	仕上げの送り速度		
9		<_VARI>	INT	加工タイプ		
				UNITS(一の位)と TENS(十の位):		
					1 =	荒削り、長手方向、外部
					2 =	荒削り、径方向、外部
					3 =	荒削り、長手方向、内部
					4 =	荒削り、径方向、内部
					5 =	仕上げ、長手方向、外部
					6 =	仕上げ、径方向、外部
					7 =	仕上げ、長手方向、内部
					8 =	仕上げ、径方向、内部
					9 =	全て加工、長手方向、外部
					10 =	全て加工、径方向、外部
					11 =	全て加工、長手方向、内部
					12 =	全て加工、径方向、内部
				HUNDREDS(百の位):		
					0 =	輪郭での丸み付けあり、コーナの削り残しなし
					1 =	輪郭での丸み付けなし
					2 =	前の交点に対してのみ丸み付け、コーナの削り残しが生じるおそれあり
10	DT	<DT>	REAL	送り中断時のドウェル時間		
11	Di	<DAM>	REAL	送り中断の距離		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
12	VRT	<_VRT>	REAL	輪郭からの戻し距離		
					0 =	有効な単位(インチまたはメトリック)とは無関係に、戻し距離 1 mm が内部的に使用されます。
					> 0 =	戻し距離
13		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	切り込み深さの評価	
					0 =	切り込み深さは、G グループ DIAMON/DIAMOF に対応して計算されます
					1 =	切り込み深さは、半径値として機能します(DIAMON/DIAMOF とは無関係)
14		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				THOUSANDS(千の位):		
					0 =	互換性モード:NPP 内の輪郭名称
					1 =	輪郭名称は CYCLE62 でプログラム指令され、_SC_CONT_NAME に転送されます。

20.1.29 CYCLE98 - 連続ねじ

構文

```
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>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	Z0	<_PO1>	REAL	Z の基準点(abs)
2	X0	<_DM1>	REAL	X の基準点(abs)、直径
3	Z1	<_PO2>	REAL	Z の中間点 1 (abs/inc)、<_AMODE> (UNITS)を参照
4	X1	<_DM2>	REAL	X の中間点 1 (abs/inc)、<_AMODE> (TENS)を参照、または
	X1α			ねじ傾斜角 1 (-90°～ 90°) abs では常に直径、inc では常に半径
5	Z2	<_PO3>	REAL	Z の中間点 2 (abs/inc)、<_AMODE> (HUNDREDS)を参照
6	X2	<_DM3>	REAL	X の中間点 2 (abs/inc)、<_AMODE> (THOUSANDS)を参照、または
	X2α			ねじ傾斜角 2 (-90°～ 90°) abs では常に直径、inc では常に半径
7	Z3	<_PO4>	REAL	Z の終点、(abs/inc)、 <_AMODE>(TEN THOUSANDS)を参照
8	X3	<_DM4>	REAL	X の終点、(abs/inc)、<_AMODE> (HUNDRED THOUSANDS)を参照、または
	X3α			ねじ傾斜角 3 (-90°～ 90°) abs では常に直径、inc では常に半径
9	LW	<APP>	REAL	ねじ切り始め(inc、符号なしで入力)
10	LR	<ROP>	REAL	ねじ切り上げ(inc、符号なしで入力)
11	H1	<TDEP>	REAL	ねじ山深さ(inc、符号なしで入力)
12	U	<FAL>	REAL	X と Z の仕上げ代

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
13	DP	<_IANG>	REAL	距離または角度で表した切り込み斜面、<_AMODE> (ONE MILLION)を参照		
	αP			切り込み斜面は、パラメータ<_VARI>(HUNDREDS)の設定に従って適用されます。		
					<_VARI_HUNDREDS = 0 の定義 - 互換性モード:	
					> 0 =	片面だけを切り込み
					0 =	ねじを垂直切り込み
					< 0 =	面を交互に切り込み
					_VARI_HUNDREDS<>0 の定義:	
					> 0 =	プラス側の切り込み
					0 =	中央に配置した電源装置
					< 0 =	マイナス側の切り込み
14	α0	<NSP>	REAL	1 番目のねじの開始角度オフセット		
15		<NRC>	INT	荒削り切削の回数、<_VARI> (TEN THOUSANDS)を参照		
16	NN	<NID>	INT	エアーカットの回数		
17	P0	<_PP1>	REAL	ねじの 1 番目の区間のピッチ、<_PITA>を参照		
18	P1	<_PP2>	REAL	ねじの 2 番目の区間のピッチ、<_PITA>を参照		
19	P2	<_PP3>	REAL	ねじの 3 番目の区間のピッチ、<_PITA>を参照		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
20		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	用途	
					1 =	直線切り込みのおねじ
					2 =	直線切り込みのめねじ
					3 =	逓減切り込みのおねじ、切削の断面は一定
					4 =	逓減切り込みのめねじ、切削の断面は一定
				TENS(十の位):	予備	
				HUNDREDS(百の位):	切り込みタイプ	
					0 =	<_IANG>の互換性モード
					1 =	片面だけを切り込み
					2 =	両面を交互に切り込み
				THOUSANDS(千の位):	予備	
				TEN THOUSANDS(万の位):	交互深さ切り込み	
					0 =	互換性、荒削り切削の回数を設定(<_NRC>)
					1 =	1 回目の切り込みの初期設定(<_MID>)
				HUNDRED THOUSANDS(十万の位):	加工タイプ	
					0 =	互換性(荒削りと仕上げ)
					1 =	荒削り
					2 =	仕上げ
					3 =	荒削りと仕上げ
				ONE MILLION(百万の位):	多条ねじの加工処理	
					0 =	ねじを昇順で加工
					1 =	ねじを降順で加工
21	n	<_NUMTH>	INT	ねじの数		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
22		<_VRT>	REAL	戻し距離(inc)		
					0 =	有効な単位(インチまたはメトリック)とは無関係に、戻し距離 1 mm が内部的に使用されます。
					> 0 =	戻し距離
23	D1	<_MID>	REAL	1 番目の切り込み<_VARI> (TEN THOUSANDS)を参照		
24	DA	<_GDEP>	REAL	ねじ切り替え深さ(「多条ねじ」でのみ有効)		
					0 =	ねじ切り替え深さに従わない
					> 0 =	ねじ切り替え深さに従う
25		<_IFLANK>	REAL	幅で表された切込み斜面(インタフェースのみ)		
26		<_PITA>	INT	ねじピッチの評価		
					0 =	ピッチの互換性モード、有効な単位(メトリック/インチ)に従って、以前と同じように<_PP1>～<_PP3>を評価
					1 =	ピッチ(mm)
					2 =	ピッチ(TPI)(ねじ山数/inch)
					3 =	ピッチ(inch)
					4 =	MODULUS
27		<_PITM1>	STRING[15]	ピッチ入力のマーカとなる文字列(インタフェースのみ)		
28		<_PITM2>	STRING[15]	ピッチ入力のマーカとなる文字列(インタフェースのみ)		
29		<_PITM3>	STRING[15]	ピッチ入力のマーカとなる文字列(インタフェースのみ)		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
30		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	---	予備
				HUNDREDS(百の位):	---	予備
				THOUSANDS(千の位):	---	予備
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1004)	
					0 =	入力:全て
					1 =	入力:簡易

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
31		<_AMODE>	INT	代替モード		
				UNITS(一の位):	1 番目の Z の中間点(Z1)	
					0 =	アブソリュート
					1 =	インクレメンタル
				TENS(十の位):	1 番目の X の中間点(X1)	
					0 =	アブソリュート
					1 =	インクレメンタル
					2 =	α
				HUNDREDS(百の位):	2 番目の Z の中間点(Z2)	
					0 =	アブソリュート
					1 =	インクレメンタル
				THOUSANDS(千の位):	2 番目の X の中間点(X2)	
					0 =	アブソリュート
					1 =	インクレメンタル
					2 =	α
				TEN THOUSANDS(万の位):	Z の終点(Z3)	
					0 =	アブソリュート
					1 =	インクレメンタル
				HUNDRED THOUSANDS(十万の位):	X の終点(X3)	
					0 =	アブソリュート
					1 =	インクレメンタル
					2 =	α
				ONE MILLION(百万の位):	切り込み斜面を角度または幅で選択	
					0 =	切り込み角度<_IANG>
					1 =	切り込み斜面<_IFLANK>
				TEN MILLION(千万の位):	一条ねじ/多条ねじ	
					0 =	互換性モード(開始角度<_NSP>が評価されます)
					1 =	一条ねじ(開始角度オフセット<_NSP>)

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
					2 =	多条

20.1.30 CYCLE99 - ねじの旋削

構文

```
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>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1	Z0	<_SPL>	REAL	基準点(abs)		
2	X0	<_SPD>	REAL	基準点(abs、常に直径)		
3	Z1	<_FPL>	REAL	<_AMODE> (UNITS)と組み合わせた終点		
4	X1	<_FPD>	REAL	<_AMODE>(TENS)と組み合わせた終点		
5	LW/LW2	<_APP>	REAL	<_AMODE> (HUNDREDS)と組み合わせたねじ切り始め、または <_AMODE> (HUNDREDS)と組み合わせたねじ切り始めとねじ切り上げ		
6	LR	<_ROP>	REAL	ねじ切り上げ		
7	H1	<_TDEP>	REAL	ねじ山深さ		
8	U	<_FAL>	REAL	X と Z の仕上げ代		
9	DP	<_IANG>	REAL	<_AMODE> (THOUSANDS)と組み合わせた、距離または角度で表した切り込み傾斜		
	αP				> 0 =	プラス側の切り込み
					< 0 =	マイナス側の切り込み
					0 =	中央に配置した電源装置
10	α0	<_NSP>	REAL	開始角度オフセット(「一条ねじ」でのみ有効)		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
11	ND	<_NRC>	INT	荒削り切削の回数、<_VARI> (TEN THOUSANDS)と組み合わせ		
12	NN	<_NID>	INT	エアーカットの回数		
13	P	<_PIT>	REAL	<_PITA>と組み合わせた値としてのピッチ		
14		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	用途	
					1 =	直線切り込みのおねじ
					2 =	直線切り込みのめねじ
					3 =	逓減切り込みのおねじ、切削の断面は一定
					4 =	逓減切り込みのめねじ、切削の断面は一定
				TENS(十の位):	予備	
				HUNDREDS(百の位):	切り込みタイプ	
					1 =	片面だけを切り込み
					2 =	両面を交互に切り込み
				THOUSANDS(千の位):	予備	
				TEN THOUSANDS(万の位):	交互深さ切り込み	
					0 =	荒削り切削の回数を設定 (<_NRC>)
					1 =	1 回目の切り込みの初期設定 (<_MID>)
				HUNDRED THOUSANDS(十万の位):	加工タイプ	
					1 =	荒削り
					2 =	仕上げ
					3 =	荒削りと仕上げ
				ONE MILLION(百万の位):	多条ねじの加工処理	
					0 =	ねじを昇順で加工
					1 =	ねじを降順で加工
15	n	<_NUMTH>	INT	ねじの数		
16	VR	<_SDIS>	REAL	戻り距離、inc		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
17	D1	<_MID>	REAL	<_VARI> (TEN THOUSANDS)と組み合わせた 1 番目の切り込み深さ		
18	DA	<_GDEP>	REAL	ねじ切り替え深さ(「多条ねじ」でのみ有効)		
					0 =	ねじ切り替え深さに従わない
					> 0 =	ねじ切り替え深さに従う
19	g	<_PIT1>	REAL	一回転当たりのピッチの変更		
					0 =	ピッチは一定(G33)
					> 0 =	ピッチを増やす(G34)
					< 0 =	ピッチを減らす(G35)
20		<_FDEP>	REAL	切り込み深さ(符号なしで入力)		
21	N1	<_GST>	INT	<_AMODE> (HUNDRED THOUSANDS)と組み合わせた開始ねじ N1 = 1...N		
22		<_GUD>	INT	予備		
23		<_IFLANK>	REAL	幅で表された切込み斜面(インタフェースのみ)		
24		<_PITA>	INT	ピッチの単位(PIT および/または MPIT の評価)		
					0 =	ピッチ(mm) - MPIT/PIT 評価
					1 =	ピッチ(mm) - PIT 評価
					2 =	ピッチ(TPI) - PIT 評価(ねじ山数/inch)
					3 =	ピッチ(inch) - PIT 評価
					4 =	MODULE - PIT 評価
25		<_PITM>	STRING[15]	ピッチ入力のマーカとなる文字列(インタフェースのみ)[1)]		
26		<_PTAB>	STRING[20]	ねじテーブルの文字列(インタフェースのみ)[1)]		
27		<_PTABA>	STRING[20]	ねじテーブルの選択のための文字列(インタフェースのみ)[1)]		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
28		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	ねじのタイプ	
					0 =	長手方向ねじ
					1 =	スクロールねじ
					2 =	テーパねじ
				HUNDREDS(百の位):	---	予備
				THOUSANDS(千の位):	---	予備
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1004)	
					0 =	入力:全て
					1 =	入力:ベーシック

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
29		<_AMODE>	INT	代替モード		
				UNITS(一の位):	Z のねじ長	
					0 =	アブソリュート
					1 =	インクレメンタル
				TENS(十の位):	X のねじ長	
					0 =	アブソリュート、径方向軸の直径の値
					1 =	インクレメンタル、径方向軸の半径の値
					2 =	α
				HUNDREDS(百の位):	アプローチ/ねじ切り始めの軌跡 <_APP>の計算	
					0 =	ねじ切り始め<_APP>
					1 =	ねじ切り始め = ねじ切り上げ <_APP>= -<_ROP>
					2 =	ねじ切り始め軌跡の指定 <_APP>= -<_APP>
				THOUSANDS(千の位):	切り込み斜面を角度または幅で選択	
					0 =	切り込み角度<_IANG>
					1 =	切り込み斜面<_IFLANK>
				TEN THOUSANDS(万の位):	一条ねじ/多条ねじ	
					0 =	一条ねじ(開始角度オフセット<_NSP>)
					1 =	多条
				HUNDRED THOUSANDS(十万の位):	開始ねじ<_GST>	
					0 =	完全な加工
					1 =	このねじから加工を開始
					2 =	このねじのみを加工
				ONE MILLION(百万の位):	長手方向ねじの真直度補正	
					0 =	セグメントの高さ、凹/凸ネジ XS

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
					1 =	凹/凸ネジ半径 RS
30	XS/RS	<_S_XRS>	REAL	<_AMODE>と組み合わせた長手方向ねじの真直度補正:ONE MILLION(百万の位)		

注記

1)パラメータ <_PITM>, <_PTAB>と<_PTABA>は、入力画面のねじテーブルでのねじの選択にのみ使用されます。
ねじテーブルは、サイクル運転時にサイクル定義を介してアクセスすることはできません。

20.1.31 CYCLE435 - ドレッサー座標系の設定

構文

```
CYCLE435(<_T>, <_DD>, <S_TA>, <S_DA>, <S_AD>, <S_AL>, <S_PVD>,
<S_PVL>, <S_PD>, <S_PL>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1		<_T>	STRING[32]	砥石の工具名
2		<_DD>	INT	砥石の刃先番号
3		<S_TA>	STRING[32]	目立て工具のレファレンス点 - 目立て工具の名称
4		<S_DA>	INT	目立て工具の刃先番号
5		<S_AD>	REAL	目立て値、直径
6		<S_AL>	REAL	目立て値、面
7		<S_PVD>	REAL	成形研削オフセット、直径
8		<S_PVL>	REAL	成形研削オフセット、面
9		<S_PD>	REAL	成形研削代、直径
10		<S_PL>	REAL	成形研削代、面

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
11		<_AMODE>	INT	代替モード		
				UNITS(一の位):	サイクルの終了時に有効な工具	
					0 =	目立て工具が有効
					1 =	といしが有効

20.1.32 CYCLE495 - 成形研削

構文

```
CYCLE495(<_T>, <_DD>, <_SC>, <_F>, <_VARI>, <_D>, <_DX>, <_DZ>,
<S_PA>, <S_N>, <_DMODE>, <_AMODE>, <S_FW>, <S_HW>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1		<_T>	STRING[20]	砥石の工具名
2		<_DD>	INT	砥石の刃先番号
3		<_SC>	REAL	障害物を回避するための戻し距離、インクレメンタル
4		<_F>	REAL	成形研削の送り速度

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
5		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	成形研削のタイプ	
					1 =	軸と平行
					2 =	輪郭に平行
				TENS(十の位):	加工方向	
					0 =	引き 刃先位置 1 ～ 4 で可能
					1 =	押し 刃先位置 1 ～ 4 で可能
					2 =	交互 刃先位置 1 ～ 8 で可能
					3 =	始点 → 終点 刃先位置 1 ～ 8 で可能
					4 =	終点 → 始点 刃先位置 1 ～ 8 で可能
				HUNDREDS(百の位):	切り込み方向	
					1 =	G18 の場合切込み X、または G19 の場合切込み Y
					2 =	G18 の場合切込み X+、または G19 の場合切込み Y+
					3 =	G18 および G19 の場合切込み Z-
					4 =	G18 および G19 の場合切込み Z+
6		<_D>	REAL	軸に平行な成形研削タイプの目立て値		
7		<_DX>	REAL	輪郭に平行な成形研削タイプの G18 の目立て値 X と G19 の目立て値 Y		
8		<_DZ>	REAL	軸に平行な成形研削タイプの G18 および G19 の目立て値 Z		
9		<S_PA>	REAL	成形研削代		
10		<S_N>	INT	成形研削プログラムでのストローク数		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
11		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
12		<_AMODE>	INT	代替モード		
				UNITS(一の位):	成形研削の選択、新規/継続	
					1 =	新規
					2 =	続行
				TENS(十の位):	成形研削代の選択	
					0 =	荒削り輪郭から輪郭の最低点まで
					1 =	荒削り輪郭から輪郭の最高点まで
13		<S_FW>	REAL	目立て工具のクリア角度		
14		<S_HW>	REAL	目立て工具のホルダー角度		

20.1.33 CYCLE800 - 旋回

構文

```
CYCLE800(<_FR>, <_TC>, <_ST>, <_MODE>, <_X0>, <_Y0>, <_Z0>, <_A>,
<_B>, <_C>, <_X1>, <_Y1>, <_Z1>, <_DIR>, <_FR_I>, <_DMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1		<_FR>	INT	後退モード	0 =	後退なし
					1 =	機械軸 Z の後退
					2 =	機械軸 Z、次に XY の後退
					3 =	予備
					4 =	工具方向の最大後退
					5 =	工具方向のインクレメンタル後退
2		<_TC>	STRING[32]	旋回データブロックの名称:	""	""(名称なし)、存在する旋回データブロックが 1 つのみの場合
					"0"	旋回データブロックの選択を解除(旋回フレームを削除)

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
3		<_ST>	INT	座標変換状態		
				UNITS(一の位):		
					0 =	新規。旋回レベルが削除され、現在のパラメータを使用して再計算されます。
					1 =	追加。旋回レベルが現在の旋回レベルに追加されます。
				TENS(十の位):	工具先端追従の有無(旋回機能がセットアップで設定されている場合のみ有効)	
					0 =	工具先端を追従しない
					1 =	工具先端を追従する(TRAORI)
				HUNDREDS(百の位):	工具の移動/位置合わせ(機能は工具旋回入力画面で表示されます)	
					0 =	工具はアプローチしない
					1 =	工具はアプローチする(できれば、ラジアルミルで)
					2 =	**旋削工具**を位置合わせする(B軸キネマティックが旋回セットアップでフライス削り用に設定されている場合)
					3 =	**フライス工具**を位置合わせする(B 軸キネマティックが旋回セットアップでフライス削り用に設定されている場合)
				THOUSANDS(千の位):	内部「JOG での旋回」パラメータ	
				TEN THOUSANDS(万の位):	方向パラメータ<_DIR>を参照してください	
					0 =	旋回「あり」
					1 =	旋回「なし」、「負」方向 3)
					2 =	旋回「なし」、「正」方向 3)

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
				HUNDRED THOUSANDS(十万の位):	方向パラメータ<_DIR>を参照してください	
					0 =	互換性
					1 =	方向の選択「負」が最適化されています(ユーザーインタフェースのみ)[4)]
					2 =	方向の選択「正」が最適化されています(ユーザーインタフェースのみ)[4)]

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
4		<_MODE> 5)	INT	旋回モード旋回角度と旋回処理(ビット指定)の評価		
				ビット:7 6	0 0:	軸毎の旋回角度 -> パラメータ<_A>, <_B>, <_C> を参照してください
					0 1:	立体角 -> パラメータ<_A>, <_B>1)を参照してください
					1 0:	投影角 -> パラメータ<_A>, <_B>, <_C>1)を参照してください
					1 1:	ダイレクト回転軸旋回モード -> パラメータ<_A>, <_B>1)を参照してください
				ビット:5 4 3 2 1 0 (立体角には適用されません)	x x x x 0 1	X を中心とした 1 番目の回転_A
					x x x x 1 0	Y を中心とした 1 番目の回転_A
					x x x x 1 1	Z を中心とした 1 番目の回転_A
					x x 0 1 x x	X を中心とした 2 番目の回転_B
					x x 1 0 x x	Y を中心とした 2 番目の回転_B
					x x 1 1 x x	Z を中心とした 2 番目の回転_B
					0 1 x x x x	X を中心とした 3 番目の回転_C
					1 0 x x x x	Y を中心とした 3 番目の回転_C
					1 1 x x x x	Z を中心とした 3 番目の回転_C
5	X0	<_X0>	REAL	回転前の基準点 X		
6	Y0	<_Y0>	REAL	回転前の基準点 Y		
7	Z0	<_Z0>	REAL	回転前の基準点 Z		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
8	X(A)	<_A>	REAL	パラメータ<_MODE>の設定に基づいた 1 番目の回転		
9	Y(B)	<_B>	REAL	パラメータ<_MODE>の設定に基づいた 2 番目の回転		
10	Z(C)	<_C>	REAL	パラメータ<_MODE>の設定に基づいた 3 番目の回転		
11	X1	<_X1>	REAL	回転後の基準点 X		
12	Y1	<_Y1>	REAL	回転後の基準点 Y		
13	Z1	<_Z1>	REAL	回転後の基準点 Z		
14	-または+	<_DIR>	INT	回転軸の移動を開始(初期設定 = -1)		
					-1 =	回転軸 1 と 2 のうち、どちらか小さい値のほうに位置決めする [2]
					+1 =	回転軸 1 と 2 のうち、どちらか大きい値のほうに位置決めする [2]
					0 =	旋回しない(旋回フレームの計算のみ) [1] [3]
15	FR	<_FR_I>	REAL	工具方向の後退の値(inc)、インクレメンタル		
16		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	工具位置合わせ時のベータ値の表示	
					0 =	値
					1 =	矢印

注記

次の転送パラメータが間接的に(パラメータとして)プログラム指令されている場合、入力画面はリセットされません: <_FR>, <_ST>, <_TC>, <_MODE>, <_DIR>

1) 旋回機能がセットアップで設定されている場合のみ選択できます。

2) 回転軸 1 または 2 を基準にした方向が旋回セットアップで設定されている場合に選択可能です。

方向基準が「なし」の場合、選択欄はありません。

3) 旋回の選択「なし」は、SD 55221 ビット 0 でグレーの透過色で表示できます。

旋回「なし」、「負」方向は<_DIR> = 0 と_ST TEN THOUSANDS = 1 に対応しています。

旋回「なし」、「正」方向は<_DIR> = 0 と_ST TEN THOUSANDS = 2 に対応しています。

4) 回転軸 1 または 2 の方向の選択は、方向基準のある回転軸が極位置にある場合(位置データが 0 に等しい)にも発生します。

5) 割り当て例:軸毎の回転、回転順序 ZYX

2 進法:00011011 10 進法:27

軸識別子 XYZ は、NC チャネルのジオメトリ軸に対応しています。XYZ 軸を中心にした個別の回転がおこなえます。たとえば、ZXZ を中心にした回転順序は、1 回の CYCLE800 呼び出しでは行えません。

20.1.34 CYCLE801 - 格子またはフレーム

構文

```
CYCLE801(<_SPCA>, <_SPCO>, <_STA>, <_DIS1>, <_DIS2>, <_NUM1>,
<_NUM2>, <_VARI>, <_UMODE>, <_ANG1>, <_ANG2>, <_HIDE>, <_NSP>,
<_DMODE>)
```

パラメータ

No.	マスクパラメータ	パラメータ パラメータ	データタイプ	意味
1	X0	<_SPCA>	REAL	1 番目の軸の位置決めパターン(格子/フレーム)の基準点(abs)
2	Y0	<_SPCO>	REAL	2 番目の軸の位置決めパターン(格子/フレーム)の基準点(abs)

No.	マスクパラメータ	パラメータ パラメータ	データタイプ	意味		
3	α0	<_STA>	REAL	基本回転角度 (1 番目の軸に対する角度)	< 0 =	右回り
					> 0 =	左回り
4	L1	<_DIS1>	REAL	列の距離(1 番目の軸からの距離、符号なしで入力)		
5	L2	<_DIS2>	REAL	行の距離(2 番目の軸からの距離、符号なしで入力)		
6	N1	<_NUM1>	INT	列数		
7	N2	<_NUM2>	INT	行数		
8		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	位置決めパターン	
					0 =	格子
					1 =	フレーム
				TENS(十の位):	予備	
				HUNDREDS(百の位):	予備	
9		<_UMODE>	INT	予備		
10	αX	<_ANG1>	REAL	1 番目の軸によるせん断角(線が 1 番目の軸に対して斜めに配置されています)		
					< 0 =	右回りの角度 (0~-90°)
					> 0 =	左回りの角度 (0~90°)
11	αY	<_ANG2>	REAL	2 番目の軸によるせん断角(線が 2 番目の軸に対して斜めに配置されています)		
					< 0 =	右回りの角度 (0~-90°)
					> 0 =	左回りの角度 (0~90°)
12		<_HIDE>	STRING [200]	非表示位置 ● 最大 198 文字 ● 連続する位置番号の指定。例:"1,3" (位置 1 と 3 は実行されません)		
13		<_NSP>	INT	予備		

No.	マスクパラメータ	パラメータ パラメータ	データタイプ	意味		
14		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)

20.1.35 CYCLE802 - 任意の位置

構文

```
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>)
```

パラメータ

No.	マスクパラメータ	パラメータ 内部	データタイプ	意味		
1		<_XA>	INT	すべての X 位置の代替値(9 桁の 10 進数値) 10 進数値の桁数は次のとおりです。876543210 (桁位置は穴あけ位置 Xn に対応しています)		
				位置データ:	1 =	アブソリュート(1 番目のプログラム指令位置は常にアブソリュートです)
					2 =	インクレメンタル

No.	マスクパラメータ	パラメータ 内部	データタイプ	意味		
2		<_YA>	INT	すべての Y 位置の代替値(9 桁の 10 進数値) 10 進数値の桁数は次のとおりです。876543210 (桁位置は穴あけ位置 Yn に対応しています)		
				位置データ:	1 =	アブソリュート(1 番目のプログラム指令位置は常にアブソリュートです)
					2 =	インクレメンタル
3	X0	<_X0>	REAL	1 番目の位置 X		
4	Y0	<_Y0>	REAL	1 番目の位置 Y		
5	X1	<_X1>	REAL	2 番目の位置 X		
6	Y1	<_Y1>	REAL	2 番目の位置 Y		
7	X2	<_X2>	REAL	3 番目の位置 X		
8	Y2	<_Y2>	REAL	3 番目の位置 Y		
9	X3	<_X3>	REAL	4 番目の位置 X		
10	Y3	<_Y3>	REAL	4 番目の位置 Y		
11	X4	<_X4>	REAL	5 番目の位置 X		
12	Y4	<_Y4>	REAL	5 番目の位置 Y		
13	X5	<_X5>	REAL	6 番目の位置 X		
14	Y5	<_Y5>	REAL	6 番目の位置 Y		
15	X6	<_X6>	REAL	7 番目の位置 X		
16	Y6	<_Y6>	REAL	7 番目の位置 Y		
17	X7	<_X7>	REAL	8 番目の位置 X		
18	Y7	<_Y7>	REAL	8 番目の位置 Y		
19	X8	<_X8>	REAL	9 番目の位置 X		
20	Y8	<_Y8>	REAL	9 番目の位置 Y		

No.	マスクパラメータ	パラメータ 内部	データタイプ	意味		
21		<_VARI>	INT	加工		
				HUNDREDS(百の位):	(注文生産現場からの呼び出しの場合のみ) (現時点では 0 と 2 のみが評価されます)	
					0 =	主軸をクランプしません
					1 =	G00 または G01 による垂直切り込みの場合のみ主軸をクランプします
					2 =	加工中は常に主軸をクランプします
				THOUSANDS(千の位):	予備	
				TEN THOUSANDS(万の位):	回転軸あり/なしの位置決めパターン – 軸の組み合わせ (<_VARI> HUNDRED THOUSANDS と)	
					0 =	XY (XY のみで回転軸なし、互換性)
					1 =	X、Y または Z と回転軸: XA、YB、ZC (1 つの回転軸と回転軸の回転の中心となるジオメトリ軸)
					2 =	XY と回転軸: XYA、XYB、XYC (1 つの回転軸と 1 番目および 2 番目のジオメトリ軸、TRACYL なし)
				HUNDRED THOUSANDS(十万の位):	回転軸	
					0 =	回転軸なし (XY のみ、互換性)
					1 =	A 軸(X を中心とした回転軸)
					2 =	B 軸 (Y を中心とした回転軸)
					3 =	C 軸(Z を中心とした回転軸)

No.	マスクパラメータ	パラメータ 内部	データタイプ	意味		
				TEN MILLIONS + ONE MILLION(千万 + 百万の位):	回転軸ありの位置決めパターン – オフセット (同じ軸を中心とした複数の回転軸用、インデックスが大きすぎる場合は、1 番目の軸)	
					00 =	1 番目の A、B または C 軸または互換性のため
					01 =	2 番目の A、B または C 軸
					...	
					19 =	20 番目の A、B または C 軸
22		<_UMODE>	INT	クランプする主軸の選択:(注文生産現場からの呼び出しの場合のみ) (ユーザーサイクル CUST_TECHCYC の呼び出し)		
					3 =	主軸のクランプ/解放
					23 =	対向主軸のクランプ/解放
23		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
24		<S_ABA>	INT	すべての AB 位置の代替値(9 桁の 10 進数値) 10 進数値の桁数は次のとおりです。876543210 (桁位置は位置 ABn に対応しています)		
				位置データ:	1 =	アブソリュート(1 番目のプログラム指令位置は常にアブソリュートです)
					2 =	インクレメンタル
25	A0	<S_AB0>	REAL	回転軸ありの位置決めパターンの 1 番目の回転軸位置 (<_VARI>と組み合わせて))		
26	A1	<S_AB1>	REAL	回転軸ありの位置決めパターンの 2 番目の回転軸位置		
27	A2	<S_AB2>	REAL	回転軸ありの位置決めパターンの 3 番目の回転軸位置		

No.	マスクパラメータ	パラメータ 内部	データタイプ	意味
28	A3	<S_AB3>	REAL	回転軸ありの位置決めパターンの 4 番目の回転軸位置
29	A4	<S_AB4>	REAL	回転軸ありの位置決めパターンの 5 番目の回転軸位置
30	A5	<S_AB5>	REAL	回転軸ありの位置決めパターンの 6 番目の回転軸位置
31	A6	<S_AB6>	REAL	回転軸ありの位置決めパターンの 7 番目の回転軸位置
32	A7	<S_AB7>	REAL	回転軸ありの位置決めパターンの 8 番目の回転軸位置
33	A8	<S_AB8>	REAL	回転軸ありの位置決めパターンの 9 番目の回転軸位置

注記

パラメータ X1/Y1/A1~X8/Y8/A8 に必要でない位置は、無視することができます。ただし、<_XA>,<_YA>と<S_ABA>の代替値を 9 つのすべての位置に対して全て与えてください。

位置決めパターン XA、YB または ZC (ジオメトリ軸と回転軸)の場合、位置決めパターンによって移動しない加工平面の軸(G17 の Y と XA)は、サイクル呼び出しの前に位置決めしてください。

20.1.36 CYCLE830 - 深穴ドリル 2

構文

```
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>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<RTP>	REAL	イニシャル点(abs)
2	Z0	<RFP>	REAL	基準点(abs)
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なし)
4	Z1	<_DP>	REAL	最終穴あけ深さ abs/inc (<_AMODE>UNITS を参照)

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
5	d	<FDEP>	REAL	アブソリュートまたはインクレメンタルの切り屑破断/除去のための 1 番目の穴あけ深さ、予備穴あけあり/なしで基準点を基準、またはパイロット穴深さを基準(<_AMODE> TEN THOUSANDS を参照)
6	DF	<_DAM>	REAL	各々の追加切り込みの絶対値/パーセント値、逓減アブソリュート値/パーセント値(<_AMODE> HUNDRED THOUSANDS を参照)
7	DTB	<DTB>	REAL	各穴あけ深さでのドウェル時間(<_AMODE> TENS を参照)
8	DTS	<DTS>	REAL	始点での切り屑除去中のドウェル時間(<_AMODE> HUNDREDS を参照)
9	FD1	<FRF>	REAL	最初の切り込み送り速度のパーセント値(<_AMODE> TEN MILLION を参照)

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
10		<VARI>	INT	加工		
				UNITS(一の位):	切り屑破断/切り屑除去	
					0 =	一回の切削
					1 =	切り屑切断
					2 =	切り屑除去
					3 =	切り屑切断と切り屑除去
				TENS(十の位):	切り屑除去中の後退	
					0 =	パイロット穴深さまで
					1 =	安全距離まで
				HUNDREDS(百の位):	ソフトな最初の切削	
					0 =	なし
					1 =	あり
				THOUSANDS(千の位):	貫通穴あけ	
					0 =	なし
					1 =	あり
				TEN THOUSANDS(万の位):	予備穴あけ/パイロット穴	
					0 =	予備穴あけなし
					1 =	予備穴あけあり
					2 =	パイロット穴あり
				HUNDRED THOUSANDS(十万の位):	後退	
					0 =	パイロット穴深さまで
					1 =	イニシャル点
11	V1	<_MDEP>	REAL	最小インクレメンタル切り込み(逓減パーセント値のみ)		
12	V2	<_VRT>	REAL	各インクレメンタル加工ステップ終了後の後退量(切り屑破断の場合のみ)		
					0 =	初期値 1 mm
					> 0 =	可変 後退距離
13	DT	<_DTD>	REAL	最終穴あけ深さでのドウェル時間(<_AMODE> THOUSANDS を参照)		
14	V3	<_DIS1>	REAL	切り屑除去の場合のみのインクレメンタルのリミット距離(<_AMODE> ONE MILLION を参照)		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
15	FP	<S_FP>	REAL	パイロット穴への移動の送り速度(値としてまたは%単位で)(<S_AMODE2> HUNDREDS と組み合わせて)		
16		<S_SDAC2>	INT	アプローチ時の主軸の回転方向		
					3 =	M3
					4 =	M4
					5 =	M5 (初期設定)
17	SP	<S_SV2>	REAL	一定主軸速度でのアプローチ	(<S_AMODE2> TEN MILLION を参照)	
	V4				定切削速度	
				穴あけ速度の%での主軸速度		
18	F	<S_FB>	REAL	穴あけ送り速度(<S_AMODE2> UNITS を参照)		
19		<_SDAC>	REAL	穴あけ時の主軸の回転方向		
					3 =	M3
					4 =	M4
20	S	<_SV1>	REAL	一定主軸速度での	穴あけ(<S_AMODE2> ONE MILLION を参照)	
	V5				定切削速度	
21	SPOS	<S_SPOS>	REAL	主軸位置決め、M5 でのアプローチの場合のみ		
22	ZA	<S_ZA>	REAL	基準点を基準にしたインクレメンタルの予備穴あけ深さまたはアブソリュート(<S_AMODE3> UNITS を参照)		
23	FA	<S_FA>	REAL	予備穴あけ送り速度(値としてまたは%単位で)(<S_AMODE2> TENS と組み合わせて)		
24	ZP	<S_ZP>	REAL	基準点を基準にしたインクレメンタルのパイロット穴またはアブソリュートまたは穴直径の係数(<S_AMODE3> TENS を参照)		
25	FS	<S_FS>	REAL	最初の切削の送り速度(値としてまたは%単位で)(<S_AMODE2> THOUSANDS と組み合わせて)		
26	ZS1	<S_ZS1>	REAL	一定送り速度での各 1 番目切削の深さ(inc)		
27	ZS2	<S_ZS2>	REAL	送り速度増速での各 1 番目切削の深さ(inc)		
28	N	<S_N>	INT	各切り屑除去の前の切り屑切断の回数		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
29	ZD	<S_ZD>	REAL	最終穴あけ深さを基準にしたインクレメンタルの残りの穴あけ深さまたはアブソリュート(<S_AMODE3> HUNDREDS を参照)		
30	FD	<S_FD>	REAL	残りの穴あけ送り速度(値としてまたは%単位で) (<S_AMODE2> TEN THOUSANDS と組み合わせて)		
31	FR	<S_FR>	REAL	後退送り速度(<S_AMODE2> HUNDRED THOUSANDS と組み合わせて)		
32		<S_SDAC3>	INT	後退のときの主軸の回転方向		
					3 =	M3
					4 =	M4
					5 =	M5
33	SR	<S_SV3>	REAL	後退速度	一定主軸速度(<S_AMODE2> HUNDRED MILLION を参照)	
	V6				定切削速度	
				穴あけ速度の%での主軸速度		
34	クーラント ON	<S_CON>	STRING[10]	クーラントオン、M 命令またはサブプログラム呼び出し		
35	クーラント オフ	<S_COFF>	STRING[10]	クーラントオフ、M 命令またはサブプログラム呼び出し		
36		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	予備	
				TENS(十の位):	先端/シャンクを基準にした穴あけ深さ	
					0 =	先端
					1 =	シャンク

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
37		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	予備	
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):	予備	
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1004)	
					0 =	入力:全て
					1 =	入力:ベーシック

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
38		<_AMODE>	INT	代替モード 1		
				UNITS(一の位):	穴あけ深さ = 最終穴あけ深さ Z1 abs/inc	
					0 =	インクレメンタル
					1 =	アブソリュート
				TENS(十の位):	各穴あけ深さでのドウェル時間 DTB(秒(s)/回転数(rev))	
					0 =	秒(s)
					1 =	回転数(rev)
				HUNDREDS(百の位):	切り屑除去のドウェル時間 DTS(秒(s)/回転(rev))	
					0 =	秒(s)
					1 =	回転数(rev)
				THOUSANDS(千の位):	最終穴あけ深さ DT でのドウェル時間(秒(s)/回転数(rev))	
					0 =	秒(s)
					1 =	回転数(rev)
				TEN THOUSANDS(万の位):	1 番目の穴あけ深さ D (abs/inc)	
					0 =	インクレメンタル
					1 =	アブソリュート
				HUNDRED THOUSANDS(十万の位):	各々の追加切り込みの絶対値/パーセント値(逓減量)	
					0 =	絶対値
					1 =	パーセント値(0.001~100%)
				ONE MILLION(百万の位):	リミット距離 V3 自動/手動	
					0 =	自動(サイクルで計算)
					1 =	手動(プログラム指令値)

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
39		<S_AMODE2>	INT	代替モード 2		
				UNITS(一の位):	UNITS(一の位):穴あけ送り速度 F	
					0 =	F/min
					1 =	F/rev
				TENS(十の位):	予備穴あけ送り速度の評価 FA	
					0 =	穴あけ送り速度の%として
					1 =	F/min
					2 =	F/rev
				HUNDREDS(百の位):	パイロット穴への移動のための送り速度 FP の評価	
					0 =	穴あけ送り速度の%として
					1 =	F/min
					2 =	F/rev
				THOUSANDS(千の位):	最初の切削の送り速度 FS の評価	
					0 =	穴あけ送り速度の%として
					1 =	F/min
					2 =	F/rev
				TEN THOUSANDS(万の位):	貫通穴あけ送り速度 FD の評価	
					0 =	穴あけ送り速度の%として
					1 =	F/min
					2 =	F/rev
				HUNDRED THOUSANDS(十万の位):	後退送り速度 FR	
					0 =	F/min
					1 =	早送り
				ONE MILLION(百万の位):	穴あけ - 主軸速度/切削速度 (S/V5)	
					0 =	一定主軸速度
					1 =	定切削速度
				TEN MILLION(千万の位):	主軸速度/切削速度(SP/V4)でのアプローチ	
					0 =	一定主軸速度
					1 =	定切削速度

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
					2 =	穴あけ速度の%での主軸速度
				HUNDRED MILLIONS(一億の位):	後退 - 主軸速度/切削速度(SR/V6)	
					0 =	一定主軸速度
					1 =	定切削速度
					2 =	穴あけ速度の%での主軸速度
40		<S_AMODE3>	INT	代替モード 3		
				UNITS(一の位):	穴あけ深さ ZA abs/inc	
					0 =	インクレメンタル
					1 =	アブソリュート
				TENS(十の位):	パイロット穴 ZP の深さ	
					0 =	インクレメンタル
					1 =	アブソリュート
					2 =	穴直径の係数
				HUNDREDS(百の位):	残りの穴あけ深さ ZD abs/inc	
					0 =	インクレメンタル
					1 =	アブソリュート
41	ZPV	<S_ZPV>	REAL	パイロット穴深さからのインクリメンタルのリミット距離		

20.1.37 CYCLE832 - 高速設定

構文

```
CYCLE832(<S_TOL>, <S_TOLM>, <S_OTOL>)
```

注記

CYCLE832 により、機械のセットアップ時に必要な工作機械メーカによる調整の仕事が軽減されることはありません。このサイクルは、加工プロセスと NCU 設定(プリコントロール、加々速度一定など)に関連する軸の調整をおこないます。

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1	許容範囲	<S_TOL>	REAL	輪郭の許容範囲 輪郭の許容範囲は、ジオメトリ軸の軸の許容範囲に対応しています。		
2		<S_TOLM>	INT	加工タイプ(用途)		
				UNITS(一の位):		
					0 =	解除
					1 =	仕上げ
					2 =	荒仕上げ
					3 =	荒削り
				TENS(十の位):		
					0 =	互換性 [1]または旋回の許容範囲なし
					1 =	パラメータ<S_OTOL>の旋回許容範囲
				百の位 ... HUNDRED THOUSANDS(十万の位)	割り付けられた 互換性が 理由で	
				ONE MILLION(百万の位):		
					0 =	互換性。最上の金型形成機能が自動的に使用されます。 ● オプション[最上面]が無効です。 ☐ Advanced Surface ● オプション[最上面]が有効です。 ⇒ スムージングを行う最上面
					1 =	スムージングを行わない最上面
					2 =	スムージングを行う最上面

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
3	ORI の許容範囲	<S_OTOL>	REAL	旋回の許容範囲またはタイプの識別子 CYCLE832 ワークの旋回の許容範囲パラメータです。 これは、ダイナミックな方向座標変換(5 軸同時加工など)を備えた機械での高速加工プログラムの実行時に必要です。 パラメータ<S_OTOL>をプログラム指令してください。これは 3 軸機械での工具のオリエンテーションを使用しないプログラムの用途にも適用されます(<S_OTOL> = 1)。

1) 旋回の許容範囲は輪郭の許容範囲にサイクルセッティングデータ SD55441 ～ SD55443 の係数を乗算して得られます。

参照先:

試運転マニュアル ベースソフトウェアと操作ソフトウェア; SINUMERIK Operate(IM9)、「高速設定機能の設定(CYCLE832)」の章

プレーンテキスト入力

サイクル呼び出しを読みやすくするために、パラメータ<S_TOLM>(加工タイプ)をプレーンテキストで入力することもできます。プレーンテキストは言語に依存しません。次の入力が可能です。

_OFF	FOR	0	解除
_FINISH	FOR	1	仕上げ
_SEMIFIN	FOR	2	荒仕上げ
_ROUGH	FOR	3	荒削り
_ORI_FINISH	FOR	11	旋回の許容範囲の入力による仕上げ
_ORI_SEMIFIN	FOR	12	旋回の許容範囲の入力による仕上げ
_ORI_ROUGH	FOR	13	旋回の許容範囲の入力による荒削り
_TOP_SURFACE_SMOOTH_OFF	FOR	1000000	スムージングを行わない最上面

_TOP_SURFACE_SMOOTH_ON	FOR	2000000	スムージングを行う最上面

最上面のプレーンテキストの入力では、プレーンテキストは以下の例に示すように組み合されます。

```
CYCLE832(0.1, _TOP_SURFACE_SMOOTH_OFF+_ORI_FINISH, 1)
```

注記

プレーンテキストは G グループ 59 の機能名称(軌跡補間のダイナミック応答モード)に基づいています。これらのプレーンテキストにより、3 軸機械と多軸方向座標変換(TRAORI)を備えた機械は用途で明確に区別されます。

CYCLE832 の選択解除

CYCLE832 を選択解除する場合は、パラメータ<S_TOL>を 0 で転送してください。

例:CYCLE832(0,0,1)

CYCLE832 の選択解除には、構文 CYCLE832() も使用できます。

例

例 1: 3 軸機械での方向座標変換を使用しない CYCLE832

a) プレーンテキスト入力によるサイクル呼び出し

プログラムコード	コメント
G710	; 単位系はメトリックです。
CYCLE832(0.004,_FINISH,1)	; 次の設定による CYCLE832 呼び出し: 輪郭の許容範囲 = 0.004 mm、加工タイプ:仕上げ
...	; 高速加工プログラムの実行

b) プレーンテキスト入力を使用しないサイクル呼び出し

プログラムコード	コメント
G710	; 上記を参照してください
CYCLE832(0.004,1,1)	; 上記を参照してください
...	; 上記を参照してください

例 2: 5 軸機械での方向座標変換を使用した CYCLE832

a) プレーンテキスト入力によるサイクル呼び出しと選択解除

プログラムコード	コメント
G710	; 単位系はメトリックです。
TRAORI	; 方向座標変換を起動します。
CYCLE832(0.3,_ORI_ROUGH,0.8)	; 次の設定による CYCLE832 呼び出し: 輪郭の許容範囲 = 0.3 mm、加工タイプ:旋回の許容範囲の入力による荒削り; 旋回の許容範囲 = 0.8°
...	; 高速加工プログラムの実行
CYCLE832(0,_OFF,1)	; 輪郭の許容範囲 = 0、 加工タイプ:CYCLE832 の解除、 旋回の許容範囲 = 0°

b) プレーンテキスト入力を使用しないサイクル呼び出しと選択解除

プログラムコード	コメント
G710	; 上記を参照してください
TRAORI	; 上記を参照してください
CYCLE832(0.3,13,0.8)	; 上記を参照してください
...	; 上記を参照してください
CYCLE832(0,0,1)	; 上記を参照してください

20.1.38 CYCLE840 - フローティングチャックによるタッピング

構文

```
CYCLE840(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDR>, <SDAC>,
<ENC>, <MPIT>, <PIT>, <_AXN>, <_PITA>, <_TECHNO>, <_PITM>,
<_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<RTP>	REAL	イニシャル点(abs)
2	Z0	<RFP>	REAL	基準点(abs)
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<DP>	REAL	穴あけ深さ(abs)、<_AMODE>を参照
5	Z1	<DPR>	REAL	穴あけ深さ(inc)、<_AMODE>を参照

No.	マスクパラメータ	内部パラメータ	データタイプ	意味			
6	DT	<DTB>	REAL	穴あけ深さでのドウェル時間(s)/後退後の安全距離、<ENC>を参照			
7		<SDR>	INT	後退時の回転方向			
8	SDE	<SDAC>	INT	サイクル終了後の回転方向			
9		<ENC>	INT	主軸エンコーダを搭載したタッピング(G33)/主軸エンコーダを搭載しないタッピング(G63)			
					0 =	主軸エンコーダを搭載	- <MPIT>/ <PIT> からのピッチ - DT なし
					20 =	主軸エンコーダを搭載	- <MPIT>/ <PIT> からのピッチ - 安全距離への後退後の DT あり
					11 =	主軸エンコーダを搭載しない	- <MPIT>/ <PIT> からのピッチ - 穴あけ深さでの DT あり
					1 =	主軸エンコーダを搭載しない	- プログラム指令送り速度からのピッチ - 穴あけ深さ DT による (送り速度 = 回転速度・ピッチ)
10		<MPIT>	REAL	「ISO メトリック」のねじサイズのみ(ピッチは運転時に内部で計算されます) 値の範囲:3 ～ 48 (M3 ～ M48 の場合)、<PIT>の代わり			
11		<PIT>	REAL	数値で表したピッチ、単位については<_PITA>を参照 値の範囲: > 0、MPIT の代わり			
12		<_AXN>	INT	穴あけ軸	0 =	3 番目のジオメトリ軸	
					1 =	1 番目のジオメトリ軸	
					2 =	2 番目のジオメトリ軸	
					≥ 3 =	3 番目のジオメトリ軸	

No.	マスクパラメータ	内部パラメータ	データタイプ	意味			
13		<_PITA>	INT	ピッチ単位(<PIT>と<MPIT>の評価)			
					0 =	ピッチ(mm)	- 評価<MPIT>/<PIT>
					1 =	ピッチ(mm)	- 評価<PIT>
					2 =	ピッチ(TPI)	- <PIT>の評価(ねじ山数/inch)
					3 =	ピッチ(inch)	- 評価<PIT>
					4 =	MODULUS	- 評価<PIT>
14		<_TECHNO>	INT	用途 [1]			
				UNITS(一の位):	イグザクトストップ動作		
					0 =	サイクル呼び出し前と同じイグザクトストップ動作が有効	
					1 =	イグザクトストップ G601	
					2 =	イグザクトストップ G602	
					3 =	イグザクトストップ G603	
				TENS(十の位):	フィードフォワード制御		
					0 =	サイクル呼び出し前と同じフィードフォワード制御あり/なしが有効	
					1 =	フィードフォワード制御あり FFWON	
					2 =	フィードフォワード制御なし FFWOF	
15		<_PITM>	STRING[15]	ピッチ入力のマーカとなる文字列 [2]			
16		<_PTAB>	STRING[5]	ねじテーブルの文字列(""、「ISO」、「BSW」、「BSP」、「UNC」)[2]			
17		<_PTABA>	STRING[20]	ねじテーブルからの選択のための文字列(例:「M 10」、「M 12」、...)[2]			
18		<_GMODE>	INT	予備			

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
19		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	予備	
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):	互換性モード(または再コンパイル入力画面のみ)、MD 52216 ビット 0 = 1 の場合 [1]	
					0 =	テクノロジパラメータが表示されます(互換性):TECHNO パラメータが有効
					1 =	テクノロジパラメータは表示されません:「サイクル呼び出し前と同じ」プログラム値が有効
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1004)	
					0 =	入力:全て
					1 =	入力:簡易
20		<_AMODE>	INT	代替モード		
				UNITS(一の位):	穴あけ深さ Z1 (abs/inc)	
					0 =	互換性、 <DP>/<DPR>のプログラム指令から
					1 =	インクレメンタル
					2 =	アブソリュート

[1] テクノロジフィールドは、SD52216 $MCS_FUNCTION_MASK_DRILL の設定データに応じて非表示になる場合があります。

[2] パラメータ 15、16、17 は、入力画面ねじテーブルでのねじの選択にのみ使用されます。ねじテーブルは、サイクル運転時にサイクル定義を介してアクセスすることはできません。

20.1.39 CYCLE899 - オープン溝のフライス削り

構文

```
CYCLE899(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_LENG>, <_WID>, <_PA>,
<_PO>, <_STA>, <_MID>, <_MIDA>, <_FAL>, <_FALD>, <_FFP1>,
<_CDIR>, <_VARI>, <_GMODE>, <_DMODE>, <_AMODE>, <_UMODE>, <_FS>,
<_ZFS>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1	RP	<_RTP>	REAL	イニシャル点(abs)		
2	Z0	<_RFP>	REAL	工具軸の基準点(abs)		
3	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)		
4	Z1	<_DP>	REAL	溝の深さ(abs/inc)、<_AMODE>を参照		
5	L	<_LENG>	REAL	溝の長さ(inc)		
6	W	<_WID>	REAL	溝の幅(inc)		
7	X0	<_PA>	REAL	基準点/開始点、1 番目の軸(abs)		
8	Y0	<_PO>	REAL	基準点/開始点、2 番目の軸(abs)		
9	α0	<_STA>	REAL	1 番目の軸を基準にした回転角度		
10	DZ	<_MID>	REAL	最大切り込み深さ(inc)、渦巻きフライス削りの場合のみ		
11	DXY	<_MIDA>	REAL	最大平面切り込み、<_AMODE>を参照		
12	UXY	<_FAL>	REAL	平面の仕上げ代		
13	UZ	<_FALD>	REAL	深さの仕上げ代		
14	F	<_FFP1>	REAL	送り速度		
15		<_CDIR>	INT	加工方向		
				UNITS(一の位):		
					0 =	ダウンカット
					1 =	アップカット
					4 =	点滅

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
16		<_VARI>	INT	加工タイプ		
				UNITS(一の位):		
					1 =	荒削り
					2 =	仕上げ
					3 =	底面の仕上げ
					4 =	端面の仕上げ
					5 =	荒仕上げ
					6 =	面取り
				TENS(十の位):	予備	
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):		
					1 =	渦巻きフライス加工
					2 =	プランジ切削
17		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	予備	
				TENS(十の位):	予備	
				HUNDREDS(百の位):	加工または開始点の計算のみのどちらかを選択します。	
					1 =	通常の加工
				THOUSANDS(千の位):	中心/端面による位置指令方法	
					0 =	中心による位置指令方法
					1 =	端面を使用した「左側」の位置指令方法(1 番目の軸の「-」方向)
					2 =	端面を使用した「右側」の位置指令方法(1 番目の軸の「+」方向)

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
18		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	---	予備
				HUNDREDS(百の位):	---	予備
				THOUSANDS(千の位):	---	予備
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1004)	
					0 =	入力:全て
					1 =	入力:簡易
19		<_AMODE>	INT	代替モード		
				UNITS(一の位):	溝の深さ Z1	
					0 =	アブソリュート
					1 =	インクレメンタル
				TENS(十の位):	平面切り込み(<_MIDA>) DXY の単位	
					0 =	mm
					1 =	工具直径の%
				HUNDREDS(百の位):	面取りの切り込み深さ ZFS	
					0 =	アブソリュート
					1 =	インクレメンタル
20		<_UMODE>	INT	予備		
21	FS	<_FS>	REAL	面取り幅(inc)		
22	ZFS	<_ZFS>	REAL	面取りの切り込み深さ(工具先端) (abs/inc)、<_AMODE>を参照		

20.1.40 CYCLE930 - 溝削り

構文

```
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>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	X0	<_SPD>	REAL	平面軸の基準点(常に直径)
2	Z0	<_SPL>	REAL	長手軸の基準点
3	B1	<_WIDG>	REAL	溝の底面の幅
4	B2	<_WIDG2>	REAL	溝の上部の幅(インタフェースのみ)
5	T1	<_DIAG>	REAL	基準点の溝の深さ abs で長手方向の加工の場合 = 直径、その他の場合は inc
6	T2	<_DIAG2>	REAL	基準点の反対側の溝の深さ(インタフェースのみ)、 abs で長手方向の加工の場合 = 直径、その他の場合は inc
7	α0	<_STA>	REAL	傾斜角(-180 ≤ <_STA> ≤ 180)
8	α1	<_ANG1>	REAL	基準点で特定された溝の側面の側面角度 1 (0 ≤ <_ANG1> < 90)
9	α2	<_ANG2>	REAL	基準点の反対側の側面角度 2 (0 ≤ <_ANG2> < 90)
10	R1/FS1	<_RCO1>	REAL	丸み付け半径または面取り幅 1、基準点の外部
11	R2/FS2	<_RCI1>	REAL	丸み付け半径または面取り幅 2、基準点の内部
12	R3/FS3	<_RCI2>	REAL	丸み付け半径または面取り幅 3、基準点の反対側の内部
13	R4/FS4	<_RCO2>	REAL	丸み付け半径または面取り幅 4、基準点の反対側の外部
14	U	<_FAL>	REAL	X と Z の仕上げ代、<_VARI> (TEN THOUSANDS)を参照(符号なしで入力)

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
15	D	<_IDEP1>	REAL	切り込み時の最大切り込み深さ(符号なしで入力)		
					0 =	1 回目の切削は直接深さいっぱいまで
					> 0 =	1 回目の切削<_IDEP1>、2 回目の切削 2 · <_IDEP1>など
16	SC	<_SDIS>	REAL	安全距離(符号なしで入力)		
17		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	予備	
				TENS(十の位):	加工プロセス	
					1 =	荒削り
					2 =	仕上げ
					3 =	荒削りと仕上げ
				HUNDREDS(百の位):	長手方向/径方向の外部/内部 +Z/+Z と+X/-X の位置決め	
					1 =	長手方向/外部 +Z
					2 =	径方向/内部 -X
					3 =	長手方向/内部 +Z
					4 =	径方向/内部 +X
					5 =	長手方向/外部 -Z
					6 =	径方向/外部 -X
					7 =	長手方向/内部 -Z
					8 =	径方向/外部 +X
				THOUSANDS(千の位):	基準点の位置	
					0 =	上側の基準点
					1 =	下側の基準点
				TEN THOUSANDS(万の位):	仕上げ代の仕様を定義します	
					0 =	輪郭に平行の仕上げ代 U
					1 =	UX と UZ 仕上げ代を分離

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
18		<_DN>	INT	工具の 2 番目の刃先の D 番号		
					> 0 =	溝削り工具の 2 番目の刃先の工具オフセットの D 番号
					0 =	2 番目の刃先はプログラム指令なし
19	N	<_NUM>	INT	溝の数(0 = 1 つの溝)		
20	DP	<_DBH>	REAL	溝間の距離(<_NUM> > 1 の場合にのみ必要)		
21	F	<_FF1>	REAL	送り速度		
22		<_NR>	INT	溝の形状の識別は、形状選択用の垂直ソフトキーに対応しています。		
					0 =	面取り/丸み付けなしの 90° の側面
					1 =	面取り/丸み付けあり(α0 なし)の傾斜側面
					2 =	1 と同じ、ただしテーパ上(α0 あり)
23	UX	<_FALX>	REAL	X 軸の仕上げ代、<_VARI> (TEN THOUSANDS)を参照(符号なしで入力)		
24	UZ	<_FALZ>	REAL	Z 軸の仕上げ代、<_VARI> (TEN THOUSANDS)を参照(符号なしで入力)		
25		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
26		<_AMODE>	INT	代替モード		
				UNITS(一の位):	溝の上部の位置指令方法(インタフェースのみ)	
					0 =	基準点
					1 =	基準点の反対側
				TENS(十の位):	深さ	
					0 =	アブソリュート
					1 =	インクレメンタル
				HUNDREDS(百の位):	幅の位置指令方法(インタフェースのみ)	
					0 =	外径(上部)
					1 =	内径(底面)
				THOUSANDS(千の位):	丸み付け/面取り 1 (<_RCO1>)	
					0 =	半径
					1 =	面取り
				TEN THOUSANDS(万の位):	丸み付け/面取り 2 (<_RCI1>)	
					0 =	半径
					1 =	面取り
				HUNDRED THOUSANDS(十万の位):	丸み付け/面取り 3 (<_RCI2>)	
					0 =	半径
					1 =	面取り
				ONE MILLION(百万の位):	丸み付け/面取り 4 (<_RCO2>)	
					0 =	半径
					1 =	面取り

20.1.41 CYCLE940 - アンダーカット形状

CYCLE940 サイクルを使用して、さまざまなアンダーカットをプログラム指令できます。場合によっては、パラメータ設定が大きく異なることがあります。

表の追加列で、どのアンダーカットタイプにどのパラメータが必要かを示します。これはサイクル画面の垂直選択ソフトキーに対応します。

- E:アンダーカット形状 E
- F:アンダーカット形状 F
- A-D:DIN 規格ねじのアンダーカット(形状 A~D)
- T:ねじのアンダーカット(形状は自由に定義)

構文

```
CYCLE940(<_SPD>, <_SPL>, <_FORM>, <_LAGE>, <_SDIS>, <_FFP>,
<_VARI>, <_EPD>, <_EPL>, <_R1>, <_R2>, <_STA>, <_VRT>, <_MID>,
<_FAL>, <_FALX>, <_FALZ>, <_PITI>, <_PTAB>, <_PTABA>, <_DMODE>,
<_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	形状のプログラミング				意味	
				E	F	A-D	T		
1	X0	<_SPD>	REAL	X	X	X	X	平面軸の基準点(常に直径)	
2	Z0	<_SPL>	REAL	X	X	X	X	長手軸の基準点(abs)	
3	FORM	<_FORM>	CHAR	X	X	X	X	アンダーカットの形状(大文字、例:「T」) アンダーカット値を選択する選択テーブル	
									A = 外径、DIN76 基準、A = 通常
									B = 外径、DIN76 基準、B = ショート
									C = 内径、DIN76 基準、C = 通常
									D = 内径、DIN76 基準、D = ショート
									E = DIN509 基準
									F = DIN509 基準
									T= 自由形状

No.	マスクパラメータ	内部パラメータ	データタイプ	形状のプログラミング				意味		
4	POSITION	<_LAGE>	INT	X	X	X	X	アンダーカットの位置(Z に平行)	0 =	外径 +Z: ____|
									1 =	外径 -Z: |____/
									2 =	内径 +Z: /-----|
									3 =	内径 -Z: |-----\
5	SC	<_SDIS>		X	X	X	X	安全距離(inc)		
6	F	<_FFP>		X	X	X	X	加工送り速度(mm/rev)		
7		<_VARI>	INT	-	-	X	X	加工タイプ		
								UNITS(一の位):	加工タイプ	
									1 =	荒削り
									2 =	仕上げ
									3 =	荒削り + 仕上げ
								TENS(十の位):	加工方法	
									0 =	輪郭に平行
									1 =	長手方向
				アンダーカット形状 E と F は常に、仕上げと同様に 1 回のパスで加工されます。						
8	X1	<_EPD>		X	X	-	-	仕上げ代 X (abs/inc)、<_AMODE>を参照		
				-	-	-	X	アンダーカットの深さ(abs/inc)、<_AMODE>を参照		
9	Z1	<_EPL>		-	X	-	-	仕上げ代 Z		
				-	-	-	X	アンダーカットの幅(abs/inc)、<_AMODE>を参照		
10	R1	<_R1>		-	-	-	X	斜面の丸み付け半径		
11	R2	<_R2>		-	-	-	X	コーナの丸み付け半径		
12	α	<_STA>		-	-	X	X	切り込み角度		
13	VX	<_VRT>		X	X	-	-	横送り X (abs/inc)、<_AMODE>を参照		
				-	-	X	X	仕上げ時の横送り X (abs/inc)、<_AMODE>を参照		
14	d	<_MID>		-	-	X	X	深さ切り込み		
15	U	<_FAL>		-	-	X	X	輪郭に平行の仕上げ代、<_AMODE>を参照		
16	UX	<_FALX>		-	-	X	X	仕上げ代 X		
17	UZ	<_FALZ>		-	-	X	X	仕上げ代 Z		

No.	マスクパラメータ	内部パラメータ	データタイプ	形状のプログラミング	意味
18	P	<_PITI>	INT	- - X -	ピッチ、形状 A~D を選択。M1 ... M68 に対応。
					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 - -	半径/深さ、形状 E、F を選択。
					0 = 0.6 x 0.3 1 = 1.0 x 0.4 2 = 1.0 x 0.2 3 = 1.6 x 0.3 4 = 2.5 x 0.4 5 = 4.0 x 0.5 6 = 0.4 x 0.2 7 = 0.6 x 0.2 8 = 0.1 x 0.1 9 = 0.2 x 0.1
19		<_PTAB>	STRING [5]		ねじテーブルの文字列(""、「ISO」、「BSW」、「BSP」、「UNC」) (インタフェースのみ)
20		<_PTABA>	STRING [20]		ねじテーブルからの選択のための文字列 (例: 「M10」、「M12」、...) (インタフェースのみ)
21		<_DMODE>	INT		表示モード
				X X X X	UNITS(一の位): 加工平面 G17/G18/G19
					0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 = G17 (サイクルでのみ有効)
					2 = G18 (サイクルでのみ有効)
					3 = G19 (サイクルでのみ有効)

No.	マスクパラメータ	内部パラメータ	データタイプ	形状のプログラミング				意味		
22		<_AMODE>	INT					代替モード		
				X	X	-	X	UNITS(一の位):	パラメータ<_EPD>仕上げ代 X またはアンダーカットの深さ	
									0 =	アブソリュート(常に直径)
									1 =	インクレメンタル
				X	X	-	X	TENS(十の位):	パラメータ<_EPL> 仕上げ代 Z またはアンダーカットの幅	
									0 =	アブソリュート
									1 =	インクレメンタル
				X	X	X	X	HUNDREDS(百の位):	パラメータ<_VRT>横送り X	
									0 =	アブソリュート(常に直径)
									1 =	インクレメンタル
				-	-	X	X	THOUSANDS(千の位):	仕上げ代	
									0 =	輪郭に平行の仕上げ代 (<_FAL>)
									1 =	個別の加工仕上げ代 (<_FALX>/<_FALZ>)

20.1.42 CYCLE951 - 荒削り

構文

```
CYCLE951(<_SPD>, <_SPL>, <_EPD>, <_EPL>, <_ZPD>, <_ZPL>, <_LAGE>,
<_MID>, <_FALX>, <_FALZ>, <_VARI>, <_RF1>, <_RF2>, <_RF3>,
<_SDIS>, <_FF1>, <_NR>, <_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	X0	<_SPD>	REAL	基準点(abs、常に直径)
2	Z0	<_SPL>	REAL	基準点(abs)

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
3	X1	<_EPD>	REAL	終点		
4	Z1	<_EPL>	REAL	終点		
5	XM α1 α2	<_ZPD>	REAL	中間点、<_DMODE> (TENS)を参照		
6	ZM α1 α2	<_ZPL>	REAL	中間点、<_DMODE> (TENS)を参照		
7	位置	<_LAGE>	INT	切削コーナの位置	0 =	外部/背面
					1 =	外部/前面
					2 =	内部/背面
					3 =	内部/前面
8	d	<_MID>	REAL	切り込み時の最大切り込み深さ		
9	UX	<_FALX>	REAL	X の仕上げ代		
10	UZ	<_FALZ>	REAL	Z の仕上げ代		
11		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	座標系の切削方向(長手方向または径方向)	
					1 =	長手方向
					2 =	正面
				TENS(十の位):		
					1 =	最終加工代まで荒削り
					2 =	仕上げ
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):	予備	
				TEN THOUSANDS(万の位):	予備	
12	R1/FS1	<_RF1>	REAL	丸み付け半径または面取り幅 1、<_AMODE> (TEN THOUSANDS)を参照		
13	R2/FS2	<_RF2>	REAL	丸み付け半径または面取り幅 2、<_AMODE> (HUNDRED THOUSANDS)を参照		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
14	R3/FS3	<_RF3>	REAL	丸み付け半径または面取り幅 3、<_AMODE> (ONE MILLION) を参照		
15	SC	<_SDIS>	REAL	安全距離		
16	F	<_FF1>	REAL	荒削り/仕上げの送り速度		
17		<_NR>	INT	切削タイプの識別(形状選択用の垂直ソフトキーに対応)		
					0 =	切削 1、面取り/丸み付けなしの 90°コーナ
					1 =	切削 2、面取り/丸み付けありの 90°コーナ
					2 =	切削 3、面取り/丸み付けありの任意のコーナ
18		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	入力<_ZPD>/<_ZPL>の形式	
					0 =	Xm/Zm
					1 =	Xm/α1
					2 =	Xm/α2
					3 =	α1/Zm
					4 =	α2/Zm
					5 =	α1/α2

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
21		<_AMODE>	INT	代替モード		
				UNITS(一の位):	X の中間点	
					0 =	アブソリュート、径方向軸の直径の値
					1 =	インクレメンタル、径方向軸の半径の値
				TENS(十の位):	Z の中間点	
					0 =	アブソリュート
					1 =	インクレメンタル
				HUNDREDS(百の位):	X の終点	
					0 =	アブソリュート、径方向軸の直径の値
					1 =	インクレメンタル、径方向軸の半径の値
				THOUSANDS(千の位):	Z の終点	
					0 =	アブソリュート
					1 =	インクレメンタル
				TEN THOUSANDS(万の位):	丸み付け/面取り 1	
					0 =	半径
					1 =	面取り
				HUNDRED THOUSANDS(十万の位):	丸み付け/面取り 2	
					0 =	半径
					1 =	面取り
				ONE MILLION(百万の位):	丸み付け/面取り 3	
					0 =	半径
					1 =	面取り

20.1.43 CYCLE952 - 輪郭の溝削り

構文

```
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>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	PRG	<_PRG>	STRING[100]	切削プログラムの名称
2	CON	<_CON>	STRING[100]	素材の更新済みの輪郭が呼び出されるプログラムの名称(削り残し加工時)
3	CONR	<_CONR>	STRING[100]	素材の更新済みの輪郭(<_AMODE> TEN THOUSANDS を参照)が書き込まれるプログラムの名称

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
4		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	切削のタイプ	
					1 =	長手方向
					2 =	正面
					3 =	輪郭に平行
				TENS(十の位):	製造工程(<_GMODE> HUNDREDS を参照)	
					1 =	荒削り
					2 =	仕上げ
					3 =	予備
					4 =	荒削り、2 個のチャネル
					5 =	仕上げ、2 個のチャネル
				HUNDREDS(百の位):	加工方向	
					1 =	加工方向 X -
					2 =	加工方向 X +
					3 =	加工方向 Z -
					4 =	加工方向 Z +
				THOUSANDS(千の位):	切り込み方向	
					1 =	外部 X -
					2 =	内部 X +
					3 =	前面 Z -
					4 =	背面 Z +
				TEN THOUSANDS(万の位):	仕上げ代の仕様を定義します	
					0 =	UX と UZ 仕上げ代を分離
					1 =	輪郭に平行の仕上げ代 U
				HUNDRED THOUSANDS(十万の位):	丸み付け	
					0 =	互換性、自動丸み付け
					1 =	輪郭での丸み付けあり
					2 =	丸み付けなし
					3 =	自動丸み付け

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
				ONE MILLION(百万の位):	レリーフカット	
					0 =	溝加工時に位置は評価されません - 削り残しと溝旋削 - 削り残し
					1 =	レリーフカットの加工
					2 =	レリーフカットの加工なし
				TEN MILLION(千万の位):	回転中心の背面/前面	
					0 =	回転中心の前面で加工
					1 =	予備
5	F	<_F>	REAL	荒削り/仕上げの送り速度		
	FZ			横軸の溝旋削の切り込み		
6	FR	<_FR>	REAL	レリーフカット切り込み時の送り速度、荒削り		
	FX			縦軸の溝旋削の切り込み		
7	RP	<_RP>	REAL	内部加工用のイニシャル点(abs、常に直径)		
8	d	<_D>	REAL	荒削り切り込み(<_AMODE> UNITS を参照)		
9	Dx	<_DX>	REAL	X 切り込み(<_AMODE> UNITS を参照)		
10	DZ	<_DZ>	REAL	Z 切り込み(<_AMODE> UNITS を参照)		
11	UX	<_UX>	REAL	仕上げ代 X、(<_VARI> TEN THOUSANDS を参照)		
12	UZ	<_UZ>	REAL	仕上げ代 Z、(<_VARI> TEN THOUSANDS を参照)		
13	U	<_U>	REAL	輪郭に平行の仕上げ代、(<_VARI> TEN THOUSANDS を参照)		
14	U1	<_U1>	REAL	仕上げ時の追加仕上げ代(<_AMODE> THOUSANDS を参照)		
15	BL	<_BL>	INT	素材の定義	1 =	仕上げ代付きの円筒
					2 =	仕上げ部の輪郭の仕上げ代
					3 =	素材の輪郭が指定されています
16	XD	<_XD>	REAL	素材 X の定義(<_AMODE> HUNDRED THOUSANDS を参照)		
17	ZD	<_ZD>	REAL	素材 Z の定義(<_AMODE> ONE MILLION を参照)		
18	XA	<_XA>	REAL	限界値 1 X (abs、常に直径)		
19	ZA	<_ZA>	REAL	限界値 1 Z (abs)		
20	XB	<_XB>	REAL	限界値 2 X (<_AMODE> TEN MILLIONS を参照)		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
21	ZB	<_ZB>	REAL	限界値 2 Z (<_AMODE> HUNDRED MILLIONS を参照)		
22	XDA	<_XDA>	REAL	前面の 1 番目の溝位置の溝加工限界値 1(abs、常に直径)		
23	XDB	<_XDB>	REAL	前面の 1 番目の溝位置の溝加工限界値 2(abs、常に直径)		
24	N	<_N>	INT	溝の数		
25	DP	<_DP>	REAL	溝の間の距離	直線溝:Z 軸に平行	
					径方向の溝:X 軸に平行	
26	Di	<_DI>	REAL	切り込みの中断のための距離	0 =	中断なし
					> 0 =	中断あり
27	SC	<_SC>	REAL	障害物を回避するための安全距離、インクレメンタル		
28	D2	<_DN>	INT	工具の 2 番目の刃先の D 番号。プログラムされていない場合 ⇒ D+1		

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
29		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	予備	
				TENS(十の位):	予備	
				HUNDREDS(百の位):	加工または開始点の計算のみのどちらかを選択します。	
					0 =	通常の加工(互換性モードは不要)
					1 =	通常の加工
					2 =	開始点を計算 - 加工なし (ShopMill/ShopTurn からの呼び出しのみ)
				THOUSANDS(千の位):	限界値	
					0 =	なし
					1 =	あり
				TEN THOUSANDS(万の位):	限界値 1 X の入力	
					0 =	なし
					1 =	あり
				HUNDRED THOUSANDS(十万の位):	限界値 2 X の入力	
					0 =	なし
					1 =	あり
				ONE MILLION(百万の位):	限界値 1 Z の入力	
					0 =	なし
					1 =	あり
				TEN MILLION(千万の位):	限界値 2 Z の入力	
					0 =	なし
					1 =	あり

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
30		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				TENS(十の位):	テクノロジモード	
					1 =	輪郭に沿った切削
					2 =	輪郭の溝削り
					3 =	溝旋削
				HUNDREDS(百の位):	削り残し除去加工	
					0 =	なし
					1 =	あり
				THOUSANDS(千の位):	---	予備
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1004)	
					0 =	入力:全て
					1 =	入力:簡易

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
31		<_AMODE>	INT	代替モード		
				UNITS(一の位):	切り込み方法	
					0 =	輪郭に平行した切削時の DX と DZ 切り込み
					1 =	D 切り込み
				TENS(十の位):	切り込み方法	
					0 =	可変の切削深さ(90 ... 100%)
					1 =	一定の切削深さ
				HUNDREDS(百の位):	分割切削	
					0 =	均一
					1 =	端面に割り当て
				THOUSANDS(千の位):	輪郭仕上げ代 U1 の選択、二重仕上げ	
					0 =	なし
					1 =	あり
				TEN THOUSANDS(万の位):	素材の選択の更新	
					0 =	なし
					1 =	あり
				HUNDRED THOUSANDS(十万の位):	素材 XD の仕上げ代の選択	
					0 =	アブソリュート、径方向軸の直径の値
					1 =	インクレメンタル、径方向軸の半径の値
				ONE MILLION(百万の位):	素材 ZD の仕上げ代の選択	
					0 =	アブソリュート
					1 =	インクレメンタル
				TEN MILLION(千万の位):	限界値 2 XB の選択	
					0 =	アブソリュート、径方向軸の直径の値
					1 =	インクレメンタル、径方向軸の半径の値

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
				HUNDRED MILLION(一億の位):	限界値 2 ZB の選択	
					0 =	アブソリュート
					1 =	インクレメンタル
				ONE BILLION(一兆の位):		
					0 =	マスタチャネル
					1 =	スレーブチャネル
32		<_PK>	INT	機械で使用できるチャネルが 2 個より多い場合は、協調チャネルの番号		
33	DCH	<_DCH>	REAL	チャネルオフセット		
34	FS	<_FS>	REAL	全て加工時の仕上げ送り速度		

20.1.44 CYCLE4071 - 反転点での切り込みによる長手方向の研削

構文

```
CYCLE4071(<S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>, <S_H>,
<S_A1>, <S_A2>)
```

パラメータ

No.	パラメータ	データタイプ	意味
1	<S_A>	REAL	開始位置での切り込み深さ
2	<S_B>	REAL	終了位置での切り込み深さ
3	<S_W>	REAL	研削幅
4	<S_U>	REAL	スパークアウト時間
5	<S_I>	REAL	切り込みの送り速度
6	<S_K>	REAL	径方向の切り込みの送り速度
7	<S_H>	INT	繰り返し回数
8	<S_A1>	AXIS	切り込み軸(任意選択)または 1 番目のジオメトリ軸
9	<S_A2>	AXIS	揺動軸(任意選択)または 2 番目のジオメトリ軸

機能

このサイクルは<S_H>切り込みの実行に使用されます。切り込み深さは開始位置と終了位置で変えることができます。切り込み間で接線方向の移動があります。

手順

1. 揺動軸の現在の位置でサイクルを開始します。
2. 切り込み軸を開始位置 P1<S_A>の切込み深さまで、切り込み P5 <S_I>用の送り速度で移動します。
3. スパークアウト時間 P4 <S_U>のスパークアウト。
4. 研削幅 P3<S_W>の揺動軸を移動軌跡として、径方向の切込みの送り速度 P6 <S_K>で移動します。
5. 切り込み軸を終了位置 P2<S_B>の切込み深さまで、切り込み P5 <S_I>用の送り速度で移動します。
6. スパークアウト時間 P4 <S_U>のスパークアウト。
7. 研削幅 P4<S_A>の揺動軸を移動軌跡として、径方向の切込みの送り速度 P6 <S_K>で開始点まで移動します。

この手順はシングルブロックによって中断することはできません。

この手順はプログラムされた繰り返し回数 P7(<S_H>)に従って繰り返されます。

例

以下のサイクルパラメータによる 2 つの揺動移動の実行

- 開始位置での切り込み深さ:0.02 mm
- 終了位置での切り込み深さ:0.01 mm
- ストローク：100 mm
- スパークアウト時間:1 秒

- 切り込みの送り速度:1 mm/min
- 径方向の送り速度:1000 mm/min
- 繰り返し回数:2
- 揺動軸と切り込み軸:標準ジオメトリ軸

プログラムコード

```
N10 T1 D1
N20 CYCLE4071(0.02,0.01,100,1,1,1000,2)
N30 M30
```

20.1.45 CYCLE4072 - 反転点での切り込みによる長手方向の研削とキャンセル信号

構文

```
CYCLE4072(<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>,
<S_H>, <S_A1>, <S_A2>)
```

パラメータ

No.	パラメータ	データタイプ	意味
1	<S_GAUGE>	STRING	切り込みのキャンセル条件: 1. 高速入力の回数 2. 論理式
2	<S_A>	REAL	開始位置での切り込み深さ
3	<S_B>	REAL	終了位置での切り込み深さ
4	<S_W>	REAL	研削幅
5	<S_U>	REAL	スパークアウト時間
6	<S_I>	REAL	切り込みの送り速度
7	<S_K>	REAL	径方向の切り込みの送り速度
8	<S_H>	INT	繰り返し回数
9	<S_A1>	AXIS	切り込み軸(任意選択)または 1 番目のジオメトリ軸
10	<S_A2>	AXIS	揺動軸(任意選択)または 2 番目のジオメトリ軸

機能

このサイクルは、外部キャンセル信号を考慮した<S_H>切り込みの実行に使用されます。切り込み深さは開始位置と終了位置で変えることができます。切り込み間で接線方向の移動があります。キャンセル条件が満たされたときに、深さ切り込みがキャンセルされます。深さ切り込みのキャンセル後、常に完全なストロークが 1 回実行されます。

手順

1. 揺動軸の現在の位置でサイクルを開始します。
2. 切り込み軸を開始点 P2<S_A>の切込み深さまで、切り込み P6 <S_I>用の送り速度で移動します。
3. スパークアウト時間 P5 <S_U>のスパークアウト。
4. 研削幅 P4<S_W>の揺動軸を移動軌跡として、径方向の切込みの送り速度 P7 <S_K>で移動します。
5. 切り込み軸を終了位置 P3<S_B>の切込み深さまで、切り込み P6 <S_I>用の送り速度で移動します。
6. スパークアウト時間 P5 <S_U>のスパークアウト。
7. 研削幅 P4<S_W>の揺動軸を移動軌跡として、径方向の切込みの送り速度 P7 <S_K>で開始点まで移動します。
8. キャンセルなし:上に説明した手順が、プログラムされた繰り返し回数 P7(<S_H>)に達するまで繰り返されます。
   キャンセルあり:次の開始点に達すると加工が停止します。

この手順はシングルブロックによって中断することはできません。

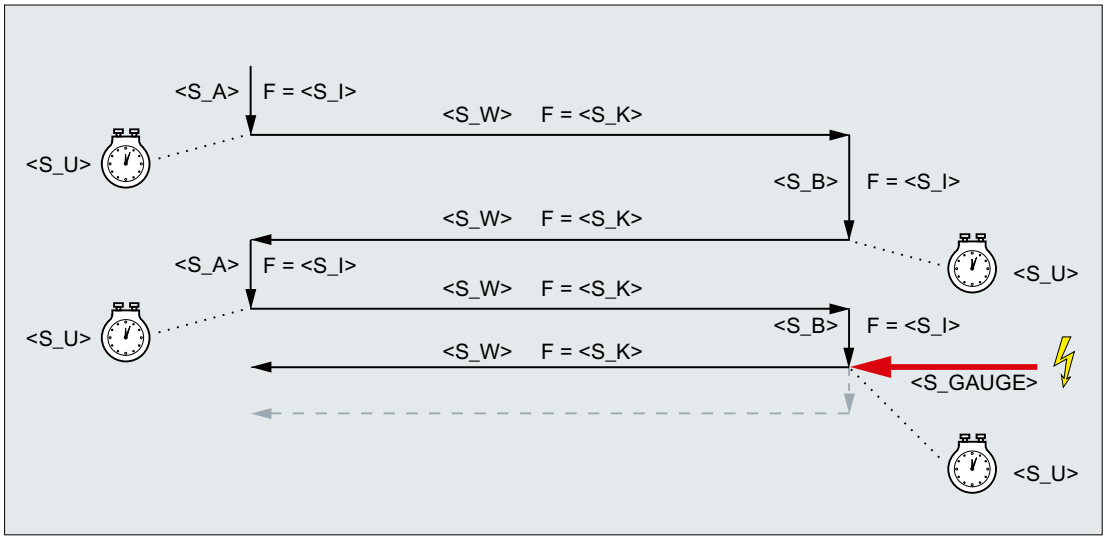


図 20-1　終了位置での切り込みのキャンセル

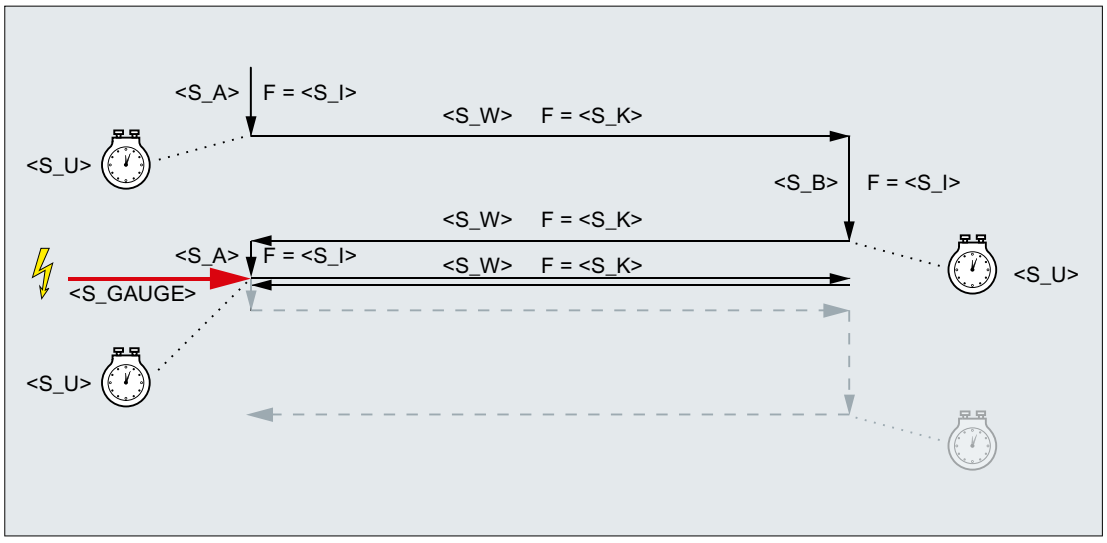


図 20-2　開始位置での切り込みのキャンセル

リソース

サイクルでは、リソースとして、ブロック全体でのシンクロナイズドアクションとシンクロナイズドアクション変数を使用します。シンクロナイズドアクションは、シンクロナイズドアクション範囲(CUS.DIR - 1 ...、CMA.DIR - 1000 ...、CST.DIR – 1199 ...)の空き領域から自動的に決定されます。シンクロナイズドアクション変数として SYG_IS[1] が使用されます。

例

例 1:2 ストロークによる揺動:

サイクルパラメータ

- 開始位置での切り込み深さ:0.02 mm
- 終了位置での切り込み深さ:0.01 mm
- ストローク：100 mm
- スパークアウト時間:1 秒
- 切り込みの送り速度:1 mm/min
- 径方向の送り速度:1000 mm/min
- 繰り返し回数:2
- 揺動軸と切り込み軸:標準ジオメトリ軸

キャンセル信号: 高速入力 1 ($A_IN[1])

プログラムコード

```
N10 T1 D1
N20 CYCLE4072("1",0.02,0.01,100,1,1,1000,2)
N30 M30
```

例 2 :2 ストロークによる揺動:

サイクルパラメータ

- 開始位置での切り込み深さ:0.02 mm
- 終了位置での切り込み深さ:0.01 mm
- ストローク：100 mm
- スパークアウト時間:1 秒
- 切り込みの送り速度:1 mm/min
- 径方向の送り速度:1000 mm/min
- 繰り返し回数:2
- 揺動軸と切り込み軸:標準ジオメトリ軸

キャンセル信号変数$A_DBR[20] < 0.01

プログラムコード

```
N10 T1 D1
N20 CYCLE4072("($A_DBR[20]<0.01)",0.02,0.01,100,1,1,1000,2)
N30 M30
```

20.1.46 CYCLE4073 - 連続切り込みによる長手方向の研削

構文

```
CYCLE4073(<S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>, <S_A1>,
<S_A2>)
```

パラメータ

No.	パラメータ	データタイプ	意味
1	<S_A>	REAL	開始位置での切り込み深さ
2	<S_B>	REAL	終了位置での切り込み深さ
3	<S_W>	REAL	研削幅
4	<S_U>	REAL	スパークアウト時間
5	<S_K>	REAL	径方向の切り込みの送り速度
6	<S_H>	INT	繰り返し回数
7	<S_A1>	AXIS	切り込み軸(任意選択)または 1 番目のジオメトリ軸
8	<S_A2>	AXIS	揺動軸(任意選択)または 2 番目のジオメトリ軸

機能

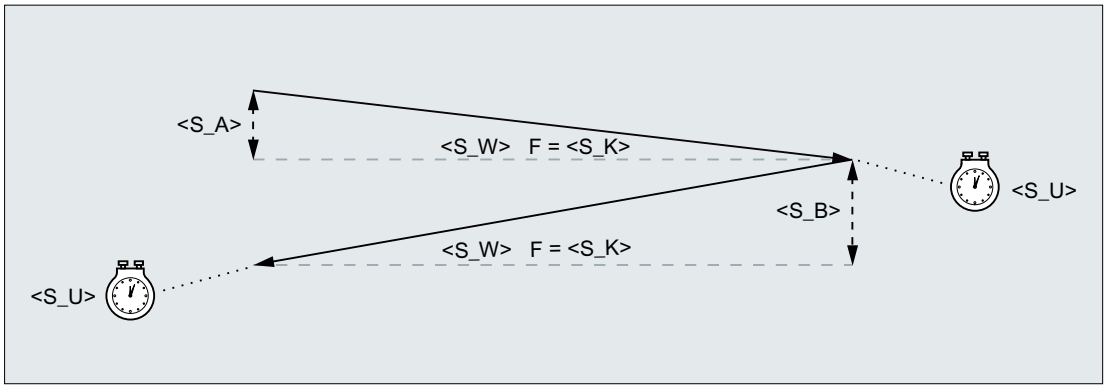
このサイクルは<S_H>切り込みの実行に使用されます。開始位置から終了位置までの切り込みと、終了位置から開始位置までの切り込みは変えることができます。

手順

1. 切り込み深さ 0 の揺動軸の現在の位置でサイクルを開始します。
2. 研削幅 P3 <S_W>の揺動軸を移動軌跡として、径方向の切込みの送り速度 P5 <S_K>で、切り込み深さを開始位置 P1<S_A>の切込み深さまで連続して深くしながら移動します。
3. スパークアウト時間 P4 <S_U>のスパークアウト。
4. 研削幅 P3 <S_W>の揺動軸を開始位置への移動軌跡として、径方向の切込みの送り速度 P5 <S_K>で、切り込み深さを終了位置 P2<S_B>の切込み深さまで連続して深くしながら移動します。
5. スパークアウト時間 P4 <S_U>のスパークアウト。

この手順はシングルブロックによって中断することはできません。

この手順はプログラムされた繰り返し回数 P7(<S_H>)に従って繰り返されます。



例

2 ストロークによる揺動:

サイクルパラメータ

- 開始位置での切り込み深さ:0.02 mm
- 終了位置での切り込み深さ:0.01 mm
- ストローク：100 mm
- スパークアウト時間:1 秒
- 径方向の送り速度:1000 mm/min
- 繰り返し回数:2
- 揺動軸と切り込み軸:標準ジオメトリ軸

プログラムコード

```
N10 T1 D1
N20 CYCLE4073(0.02,0.01,100,1,1000,2)
N30 M30
```

20.1.47 CYCLE4074 - 連続切り込みによる長手方向の研削とキャンセル信号

構文

```
CYCLE4074(<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>,
<S_A1>, <S_A2>)
```

パラメータ

No.	パラメータ	データタイプ	意味
1	<S_GAUGE>	STRING	切り込みのキャンセル条件: 1. 高速入力の回数 2. 論理式
2	<S_A>	REAL	開始位置での切り込み深さ
3	<S_B>	REAL	終了位置での切り込み深さ
4	<S_W>	REAL	研削幅
5	<S_U>	REAL	スパークアウト時間
6	<S_K>	REAL	径方向の切り込みの送り速度
7	<S_H>	INT	繰り返し回数
8	<S_A1>	AXIS	切り込み軸(任意選択)または 1 番目のジオメトリ軸
9	<S_A2>	AXIS	揺動軸(任意選択)または 2 番目のジオメトリ軸

機能

このサイクルは、外部キャンセル信号を考慮した<S_H>切り込みの実行に使用されます。たとえば、切り込み深さは開始位置と終了位置で変えることができます。キャンセル条件が満たされたときに、深さ切り込みがキャンセルされます。深さ切り込みのキャンセル後、常に完全なストロークが 1 回実行されます。

手順

1. 切り込み深さ 0 の揺動軸の現在の位置でサイクルを開始します。
2. 研削幅 P4 <S_W>の揺動軸を移動軌跡として、径方向の切込みの送り速度 P6 <S_K>で、切り込み深さを開始位置 P2<S_A>の切込み深さまで連続して深くしながら移動します。
3. スパークアウト時間 P5 <S_U>のスパークアウト。
4. 研削幅 P4 <S_W>の揺動軸を開始位置への移動軌跡として、径方向の切込みの送り速度 P6 <S_K>で、切り込み深さを終了位置 P3<S_B>の切込み深さまで連続して深くしながら移動します。
5. スパークアウト時間 P5 <S_U>のスパークアウト。
6. キャンセルなし:上に説明した手順が、プログラムされた繰り返し回数 P7(<S_H>)に達するまで繰り返されます。
   キャンセルあり:深さ切り込みがキャンセルされます。次の開始点に達すると加工が停止します。

この手順はシングルブロックによって中断することはできません。

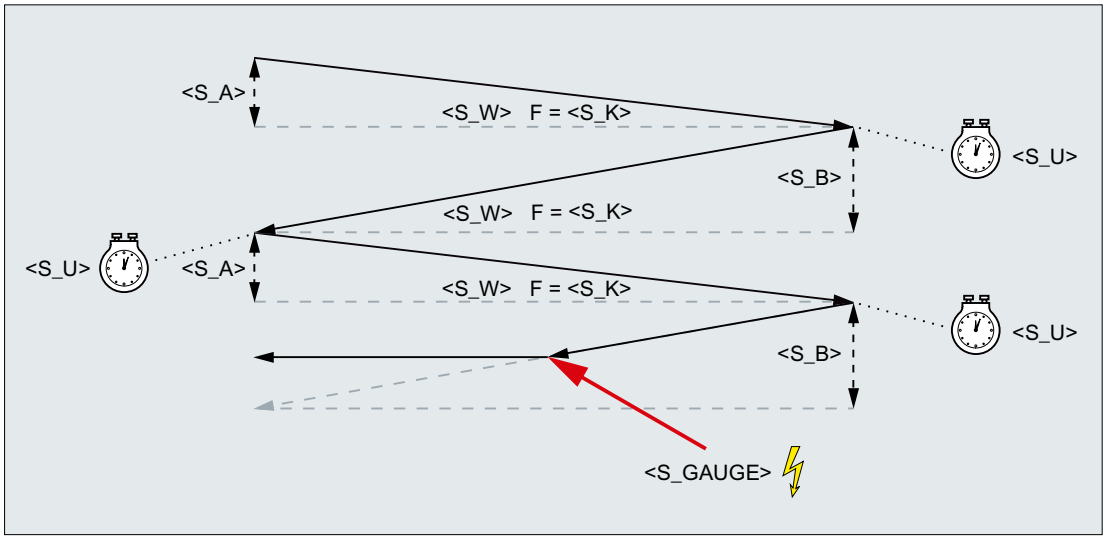


図 20-3　終了位置から開始位置までの切り込みのキャンセル

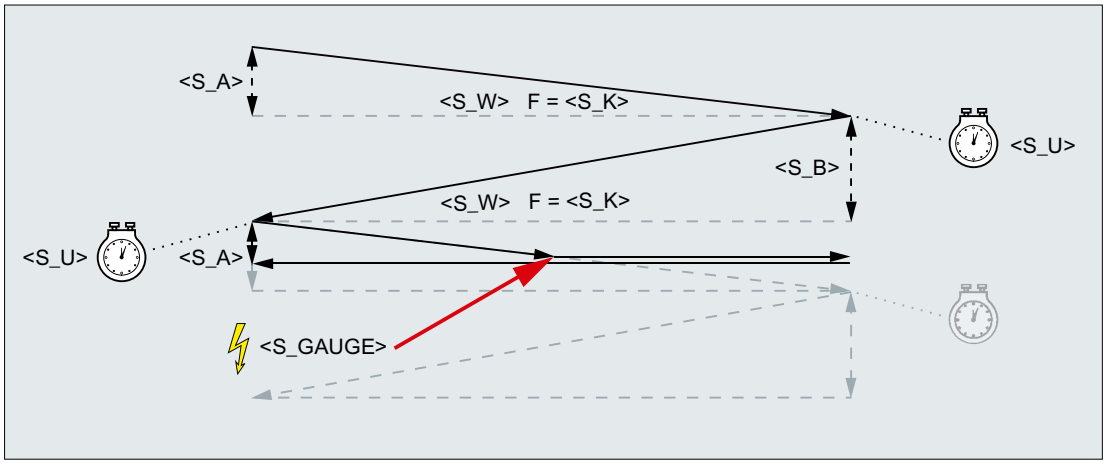


図 20-4　開始位置から終了位置までの切り込みのキャンセル

リソース

サイクルでは、リソースとして、ブロック全体でのシンクロナイズドアクションとシンクロナイズドアクション変数を使用します。シンクロナイズドアクションは、シンクロナイズドアクション範囲(CUS.DIR - 1 ...、CMA.DIR - 1000 ...、CST.DIR – 1199 ...)の空き領域から自動的に決定されます。シンクロナイズドアクション変数として SYG_IS[1] が使用されます。

例

例 1:2 ストロークによる揺動:

サイクルパラメータ

- 開始位置での切り込み深さ:0.02 mm
- 終了位置での切り込み深さ:0.01 mm
- ストローク：100 mm
- スパークアウト時間:1 秒
- 径方向の送り速度:1000 mm/min
- 繰り返し回数:2
- 揺動軸と切り込み軸:標準ジオメトリ軸

キャンセル信号: 高速入力 1 ($A_IN[1])

プログラムコード

```
N10 T1 D1
N20 CYCLE4074("1",0.02,0.01,100,1,1000,2)
N30 M30
```

例 2 :2 ストロークによる揺動:

サイクルパラメータ

- 開始位置での切り込み深さ:0.02 mm
- 終了位置での切り込み深さ:0.01 mm
- ストローク：100 mm
- スパークアウト時間:1 秒
- 径方向の送り速度:1000 mm/min
- 繰り返し回数:2
- 揺動軸と切り込み軸:標準ジオメトリ軸

キャンセル信号変数$A_DBR[20] < 0.01

プログラムコード

```
N10 T1 D1
N20 CYCLE4074("($A_DBR[20]<0.01)",0.02,0.01,100,1,1000,2)
N30 M30
```

20.1.48 CYCLE4075 - 反転点での切り込みによる平面研削

構文

```
CYCLE4075(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>,
<S_A1>, <S_A2>)
```

パラメータ

No.	パラメータ	データタイプ	意味
1	<S_I>	REAL	開始位置での切り込み深さ
2	<S_J>	REAL	終了位置での切り込み深さ
3	<S_K>	REAL	合計切り込み深さ
4	<S_A>	REAL	研削幅
5	<S_R>	REAL	切り込みの送り速度
6	<S_F>	REAL	径方向の切り込みの送り速度
7	<S_P>	REAL	スパークアウト時間
8	<S_A1>	AXIS	切り込み軸(任意選択)
9	<S_A2>	AXIS	揺動軸(任意選択)

機能

このサイクルは、切り込みステップでの切り込み全体の深さ P3 <S_K>での加工に使用されます。開始位置 P1 <S_I>と終了位置 P2 <S_J>の切り込み深さは変えることができます。切り込み間で接線方向の移動があります。

位置データ P1~P4 は負または正とすることができます。

切り込み軸 P8 <S_A1>または揺動軸 P9 <S_A2>の指定はオプションです。一方または両方のパラメータを指定していない場合、サイクルはチャネルの最初の 2 つのジオメトリ軸を使用します。

開始位置 P1 <S_I>と終了位置 P2 <S_J>の切り込み深さの合計が 0、または合計切り込み深さ P3 <S_K>が 0 の場合、スパークアウトストロークのみが実行されます。

手順

1. 揺動軸の現在の位置でサイクルを開始します。
2. 切り込み軸を開始位置 P1<S_I>の切込み深さまで、切り込み P5 <S_R>用の送り速度で移動します。

3. スパークアウト時間 P7 <S_P>のスパークアウト。
4. 研削幅 P4<S_A>の揺動軸を移動軌跡として、径方向の切り込みの送り速度 P6 <S_F>で移動します。
5. 切り込み軸を終了位置 P2<S_J>の切込み深さまで、切り込み P5 <S_R>用の送り速度で移動します。
6. スパークアウト時間 P7 <S_P>のスパークアウト。
7. 研削幅 P4<S_A>の揺動軸を移動軌跡として、径方向の切込みの送り速度 P6 <S_F>で開始点まで移動します。

この手順はシングルブロックによって中断することはできません。

この手順は合計切り込み深さ P3 <S_K>に達するまで繰り返されます。最後のストロークは、不均等に配分されます。

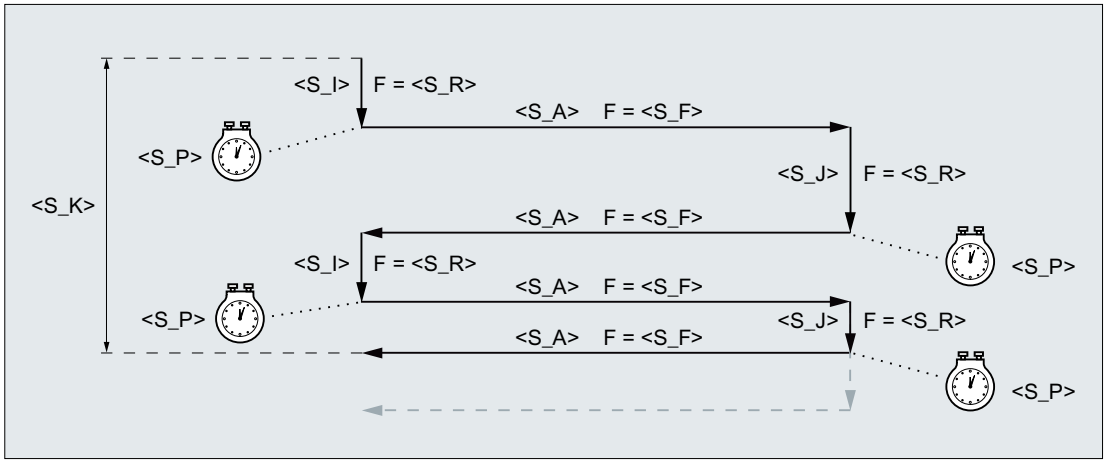


図 20-5　2 番目の反転点での切り込みで合計切り込み深さに到達

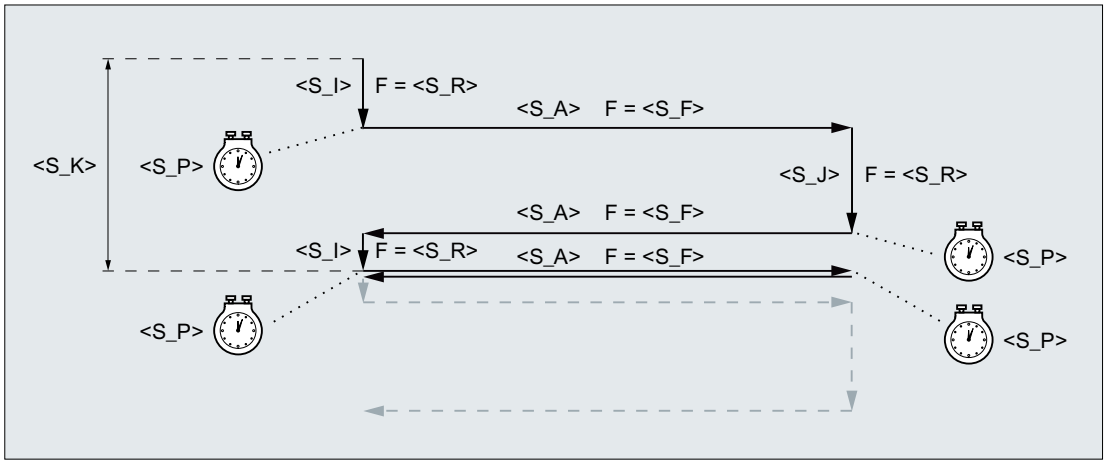


図 20-6　1 番目の反転点での切り込みで合計切り込み深さに到達

例

以下の仕様での揺動

- 開始位置での切り込み深さ 0.02 mm
- 終了位置での切り込み深さ 0.01 mm
- 合計切り込み深さ 1 mm
- ストローク 100 mm
- 切り込みの送り速度 1 mm/min
- 径方向の送り速度 1000 mm/min
- スパークアウト時間 1 秒
- 標準ジオメトリ軸

プログラムコード

```
N10 T1 D1
N20 CYCLE4075(0.02,0.01,1,100,1,1000,1)
N30 M30
```

20.1.49 CYCLE4077 - 反転点での切り込みによる平面研削とキャンセル信号

構文

```
CYCLE4077(<S_GAUGE>, <S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>,
<S_P>, <S_A1>, <S_A2>)
```

パラメータ

No.	パラメータ	データタイプ	意味
1	<S_GAUGE>	STRING	切り込みのキャンセル条件 ● 高速入力の回数 ● 論理式
2	<S_I>	REAL	開始位置での切り込み深さ
3	<S_J>	REAL	終了位置での切り込み深さ
4	<S_K>	REAL	合計切り込み深さ
5	<S_A>	REAL	研削幅
6	<S_R>	REAL	切り込みの送り速度

No.	パラメータ	データタイプ	意味
7	<S_F>	REAL	径方向の切り込みの送り速度
8	<S_P>	REAL	スパークアウト時間
9	<S_A1>	AXIS	切り込み軸(任意選択)
10	<S_A2>	AXIS	揺動軸(任意選択)

機能

このサイクルは、切り込みステップでの切り込み全体の深さ P4 <S_K>での加工に使用されます。開始位置 P2 <S_I>と終了位置 P3 <S_J>の切り込み深さは変えることができます。切り込み間で接線方向の移動があります。高速入力のキャンセル信号が 1 であるか、キャンセル条件が満たされた場合、深さ切り込みがキャンセルされます。キャンセル後、常に完全なストロークが 1 回実行されます。

位置データ P2~P5 は負または正とすることができます。

切り込み軸 P9 <S_A1>または揺動軸 P10 <S_A2>の指定はオプションです。一方または両方のパラメータを指定していない場合、サイクルはチャネルの最初の 2 つのジオメトリ軸を使用します。

開始位置 P2 <S_I>と終了位置 P3 <S_J>の切り込み深さの合計が 0、または合計切り込み深さ P4 <S_K>が 0 の場合、スパークアウトストロークのみが実行されます。

手順

1. 揺動軸の現在の位置でサイクルを開始します。
2. 切り込み軸を開始点 P2<S_I>の切込み深さまで、切り込み P6 <S_R>用の送り速度で移動します。
3. スパークアウト時間 P8 <S_P>のスパークアウト。
4. 研削幅 P5<S_A>の揺動軸を移動軌跡として、径方向の切り込みの送り速度 P7 <S_F>で移動します。
5. 切り込み軸を終了位置 P3<S_J>の切込み深さまで、切り込み P6 <S_R>用の送り速度で移動します。
6. スパークアウト時間 P8 <S_P>のスパークアウト。
7. 研削幅 P5<S_A>の揺動軸を移動軌跡として、径方向の切込みの送り速度 P7 <S_F>で開始点まで移動します。
8. キャンセルなし:上に説明した手順は合計切り込み深さ P4 <S_K>に達するまで繰り返されます。最後のストロークは、不均等に配分されます。
   キャンセルあり:加工は開始点で停止されます。

この手順はシングルブロックによって中断することはできません。

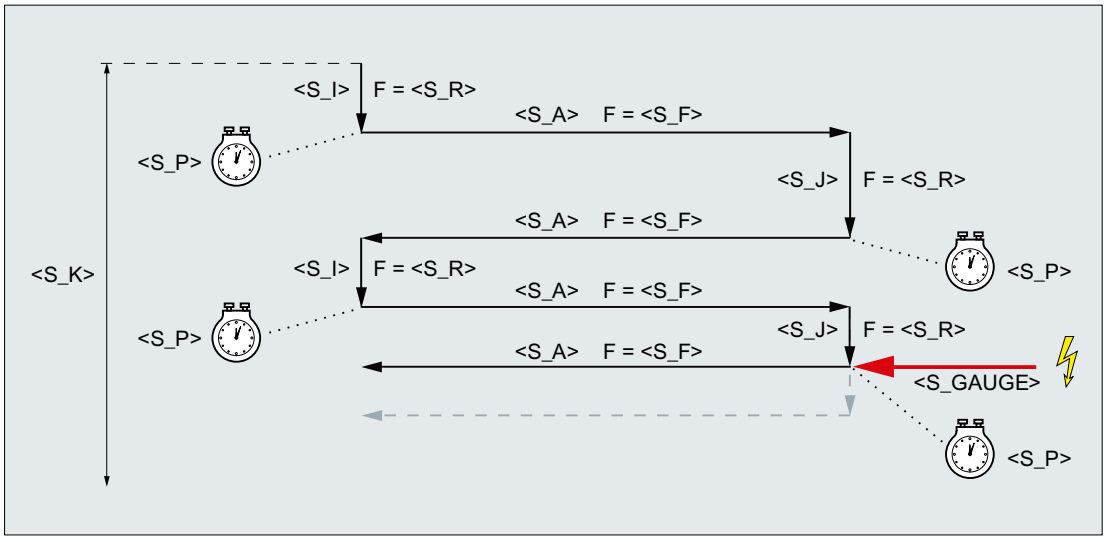


図 20-7　終了位置での切り込みのキャンセル

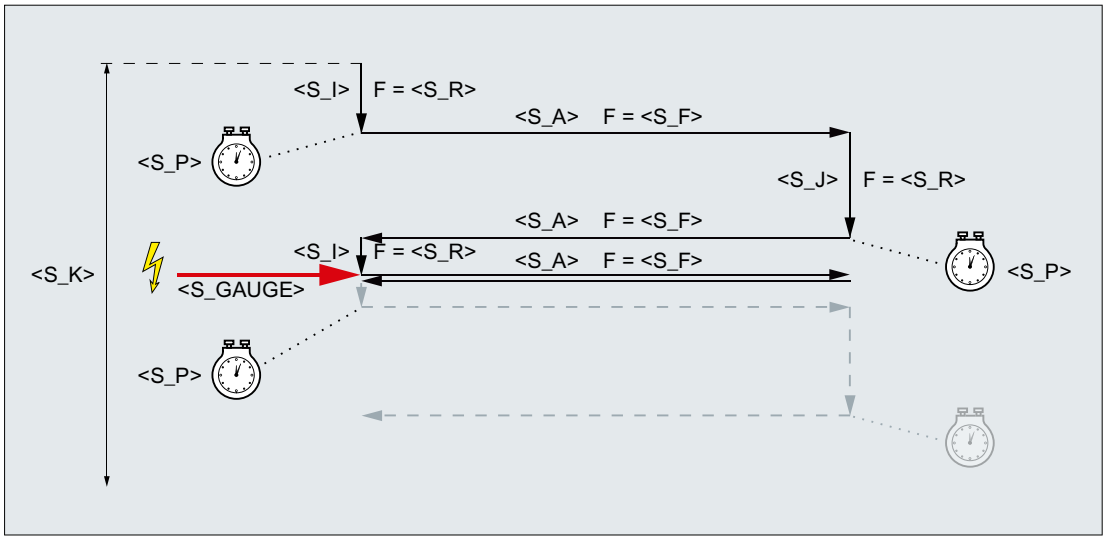


図 20-8　開始位置での切り込みのキャンセル

リソース

サイクルでは、リソースとして、ブロック全体でのシンクロナイズドアクションとシンクロナイズドアクション変数を使用します。シンクロナイズドアクションは、シンクロナイズドアクション範囲(CUS.DIR - 1 ...、CMA.DIR - 1000 ...、CST.DIR – 1199 ...)の空き領域から自動的に決定されます。シンクロナイズドアクション変数として SYG_IS[1] が使用されます。

例

例 1

以下の仕様での揺動

- 開始位置での切り込み深さ 0.02 mm
- 終了位置での切り込み深さ 0.01 mm
- 合計切り込み深さ 1 mm
- ストローク 100 mm
- 切り込みの送り速度 1 mm/min
- 径方向の送り速度 1000 mm/min
- スパークアウト時間 1 秒
- 標準ジオメトリ軸

キャンセル信号: 高速入力 1 ($A_IN[1])

プログラムコード

```
N10 T1 D1
N20 CYCLE4077("1",0.02,0.01,1,100,1,1000,1)
N30 M30
```

例 2

以下の仕様での揺動

- 開始位置での切り込み深さ 0.02 mm
- 終了位置での切り込み深さ 0.01 mm
- 合計切り込み深さ 1 mm
- ストローク 100 mm
- 切り込みの送り速度 1 mm/min
- 径方向の送り速度 1000 mm/min
- スパークアウト時間 1 秒
- 標準ジオメトリ軸

キャンセル信号: デュアルポート RAM 変数 20 が 0.01 未満($A_DBR[20] < 0.01)

プログラムコード

```
N10 T1 D1
N20 CYCLE4077("($A_DBR[20]<0.01)",0.02,0.01,1,100,1,1000,1)
```

プログラムコード
N30 M30

20.1.50 CYCLE4078 - 連続切り込みによる平面研削

構文

```
CYCLE4078(<S_I>, <S_J>, <S_K>, <S_A>, <S_F>, <S_P>, <S_A1>,
<S_A2>)
```

パラメータ

No.	パラメータ	データタイプ	意味
1	<S_I>	REAL	開始位置から終了位置までの切り込み深さ
2	<S_J>	REAL	終了位置から開始位置までの切り込み深さ
3	<S_K>	REAL	合計切り込み深さ
4	<S_A>	REAL	研削幅
5	<S_F>	REAL	送り速度
6	<S_P>	REAL	スパークアウト時間
7	<S_A1>	AXIS	切り込み軸(任意選択)
8	<S_A2>	AXIS	揺動軸(任意選択)

機能

このサイクルは、連続切り込みを使用した切り込み全体の深さ P3 <S_K>での加工に使用されます。開始位置から終了位置 P1 <S_I>までの切り込み深さと、終了位置から開始位置 P2<S_J>までのの切り込み深さは変えることができます。

位置データ P1~P4 は負または正とすることができます。

切り込み軸 P8 <S_A1>または揺動軸 P9 <S_A2>の指定はオプションです。一方または両方のパラメータを指定していない場合、サイクルはチャネルの最初の 2 つのジオメトリ軸を使用します。

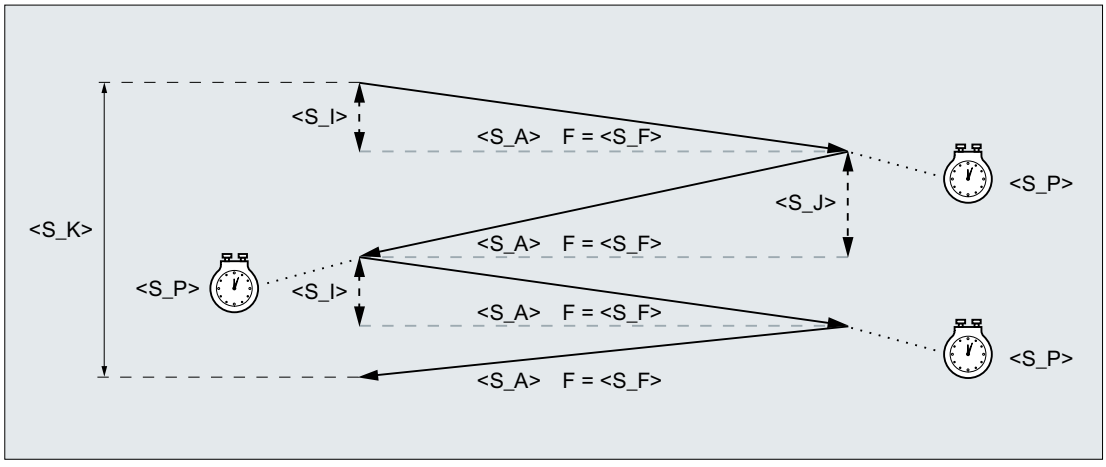
P1 <S_I>と P2 <S_J>の切り込み深さの合計が 0、または合計切り込み深さ P3 <S_K>が 0 の場合、スパークアウトストロークのみが実行されます。

手順

1. 切り込み深さ 0 の揺動軸の現在の位置でサイクルを開始します。
2. 研削幅 P4 <S_A>の揺動軸を移動軌跡として、送り速度 P5 <S_F>で、切り込み深さを開始位置 P1<S_I>の切り込み深さまで連続して深くしながら移動します。
3. スパークアウト時間 P7 <S_P>のスパークアウト。
4. 研削幅 P4 <S_A>の揺動軸を開始点への移動軌跡として、送り速度 P5 <S_F>で、切り込み深さを終了位置 P2<S_J>の切り込み深さまで連続して深くしながら移動します。
5. スパークアウト時間 P7 <S_P>のスパークアウト。
6. 研削幅 P4<S_A>の揺動軸を開始点への移動軌跡として、送り速度 P5 <S_F>で移動します。

この手順はシングルブロックによって中断することはできません。

この手順は合計切り込み深さ P3 <S_K>に達するまで繰り返されます。最後のストロークは、不均等に配分されます。



例

以下の仕様での揺動

- 開始位置での切り込み深さ 20 mm
- 終了位置での切り込み深さ 10 mm
- 合計切り込み深さ 100 mm
- ストローク 100 mm
- 送り速度 1000 mm/min
- スパークアウト時間 1 秒
- 標準ジオメトリ軸

プログラムコード

```
N10 T1 D1
N20 CYCLE4078(20,10,100,100,1000,1)
N30 M30
```

20.1.51 CYCLE4079 - 間欠切り込みによる平面研削

構文

```
CYCLE4079(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>,
<S_A1>, <S_A2>)
```

パラメータ

No.	パラメータ	データタイプ	意味
1	<S_I>	REAL	開始位置での切り込み深さ
2	<S_J>	REAL	終了位置での切り込み深さ
3	<S_K>	REAL	合計切り込み深さ
4	<S_A>	REAL	研削幅
5	<S_R>	REAL	切り込みの送り速度
6	<S_F>	REAL	径方向の切り込みの送り速度
7	<S_P>	REAL	スパークアウト時間
8	<S_A1>	AXIS	切り込み軸(任意選択)
9	<S_A2>	AXIS	揺動軸(任意選択)

機能

このサイクルは、切り込みステップでの切り込み全体の深さ P3 <S_K>での加工に使用されます。開始位置 P1 <S_I>と終了位置 P2 <S_J>の切り込み深さは変えることができます。切り込み間で接線方向の移動があります。

位置データ P1~P4 は負または正とすることができます。

切り込み軸 P8 <S_A1>または揺動軸 P9 <S_A2>の指定はオプションです。一方または両方のパラメータを指定していない場合、サイクルはチャネルの最初の 2 つのジオメトリ軸を使用します。

開始位置 P1 <S_I>と終了位置 P2 <S_J>の切り込み深さの合計が 0、または合計切り込み深さ P3 <S_K>が 0 の場合、スパークアウトストロークのみが実行されます。

手順

1. 揺動軸の現在の位置でサイクルを開始します。
2. 切り込み軸を開始位置 P1<S_I>の切込み深さまで、切り込み P5 <S_R>用の送り速度で移動します。
3. スパークアウト時間 P7 <S_P>のスパークアウト。
4. 研削幅 P4<S_A>の揺動軸を移動軌跡として、径方向の切り込みの送り速度 P6 <S_F>で移動します。
5. 切り込み軸を終了位置 P2<S_J>の切込み深さまで、切り込み P5 <S_R>用の送り速度で移動します。
6. スパークアウト時間 P7 <S_P>のスパークアウト。
7. 研削幅 P4<S_A>の揺動軸を移動軌跡として、径方向の切込みの送り速度 P6 <S_F>で開始点まで移動します。

この手順はシングルブロックによって中断することはできません。

この手順は合計切り込み深さ P3 <S_K>に達するまで繰り返されます。最後のストロークは、不均等に配分されます。

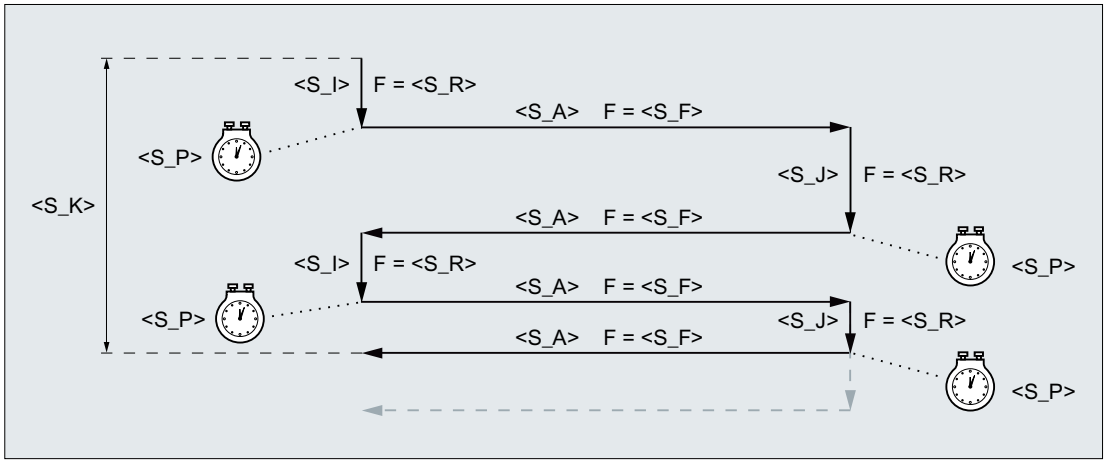


図 20-9　2 番目の反転点での切り込みで合計切り込み深さに到達

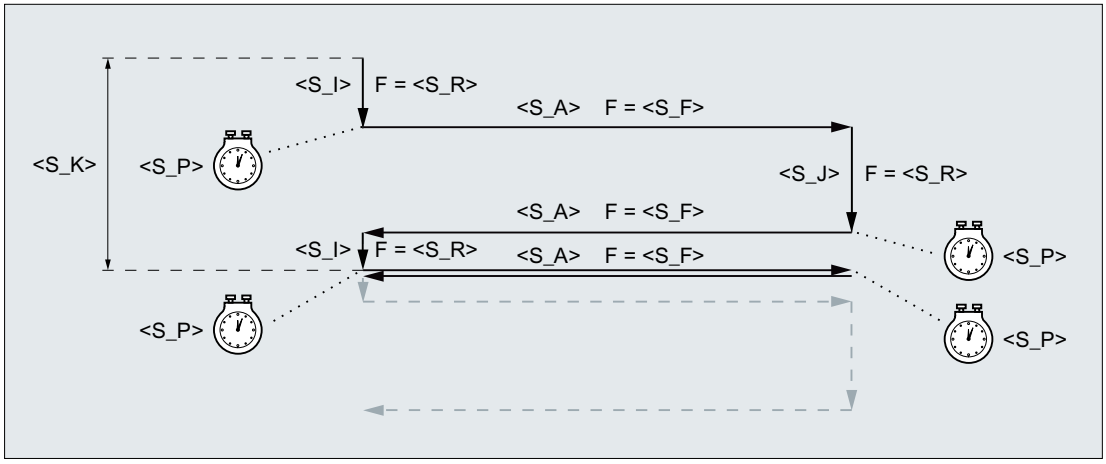


図 20-10　1 番目の反転点での切り込みで合計切り込み深さに到達

例

以下の仕様での揺動

- 開始位置での切り込み深さ 0.02 mm
- 終了位置での切り込み深さ 0.01 mm
- 合計切り込み深さ 1 mm
- ストローク 100 mm
- 切り込みの送り速度 1 mm/min
- 径方向の送り速度 1000 mm/min
- スパークアウト時間 1 秒
- 標準ジオメトリ軸

プログラムコード

```
N10 T1 D1
N20 CYCLE4079(0.02,0.01,1,100,1,1000,1)
N30 M30
```

20.1.52　GROUP_BEGIN - プログラムブロック開始

構文

```
GROUP_BEGIN(<_LEVEL>, <_NAME>, <_SP>, <_MODE>, <S_ICON>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1		<_LEVEL>	INT	レベル		
				0 =	メインレベル	
				1 =	1 番目のサブレベル	
2		<_NAME>	STRING[128]	ブロック名称		
3		<_SP>	INT	主軸		
				0 =	主軸なし	
				1 =	主軸	
				2 =	対向主軸	
4		<_MODE>	INT	モード		
				ビット 0	= 1	GROUP_ADDEND が存在します
				ビット 1	= 1	ShopTurn:自動後退(工具交換位置に移動)
				ビット 12	予備	
				ビット 13	予備	
5		<S_ICON>	STRING[32]	アイコン名(操作画面専用)		

20.1.53 GROUP_END - プログラムブロック終了

構文

```
GROUP_END(<_LEVEL>, <_SP>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
1		<_LEVEL>	INT	レベル	
				0 =	メインレベル
				1 =	1 番目のサブレベル
2		<_SP>	INT	主軸	
				0 =	主軸なし
				1 =	主軸
				2 =	対向主軸

20.1.54 GROUP_ADDEND - 試し切り追加終了

構文

```
GROUP_ADDEND(<_LEVEL>, <_SP>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
1		<_LEVEL>	INT	レベル	
				0 =	メインレベル
				1 =	1 番目のサブレベル
2		<_SP>	INT	主軸	
				0 =	主軸なし
				1 =	主軸
				2 =	対向主軸

20.1.55 必要条件

20.1.55.1 サイクル画面の機能スケーリング

機能スケーリングを有効にすると、さまざまなサイクル画面で簡易入力を選択し、最も重要なサイクルパラメータのみを表示することができます。

たとえば、簡易入力は、次のサイクル画面で選択できます。

機能	サイクル画面
穴あけ	深穴ドリル
	タッピング
フライス加工	長方形ポケット
	輪郭フライス加工:ポケット
旋削	ねじの旋削:長手方向
	輪郭旋削:切削
	輪郭旋削:溝切り
	輪郭旋削:溝旋削

当該のサイクル画面の操作画面で、オプション[入力:**簡易**]と[入力:**完全**]を選択できます。

表示されないサイクルパラメータ

簡易入力画面で表示されないサイクルパラメータは、プリセットされた固定のパラメータで、技術的には有用ですが、可変の値がないものです。あるいは、チャネル別サイクルセッティングデータによって、パラメータ設定された値がサイクルパラメータに割り当てられます。「セットアップ」の「チャネル別サイクルセッティングデータ」の下の節を参照してください。

[入力:完全]から[入力:簡易]への切り替え

[入力: 完全]の設定でサイクル画面に入力し、[入力: 簡易]に切り替えた場合、サイクル呼び出しを生成したときに、表示されなくなったパラメータには初期設定またはセッティングデータの値が使用されます。

セットアップ

チャネル別構成マシンデータ

サイクル画面の機能スケーリングは、マシンデータで有効にできます。

MD52210 $MCS_FUNCTION_MASK_DISP, bit 9 = 1 (select display "Input simple")

チャネル別サイクルセッティングデータ

サイクル画面で簡易入力を有効にしている場合、次のセッティングデータで特定のサイクルパラメータの値を指定できます。

番号	識別子	意味
SD55300	$SCS_EASY_SAFETY_CLEARANCE	安全距離
SD55301	$SCS_EASY_DWELL_TIME	ドウェル時間
SD55305	$SCS_EASY_DRILL_DEEP_FD1	深穴ドリル:パーセンテージ:1 番目の送り速度
SD55306	$SCS_EASY_DRILL_DEEP_DF	深穴ドリル:パーセンテージ:切り込み
SD55307	$SCS_EASY_DRILL_DEEP_V1	深穴ドリル:最小切り込み深さ
SD55308	$SCS_EASY_DRILL_DEEP_V2	深穴ドリル:後退距離
SD55309	$SCS_EASY_THREAD_RETURN_DIST	ねじの旋削:戻り距離

20.2 計測サイクル

計測サイクルは、当社が提供する特別なサブプログラムで、特定の計測作業を実行します。一般的なサイクルと同様に、計測サイクルは、パラメータ設定によって特定の問題に適応させることもできます。

計測サイクルは、次の領域および用途で使用できます。

- 工具計測、旋盤/フライス盤
- ワーク計測、旋盤/フライス盤

参照先

計測サイクルの詳細については、以下を参照してください。

プログラミングマニュアル、計測サイクル

テーブル

21

21.1 命令

注記

サイクル

操作リストには、NC プログラム(G コード)で発生するすべてのサイクルが含まれます。サイクルには、マスクを使用してプログラムエディタでプログラムできるものと、プログラムサポートなしでループ用にプログラムしなければならないものがあります。互換性維持のために制御装置で引き続き利用できるものの、SINUMERIK Operate プログラムエディタを使って編集できないサイクル(「互換性サイクル」)は考慮されません。

命令 A ... C

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
:	O	NC メインブロック番号、ジャンプラベル終了、結合演算子		+		PGAsl
*	O	乗算演算子		+		PGAsl
+	O	加算演算子		+		PGAsl
-	O	減算演算子		+		PGAsl
<	O	比較演算子、より小さい		+		PGAsl
<<	O	文字列用結合演算子		+		PGAsl
<=	O	比較演算子、以下		+		PGAsl
=	O	代入演算子		+		PGAsl
>=	O	比較演算子、以上		+		PGAsl
/	O	除算演算子		+		PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
/0 … … /7		ブロックをスキップします(スキップレベル 1) ... ... ブロックをスキップします(キップレベル 8)		+		PGsl
A	A	軸名称	m/s	+		PGAsl
A2	A	工具オリエンテーション:RPY 角またはオイラー角	s	+		PGAsl
A3	A	工具オリエンテーション:方向ベクトルの第 1 成分	s	+		PGAsl
A4	A	工具オリエンテーション:ブロック開始時の面法線ベクトルの第 1 成分	s	+		PGAsl
A5	A	工具オリエンテーション:ブロック終了時の面法線ベクトルの第 1 成分	s	+		PGAsl
A6	A	工具オリエンテーション:テーパの回転軸の方向ベクトルの第 1 成分	s	+		PGAsl
A7	A	工具オリエンテーション:テーパの円筒面の中間位置決めの第 1 ベクトル要素	s	+		PGAsl
ABS	F	アブソリュート値(数量)		+	+	PGAsl
AC	K	座標/位置のアブソリュート指令	s	+		PGsl
ACC	K	現在の軸加減速度の働き	M	+	+	PGsl
ACCLIMA	K	現在の最大軸加減速度の働き	M	+	+	PGAsl
ACN	K	回転軸のアブソリュート指令、負方向位置へのアプローチ	s	+		PGsl
ACOS	F	アークコサイン (三角関数)		+	+	PGAsl
ACP	K	回転軸のアブソリュート指令、正方向位置へのアプローチ	s	+		PGsl
ACTBLOCNO	P	「実行中のブロックの表示をマスク」(DISPLOF)が有効の場合でも、アラームブロックの実行中のブロック番号を出力。		+		PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
ADDFRAME	F	計測されたフレームの結合と起動		+	-	PGAsl、FB1sl (K2)
ADIS	A	軌跡機能 G1、G2、G3、...の丸み付き隙間	M	+		PGsl
ADISPOS	A	早送り G0 の丸み付き隙間	M	+		PGsl
ADISPOSA	P	IPOBRKA の許容範囲の大きさ	M	+	+	PGAsl
ALF	A	高速リトラクト角度	M	+		PGAsl
AMIRROR	G	プログラマブルミラーリング	s	+		PGsl
および	K	論理積		+		PGAsl
ANG	A	輪郭角度	s	+		PGsl
AP	A	極角度	m/s	+		PGsl
APR	K	読み出し/表示のアクセス保護		+		PGAsl
APRB	K	読み出しのアクセス権、OPI		+		PGAsl
APRP	K	読み出しのアクセス権、パートプログラム		+		PGAsl
APW	K	書き込みのアクセス保護		+		PGAsl
APWB	K	書き込みのアクセス権、OPI		+		PGAsl
APWP	K	書き込みのアクセス権、パートプログラム		+		PGAsl
APX	K	指定した言語要素を実行するためのアクセス権の定義		+		PGAsl
AR	A	開口角度	m/s	+		PGsl
AROT	G	プログラマブル座標回転	s	+		PGsl
AROTS	G	立体角によるプログラマブルフレームの回転	s	+		PGsl
AS	K	マクロ定義		+		PGAsl
ASCALE	G	プログラマブルスケーリング	s	+		PGsl
ASIN	F	算術機能、アークサイン		+	+	PGAsl
ASPLINE	G	A スプライン	M	+		PGAsl
ATAN2	F	アークタンジェント 2		+	+	PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
ATOL	A	コンプレッサ機能、旋回スムージング、およびスムージングタイプの軸別許容範囲	M	+		PGAsl
ATRANS	G	追加プログラマブルゼロオフセット	s	+		PGsl
AUXFUDEL	P	グローバルリストから、特定チャネルの補助機能を削除		+	-	FB1sl (H2)
AUXFUDELG	P	グローバルリストから、特定チャネルの補助機能グループのすべての補助機能を削除		+	-	FB1sl (H2)
AUXFUMSEQ	P	M 補助機能の出力処理を決定		+	-	FB1sl (H2)
AUXFUSYNC	P	補助機能のグローバルリストの文字列としての、チャネル別 SERUPRO と ASUB の完全なパートプログラムブロックを生成		+	-	FB1sl (H2)
AX	K	可変軸識別子	m/s	+		PGAsl
AXCTSWE	P	軸コンテナの回転		+	-	PGAsl
AXCTSWEC	P	軸コンテナの回転の有効化を取り消し		+	+	PGAsl
AXCTSWED	P	軸コンテナの回転(セットアップ用の命令タイプ)		+	-	PGAsl
AXIS	K	軸識別子、軸アドレス		+		PGAsl
AXNAME	F	入力文字列を軸識別子に変換		+	-	PGAsl
AXSTRING	F	主軸番号を文字列に変換		+	-	PGAsl
AXTOCHAN	P	特定のチャネルのための軸の要求。NC プログラムとシンクロナイズドアクションで実行可能です。		+	+	PGAsl
AXTOSPI	F	軸識別子を主軸インデックスへ変換		+	-	PGAsl
B	A	軸名称	m/s	+		PGAsl
B2	A	工具オリエンテーション:RPY 角またはオイラー角	s	+		PGAsl
B3	A	工具オリエンテーション:方向ベクトルの第 2 成分	s	+		PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
B4	A	工具オリエンテーション:ブロック開始時の面法線ベクトルの第 2 成分	s	+		PGAsl
B5	A	工具オリエンテーション:ブロック終了時の面法線ベクトルの第 2 成分	s	+		PGAsl
B6	A	工具オリエンテーション:テーパの回転軸の方向ベクトルの第 2 成分	s	+		PGAsl
B7	A	工具オリエンテーション:テーパの円筒面の中間位置決めの第 2 ベクトル要素	s	+		PGAsl
B_AND	O	ビットごとの AND		+		PGAsl
B_OR	O	ビットごとの OR		+		PGAsl
B_NOT	O	ビットごとの否定		+		PGAsl
B_XOR	O	ビットごとの排他的 OR		+		PGAsl
BAUTO	G	隣接する 3 点による最初のスプライン区間の定義	M	+		PGAsl
BLOCK	K	キーワード TO との併用により、間接サブプログラム呼び出しで処理するプログラム部分を定義します		+		PGAsl
BLSYNC	K	次回のブロック切り替えでのみ、割り込みルーチンの処理を起動します		+		PGAsl
BNAT[6]	G	最初のスプラインブロックへの自然遷移	M	+		PGAsl
Bool	K	データタイプ:TRUE/FALSE または 1/0 のブール値です		+		PGAsl
BOUND	F	値が定義された数値範囲内にあるかどうかをテストします。複数の値が等しい場合は、テスト値を返します。		+	+	PGAsl
BRISK[6]	G	スムージングをおこなわない高速の軌跡加減速度	M	+		PGAsl
BRISKA	P	プログラム指令軸に対して最大の軌跡加減速度を起動します		+	-	PGAsl
BSPLINE	G	B スプライン	M	+		PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
BTAN	G	最初のスプラインブロックへ接線方向の遷移	M	+		PGAsl
C	A	軸名称	m/s	+		PGAsl
C2	A	工具オリエンテーション:RPY 角またはオイラー角	s	+		PGAsl
C3	A	工具オリエンテーション:方向ベクトルの第 3 成分	s	+		PGAsl
C4	A	工具オリエンテーション:ブロック開始時の面法線ベクトルの第 3 成分	s	+		PGAsl
C5	A	工具オリエンテーション:ブロック終了時の面法線ベクトルの第 3 成分	s	+		PGAsl
C6	A	工具オリエンテーション:テーパの回転軸の方向ベクトルの第 3 成分	s	+		PGAsl
C7	A	工具オリエンテーション:テーパの円筒面の中間位置決めの第 3 ベクトル要素	s	+		PGAsl
CAC	K	アブソリュート位置へアプローチします		+		PGAsl
CACN	K	テーブルの記述値で負方向へアブソリュートのアプローチをおこないます		+		PGAsl
CACP	K	テーブルの記述値で正方向へアブソリュートのアプローチをおこないます		+		PGAsl
CALCDAT	F	3 または 4 点から、円弧の半径と中心点を計算します		+	-	PGAsl
CALCPOSI	F	プロテクションゾーン違反、ワーキングエリアリミット、およびソフトウェアリミットのチェック		+	-	PGAsl
CALL	K	間接サブプログラム呼び出し		+		PGAsl
CALLPATH	P	サブプログラム呼び出しのプログラム指令可能な検索パス		+	-	PGAsl
CANCEL	P	モーダルシンクロナイズドアクションをキャンセルします		+	-	FBSYsl
CASE	K	条件付きプログラム分岐		+		PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
CDC	K	位置への直接アプローチ		+		PGAsl
CDOF [6]	G	衝突監視をオフにします	M	+		PGsl
CDOF2	G	3 次元外周加工中の衝突監視をオフにします	M	+		PGsl
CDON	G	衝突監視を有効にします	M	+		PGsl
CFC [6]	G	輪郭の一定送り速度	M	+		PGsl
CFIN	G	外側半径ではなく、内側半径のみでの一定送り速度	M	+		PGsl
CFINE	F	FRAME 変数への仕上げオフセットの割り当て		+	-	PGAsl
CFTCP	G	工具中心点の一定送り速度(中心点軌跡)	M	+		PGsl
CHAN	K	データの有効範囲を指定します		+		PGAsl
CHANDATA	P	チャネルデータアクセス用のチャネル番号の設定		+	-	PGAsl
CHAR	K	データタイプ:ASCII 文字		+		PGAsl
CHF	A	面取り; 値 = 面取り長さ	s	+		PGsl
CHKDM	F	マガジン内での一義性のチェック		+	-	FBWsl
CHKDNO	F	固有の D 番号のチェック		+	-	PGAsl
CHR	A	面取り; 値 = 移動方向の面取り長さ		+		PGsl
CIC	K	ステップ値によるアプローチ位置		+		PGAsl
CIP	G	中間点経由の円弧補間	M	+		PGsl
CLEARM	P	チャネル協調の 1 個/複数のマークをリセット		+	+	PGAsl
CLRINT	P	割り込みの解除		+	-	PGAsl
CMIRROR	F	座標軸のミラーリング		+	-	PGAsl
COARSEA	K	「汎用イグザクトストップ」に達すると移動終了	M	+		PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
COLLPAIR	F	干渉ペアの要素であるかどうかをチェック		+		PGAsl
COMPCAD	G	コンプレッサ機能 COMPCAD を有効にします。	M	+		PGAsl
COMPCURV	G	コンプレッサ機能 COMPCURV を有効にします	M	+		PGAsl
COMPLETE		データの読み出しと書き込みの制御命令です		+		PGAsl
COMPOF[6]	G	NC ブロック圧縮を無効にします	M	+		PGAsl
COMPON	G	コンプレッサ機能 COMPON を有効にします	M	+		PGAsl
COMPSURF	G	コンプレッサ機能 COMPSURF を有効にします	M	+		PGAsl
CONTDCON	P	テーブル輪郭の解読を有効にします		+	-	PGAsl
CONTPRON	P	輪郭解析の起動		+	-	PGAsl
CORROF	P	すべての有効な重畳移動を解除		+	-	PGsl
CORRTRAFO	F	機械のキネマティックモデルの方向軸のオフセットベクトルまたは方向ベクトルの変更		+	-	PGAsl
cos	F	コサイン (三角関数)		+	+	PGAsl
COUPDEF	P	ELG グループ/主軸同期グループの定義		+	-	PGAsl
COUPDEL	P	ELG グループの解除		+	-	PGAsl
COUPOF	P	ELG グループ/主軸同期グループのペアの無効化		+	-	PGAsl
COUPOFS	P	スレーブ主軸の停止により、ELG グループ/主軸同期ペアを解除		+	-	PGAsl
COUPON	P	ELG グループ/主軸同期グループのペアの有効化		+	-	PGAsl
COUPONC	P	以前のプログラミングで、ELG グループ/主軸同期ペアの起動をおこないます		+	-	PGAsl
COUPRES	P	ELG グループのリセット		+	-	PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
CP [6]	G	軌跡移動	M	+		PGAsl
CPBC	K	汎用連結:ブロック切り替え条件		+	+	FB3sl (M3)
CPDEF	K	汎用連結:連結モジュールの作成		+	+	FB3sl (M3)
CPDEL	K	汎用連結:連結モジュールの削除		+	+	FB3sl (M3)
CPFMOF	K	汎用連結:完全なスイッチオフでのスレーブ軸の動作		+	+	FB3sl (M3)
CPFMON	K	汎用連結:スイッチオンでのスレーブ軸の動作		+	+	FB3sl (M3)
CPFMSON	K	汎用連結:同期モード		+	+	FB3sl (M3)
CPFPOS	K	汎用連結:スレーブ軸の同期位置		+	+	FB3sl (M3)
CPFRS	K	汎用連結:座標系基準システム		+	+	FB3sl (M3)
CPLA	K	汎用連結:マスタ軸の定義		+	-	FB3sl (M3)
CPLCTID	K	汎用連結:カーブテーブルの番号		+	+	FB3sl (M3)
CPLDEF	K	汎用連結:マスタ軸の定義と連結モジュールの作成		+	+	FB3sl (M3)
CPLDEL	K	汎用連結:連結モジュールのマスタ軸の削除		+	+	FB3sl (M3)
CPLDEN	K	汎用連結:連結係数の分母		+	+	FB3sl (M3)
CPLINSC	K	汎用連結:マスタ軸の入力値の単位変換係数		+	+	FB3sl (M3)
CPLINTR	K	汎用連結:マスタ軸の入力値のオフセット値		+	+	FB3sl (M3)
CPLNUM	K	汎用連結:連結係数の分子		+	+	FB3sl (M3)
CPLOF	K	汎用連結:連結モジュールのマスタ軸のスイッチオフ		+	+	FB3sl (M3)
CPLON	K	汎用連結:連結モジュールのマスタ軸のスイッチオン		+	+	FB3sl (M3)
CPLOUTSC	K	汎用連結:連結の出力値の単位変換係数		+	+	FB3sl (M3)
CPLOUTTR	K	汎用連結:連結の出力値のオフセット値		+	+	FB3sl (M3)
CPLPOS	K	汎用連結:マスタ軸の同期位置		+	+	FB3sl (M3)
CPLSETVAL	K	汎用連結:連結基準		+	+	FB3sl (M3)

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
CPMALARM	K	汎用連結:特別な連結関連アラーム出力のマスク		+	+	FB3sl (M3)
CPMBRAKE	K	汎用連結:特定の停止信号と停止命令に対するスレーブ軸の応答		+	-	FB3sl (M3)
CPMPRT	K	汎用連結:プログラムテストによるブロック検索実行での、パートプログラム開始時の連結動作		+	+	FB3sl (M3)
CPMRESET	K	汎用連結:RESET に対する連結動作		+	+	FB3sl (M3)
CPMSTART	K	汎用連結:パートプログラム開始時の連結動作		+	+	FB3sl (M3)
CPMVDI	K	汎用連結:特定の NC/PLC インターフェース信号に対する、スレーブ軸の応答		+	+	FB3sl (M3)
CPOF	K	汎用連結:連結モジュールのスイッチオフ		+	+	FB3sl (M3)
CPON	K	汎用連結:連結モジュールのスイッチオン		+	+	FB3sl (M3)
CPRECOF [6]	G	プログラマブル輪郭精度を解除します	M	+		PGAsl
CPRECON	G	プログラマブル輪郭精度を有効にします	M	+		PGAsl
CPRES	K	汎用連結:同期主軸連結の設定データを有効化		+	-	FB3sl (M3)
CPROT	P	チャネル別プロテクションゾーンの有効化/無効化		+	-	PGAsl
CPROTDEF	P	チャネル別プロテクションゾーンの定義		+	-	PGAsl
CPSETTYPE	K	汎用連結:連結タイプ		+	+	FB3sl (M3)
CPSYNCOP	K	汎用連結:位置同期制御「汎用」のしきい値		+	+	FB3sl (M3)
CPSYNCOP2	K	汎用連結:位置同期制御「汎用」のしきい値 2		+	+	FB3sl (M3)
CPSYNCOV	K	汎用連結:速度同期制御「汎用」のしきい値		+	+	FB3sl (M3)

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
CPSYNFIP	K	汎用連結:位置同期制御「精密」のしきい値		+	+	FB3sl (M3)
CPSYNFIP2	K	汎用連結:位置同期制御「精密」のしきい値 2		+	+	FB3sl (M3)
CPSYNFIV	K	汎用連結:速度同期制御「精密」のしきい値		+	+	FB3sl (M3)
CR	A	円弧半径	s	+		PGsl
CROT	F	現在の座標系の回転		+	-	PGAsl
CROTS	F	立体角によるプログラマブルフレームの回転(指定軸で回転)	s	+	-	PGsl
CRPL	F	任意の平面のフレーム回転		+	-	FB1sl (K2)
CSCALE	F	複数軸のスケーリング係数		+	-	PGAsl
CSPLINE	F	3 次スプライン	M	+		PGAsl
CT	G	接線方向の遷移をおこなう円弧	M	+		PGsl
CTAB	F	カーブテーブルからマスタ軸位置に応じたスレーブ軸位置を定義		+	+	PGAsl
CTABDEF	P	テーブルの定義の有効化		+	-	PGAsl
CTABDEL	P	カーブテーブルのクリア		+	-	PGAsl
CTABEND	P	テーブルの定義の無効化		+	-	PGAsl
CTABEXISTS	F	番号 n のカーブテーブルをチェック		+	+	PGAsl
CTABFNO	F	メモリで引き続き有効なカーブテーブルの数		+	+	PGAsl
CTABFPOL	F	メモリで引き続き有効な多項式の数		+	+	PGAsl
CTABFSEG	F	メモリで引き続き有効なカーブセグメントの数		+	+	PGAsl
CTABID	F	n 番目のカーブテーブルのテーブル番号を返します		+	+	PGAsl
CTABINV	F	カーブテーブルからスレーブ軸位置に応じたマスタ軸位置の定義		+	+	PGAsl
CTABISLOCK	F	番号 n のカーブテーブルのロック状態を返します		+	+	PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
CTABLOCK	P	解除と上書きのロック		+	+	PGAsl
CTABMEMTYP	F	カーブテーブル番号 n が作成されたメモリを返します。		+	+	PGAsl
CTABMPOL	F	メモリで引き続き有効な多項式の最大数		+	+	PGAsl
CTABMSEG	F	メモリで引き続き有効なカーブセグメントの最大数		+	+	PGAsl
CTABNO	F	SRAM または DRAM の、定義したカーブテーブルの合計数		+	+	FB3sl (M3)
CTABNOMEM	F	SRAM または DRAM の、定義したカーブテーブルの合計数		+	+	PGAsl
CTABPERIOD	F	番号 n のカーブテーブルのテーブルの周期性を返します		+	+	PGAsl
CTABPOL	F	メモリですでに使用している多項式の数		+	+	PGAsl
CTABPOLID	F	番号 n のカーブテーブルが使用するカーブ多項式の数		+	+	PGAsl
CTABSEG	F	メモリですでに使用しているカーブセグメントの数		+	+	PGAsl
CTABSEGID	F	番号 n のカーブテーブルが使用するカーブセグメントの数		+	+	PGAsl
CTABSEV	F	カーブテーブルのセグメントのスレーブ軸の最終値を返します		+	+	PGAsl
CTABSSV	F	カーブテーブルのセグメントのスレーブ軸の初期値を返します		+	+	PGAsl
CTABTEP	F	カーブテーブル終了位置のマスタ軸の値を返します		+	+	PGAsl
CTABTEV	F	カーブテーブル終了位置のスレーブ軸の値を返します		+	+	PGAsl
CTABTMAX	F	カーブテーブルのスレーブ軸の最大値を返します		+	+	PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
CTABTMIN	F	カーブテーブルのスレーブ軸の最小値を返します		+	+	PGAsl
CTABTSP	F	カーブテーブル開始位置のマスタ軸の値を返します		+	+	PGAsl
CTABTSV	F	カーブテーブル開始位置のスレーブ軸の値を返します		+	+	PGAsl
CTABUNLOCK	P	削除のロックと上書きのロックを無効にします		+	+	PGAsl
CTOL	A	コンプレッサ機能、旋回スムージング、およびスムージングタイプの輪郭許容範囲	M	+		PGAsl
CTRANS	F	複数軸のゼロオフセット		+	-	PGAsl
CUT2D [6]	G	2 次元 TRC	M	+		PGsl
CUT2DD	G	差分工具に対する 2½ 次元 TRC	M	+		PGsl
CUT2DF	G	現在のフレームに対する 2 次元 TRC(傾斜面)	M	+		PGsl
CUT2DFD	G	現在のフレームを基準にした差分工具に対する 2½ 次元 TRC (傾斜面)	M	+		PGsl
CUT3DC	G	外周加工の 3 次元 TRC	M	+		PGAsl
CUT3DCC	G	3 次元径補正の限界面を考慮した外周加工の 3 次元 TRC: 加工面の輪郭	M	+		PGAsl
CUT3DCCD	G	工具点中心軌跡の差分工具の限界面を考慮した外周加工の 3 次元 TRC: 限界面の切削	M	+		PGAsl
CUT3DCD	G	外周加工の差分工具に対する 3 次元 TRC	M	+		PGAsl
CUT3DF	G	向きが変化する正面削りの 3 次元 TRC	M	+		PGAsl
CUT3DFD	G	向きが変化する正面削りの差分工具に対する 3 次元 TRC	M	+		PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
CUT3DFF	G	向きが一定の正面削りの 3 次元 TRC 工具オリエンテーションは、G17 ～ G19 で定義した方向で、場合によっては、フレームによって回転します。	M	+		PGAsl
CUT3DFS	G	向きが一定の正面削りの 3 次元 TRC 工具オリエンテーションは、G17 ～ G19 で定義され、フレームには影響されません。	M	+		PGAsl
CUTCONOF [6]	G	工具径補正の無効化	M	+		PGsl
CUTCONON	G	工具径補正の有効化	M	+		PGsl
CUTMOD	A	旋回工具のオフセットデータ変更の起動(方向付け可能な工具ホルダと接続)	M	+		PGAsl
CUTMODK	A	旋回工具のオフセットデータ変更の起動(キネマティック結合により定義された方向変換と接続)	M	+		PGAsl
CYCLE60	C (T)	文字彫りサイクル		+		PGAsl
CYCLE61	C (T)	正面削り		+		PGAsl
CYCLE62	C (T)	輪郭呼び出し		+		PGAsl
CYCLE63	C (T)	輪郭ポケットフライス加工		+		PGAsl
CYCLE64	C (T)	輪郭ポケットの前加工		+		PGAsl
CYCLE70	C (T)	ねじ切り		+		PGAsl
CYCLE72	C (T)	軌跡のフライス削り		+		PGAsl
CYCLE76	C (T)	長方形スピゴットのフライス削り		+		PGAsl
CYCLE77	C (T)	円形スピゴットのフライス削り		+		PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
CYCLE78	C (T)	ねじのフライス削り		+		PGAsl
CYCLE79	C (T)	ポリゴン		+		PGAsl
CYCLE81	C (T)	穴あけ、センタリング		+		PGAsl
CYCLE82	C (T)	穴あけ、座ぐりフライス		+		PGAsl
CYCLE83	C (T)	深穴ドリル		+		PGAsl
CYCLE84	C (T)	フローティングチャックのないタッピング		+		PGAsl
CYCLE85	C (T)	リーマ加工		+		PGAsl
CYCLE86	C (T)	ボーリング		+		PGAsl
CYCLE92	C (T)	突っ切り		+		PGAsl
CYCLE95	C (T)	輪郭に沿った切削		+		PGAsl
CYCLE98	C (T)	連続ねじ		+		PGAsl
CYCLE99	C (T)	ねじ切り		+		PGAsl
CYCLE150	C (M)	計測結果の表示/ログ		+		BNMsl
CYCLE435	C (T)	目立て工具位置の計算		+		PGAsl
CYCLE495	C (T)	成形研削		+		PGAsl
CYCLE750	C (A)	CYCLE751...の内部動作サイクル CYCLE759 (実際のファンクションコールに対する MMC 命令を含む)		-		FB3sl (T4)

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
CYCLE751	C (A)	オートチューニングセッションを開く/実行/閉じる		M		FB3sl (T4)
CYCLE752	C (A)	オートチューニングセッションへの軸の追加		M		FB3sl(T4)
CYCLE753	C (A)	オートチューニングモードの選択		M		FB3sl (T4)
CYCLE754	C (A)	言語ブロックの追加/削除		M		FB3sl (T4)
CYCLE755	C (A)	データブロックのバックアップ/リストア		M		FB3sl (T4)
CYCLE756	C (A)	オートチューニング結果の有効化		M		FB3sl (T4)
CYCLE757	C (A)	オートチューニングデータの保存		M		FB3sl (T4)
CYCLE758	C (A)	パラメータ値の変更		M		FB3sl (T4)
CYCLE759	C (A)	パラメータ値の読み取り		M		FB3sl (T4)
CYCLE800	C (T)	旋回		+		PGAsl
CYCLE801	C (T)	格子またはフレーム		+		PGAsl
CYCLE802	C (T)	任意の位置		+		PGAsl
CYCLE830	C (T)	深穴ドリル 2		+		PGAsl
CYCLE832	C (T)	高速設定		+		PGAsl
CYCLE840	C (T)	フローティングチャックを使用したタッピング		+		PGAsl
CYCLE899	C (T)	オープン溝のフライス削り		+		PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
CYCLE930	C (T)	溝		+		PGAsl
CYCLE940	C (T)	アンダーカット形状		+		PGAsl
CYCLE951	C (T)	切削		+		PGAsl
CYCLE952	C (T)	輪郭の溝削り		+		PGAsl
CYCLE961	C (M)	ワークのコーナー(内側または外側)の位置を特定し、ゼロオフセットとして挿入します。		+		BNMsl
CYCLE971	C (M)	工具プローブの校正、工具計測の長さや半径(フライス削りのみ)		+		BNMsl
CYCLE973	C (M)	ワークの表面または溝でワークプローブを校正(旋削のみ)		+		BNMsl
CYCLE974	C (M)	選択した計測軸でワーク原点を決定し、1 点計測で工具補正を決定します。		+		BNMsl
CYCLE976	C (M)	特定の軸および方向の端または作業平面の校正リングまたは校正ボールでワークプローブを校正		+		BNMsl
CYCLE977	C (M)	平面の直径、幅、中心を決定		+		BNMsl
CYCLE978	C (M)	この計測サイクルは、ワーク座標系で端面の位置を計測するために使用します。		+		BNMsl
CYCLE979	C (M)	平面の中心を決定して円弧の半径を計測します。		+		BNMsl
CYCLE982	C (M)	工具プローブの校正、旋削穴あけおよびフライス工具の計測(旋削のみ)		+		BNMsl
CYCLE994	C (M)	選択した計測軸で、2 点計測でワーク原点を決定します(旋削のみ)。		+		BNMsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
CYCLE995	C (M)	工作機械の主軸の角度の計測		+		BNMsl
CYCLE996	C (M)	回転軸とキネマティックトランスフォーメーションの座標変換データの特定		+		BNMsl
CYCLE997	C (M)	ボールの直径と中心を特定し、配置された 3 個のボールの中心を測定		+		BNMsl
CYCLE998	C (M)	作業平面で参照される平面の回転位置を特定し、ワーク座標系の端面の角度を特定します。		+		BNMsl
CYCLE4071	C (T)	反転点での切り込みによる長手方向の研削		+		PGAsl
CYCLE4072	C (T)	反転点での切り込みによる長手方向の研削とキャンセル信号		+		PGAsl
CYCLE4073	C (T)	連続切り込みによる長手方向の研削		+		PGAsl
CYCLE4074	C (T)	連続切り込みによる長手方向の研削とキャンセル信号		+		PGAsl
CYCLE4075	C (T)	反転点での切り込みによる平面研削		+		PGAsl
CYCLE4077	C (T)	反転点での切り込みによる平面研削とキャンセル信号		+		PGAsl
CYCLE4078	C (T)	連続切り込みによるる平面研削		+		PGAsl
CYCLE4079	C (T)	断続的な切り込みによる平面研削		+		PGAsl

命令 D ... F

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1][2][3][4][5] 詳細は、一覧 (ページ 1059)を参照してください。						
d	O	工具オフセット番号		+		PGsl
D0	O	D0 の場合、工具のオフセットは無効		+		PGsl
DAC	K	ノンモーダルのアブソリュート指令の軸別直径指定	s	+		PGsl
DC	K	回転軸のアブソリュート指令、位置への直接アプローチ	s	+		PGsl
DCI	K	データクラス I (= Individual)の割り当て (SINUMERIK 828D のみ)		+		PGAsl
DCM	K	データクラス M (= Manufacturer)の割り当て(SINUMERIK 828D のみ)		+		PGAsl
DCU	K	データクラス U (= User)の割り当て (SINUMERIK 828D のみ)		+		PGAsl
DEF	K	変数定義		+		PGAsl
DEFAULT	K	CASE 分岐の分岐		+		PGAsl
DEFINE	K	マクロ定義のキーワード		+		PGAsl
DELAYFSTOF	P	停止遅延領域の終了区間の定義	M	+	-	PGAsl
DELAYFSTON	P	停止遅延区間の開始区間の定義	M	+	-	PGAsl
DELDL	F	追加オフセットの削除		+	-	PGAsl
DELDTG	P	残移動距離削除		-	+	FBSYsl
DELETE	P	指定したファイルの削除。ファイル名称は、パスとファイル識別子を使用して指定できます。		+	-	PGAsl
DELMLOWNER	F	工具の所有マガジンロケーションの削除		+	-	FBWsl
DELMLRES	F	マガジンロケーション予約の削除		+	-	FBWsl
DELMT	P	マルチ工具の削除		+	-	FBWsl
DELOBJ	F	キネマティック結合、プロテクションゾーン、プロテクションゾーン要素、干渉ペア、座標変換データからの要素の削除		+		PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1][2][3][4][5] 詳細は、一覧 (ページ 1059)を参照してください。						
DELT	P	工具の削除		+	-	FBWsl
DELTC	P	工具ホルダデータレコードの削除		+	-	FBWsl
DELTOOLENV	F	工具環境を記述したデータの削除		+	-	PGAsl
DIACYCOFA	K	軸別モーダル直径指定:サイクル内はオフ	M	+		FB1sl (P1)
DIAM90	G	G90 では直径指定、G91 では半径指定	M	+		PGAsl
DIAM90A	K	G90 と AC では軸別モーダル直径指定、G91 と IC では半径指定	M	+		PGsl
DIAMCHAN	K	すべての軸をマシンデータの軸機能から直径/半径指定チャネル状態へ移行		+		PGsl
DIAMCHANA	K	直径/半径指定チャネル状態へ移行		+		PGsl
DIAMCYCOF	G	チャネル別の直径指定:サイクル内はオフ	M	+		FB1sl (P1)
DIAMOF[6]	G	直径指定:オフ 通常の位置指定、工作機械メーカに問い合わせてください	M	+		PGsl
DIAMOFA	K	軸別モーダル直径指定:オフ 通常の位置指定、工作機械メーカに問い合わせてください	M	+		PGsl
DIAMON	G	直径指定:ON	M	+		PGsl
DIAMONA	K	軸別モーダル直径指定:オン 適用、工作機械メーカに問い合わせてください	M	+		PGsl
DIC	K	ノンモーダルのインクレメンタル指令の軸別直径指定	s	+		PGsl
DILF	O	後退距離(長さ)	M	+		PGsl
DISABLE	P	割り込みのオフ		+	-	PGAsl
DISC	O	工具径補正の挿入円のオーバーシュート	M	+		PGsl
DISCL	O	高速切り込み移動の終点と加工平面の距離		+		PGsl
DISPLOF	PA	実行中のブロック表示をマスクします		+		PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1][2][3][4][5] 詳細は、一覧 (ページ 1059)を参照してください。						
DISPLON	PA	実行中のブロック表示のマスクは無効です		+		PGAsl
DISPR	O	再位置決めの軌跡距離	s	+		PGAsl
DISR	O	再位置決めの距離	s	+		PGAsl
DISRP	O	滑らかなアプローチと後退時の、後退平面と加工平面との間の距離		+		PGsl
DITE	O	ねじの切り上げの軌跡	M	+		PGsl
DITS	O	ねじの切り始めの軌跡	M	+		PGsl
DIV	K	整数の除算		+		PGAsl
DI	O	ロケーションに応じた追加工具補正の選択(DL、合計セットアップオフセット)	M	+		PGAsl
DO	O	シンクロナイズドアクションのキーワード。条件を満たすと、動作を起動します		-	+	FBSYsl
DRFOF	P	ハンドルオフセット(DRF)の解除	M	+	-	PGsl
DRIVE	G	速度に応じた軌跡加減速度	M	+		PGAsl
DRIVEA	P	プログラム指令軸で膝形加減速特性を起動します。		+	-	PGAsl
DYNFINISH	G	滑らかな仕上げのダイナミック応答	M	+		PGAsl
DYNNORM[6]	G	標準ダイナミック応答	M	+		PGAsl
DYNPOS	G	位置決めモードのダイナミック応答、タッピング	M	+		PGAsl
DYNROUGH	G	荒削りのダイナミック応答	M	+		PGAsl
DYNSEMIFIN	G	仕上げのダイナミック応答	M	+		PGAsl
DZERO	P	工具オフセットユニットのすべての D 番号を無効としてマーク		+	-	PGAsl
EAUTO	G	最後の 3 点による最後のスプライン区間の定義	M	+		PGAsl
EGDEF	P	電子ギヤの定義		+	-	PGAsl
EGDEL	P	スレーブ軸の連結定義の解除		+	-	PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1][2][3][4][5] 詳細は、一覧 (ページ 1059)を参照してください。						
EGOFC	P	電子ギヤの一括解除		+	-	PGAsl
EGOFS	P	電子ギヤの選択解除		+	-	PGAsl
EGON	P	電子ギヤの起動		+	-	PGAsl
EGONSYN	P	電子ギヤの起動		+	-	PGAsl
EGONSYNE	P	アプローチモードを指定して電子ギヤを起動		+	-	PGAsl
ELSE	K	IF 条件を満たさない場合のプログラムの分岐		+		PGAsl
ENABLE	P	割り込みのオン		+	-	PGAsl
ENAT[6]	G	次の移動ブロックへの自然遷移	M	+		PGAsl
ENDFOR	K	FOR カウンタループの終了行		+		PGAsl
ENDIF	K	IF 分岐の終了行		+		PGAsl
ENDLABEL の間にあるパートプログラム区間	K	REPEAT によるパートプログラムの繰り返しの終了ラベル		+		PGAsl、FB1sl (K1)
ENDLOOP	K	プログラムの無限ループ LOOP の終了行		+		PGAsl
ENDPROC	K	開始行 PROC のあるプログラムの終了行		+		
ENDWHILE	K	WHILE ループの終了行		+		PGAsl
ESRR	P	ドライブのドライブ自律 ESR 後退のパラメータ設定		+		PGAsl
ESRS	P	ドライブのドライブ自律 ESR 停止のパラメータ設定		+		PGAsl
ETAN	G	スプライン開始位置での次の移動ブロックへの接線方向の遷移	M	+		PGAsl
EVERY	K	FALSE から TRUE への条件の遷移時にシンクロナイズドアクションを実行		-	+	FBSYsl
EX	K	指数表記の値の割り当てのキーワード		+		PGAsl
EXECSTRING	P	実行するパートプログラム行を含む文字列変数の転送		+	-	PGAsl
EXECTAB	P	移動テーブルから要素を実行		+	-	PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1][2][3][4][5] 詳細は、一覧 (ページ 1059)を参照してください。						
EXECUTE	P	プログラム実行のオン		+	-	PGAsl
EXP	F	指数関数 ex		+	+	PGAsl
EXTCALL	O	外部サブプログラムの実行		+	+	PGAsl
EXTCLOSE	P	書き込みのために開かれていた外部機器/ファイルを閉じる		+	-	PGAsl
EXTERN	K	パラメータ転送を含むサブプログラムの宣言		+		PGAsl
EXTOPEN	P	チャネルの外部機器/ファイルを書き込みのために開く		+	-	PGAsl
F	O	送り速度値 (G4 と組み合わせて、ドウェル時間もFでプログラム指令します)		+	+	PGsl
FA	K	軸送り速度	M	+	+	PGsl
FAD	O	滑らかなアプローチと後退の送り速度		+		PGsl
FALSE	K	論理定数:不正		+	+	PGAsl
FB	O	ノンモーダル送り速度		+		PGsl
FCTDEF	P	多項式関数の定義		+	-	PGAsl
FCUB	G	3 次スプラインによる可変送り速度	M	+		PGAsl
FD	O	ハンドルオーバライドの軌跡送り速度	s	+		PGsl
FDA	K	ハンドルオーバライドの軸送り速度	s	+		PGsl
FENDNORM[6]	G	コーナー減速のオフ	M	+		PGAsl
FFWOF[6]	G	フィードフォワード制御のオフ	M	+		PGAsl
FFWON	G	フィードフォワード制御のオン	M	+		PGAsl
FGREF	K	回転軸の基準半径または旋回軸の軌跡基準係数(ベクトル補間)	M	+		PGsl
FGROUP	P	軌跡送り速度を使用する単数/複数の軸の定義		+	-	PGsl
FI	K	フレームデータにアクセスするためのパラメータ:仕上げオフセット		+		PGAsl
FIFOCTRL	G	先読みバッファの制御	M	+		PGAsl

命令	タイプ 1)	意味	W 2)	TP3)	SA4)	説明の参照先 5)
1)2)3)4)5) 詳細は、一覧 (ページ 1059)を参照してください。						
FILEDATE	P	ファイルへの最後の書き込みアクセスの日付を返します		+	-	PGAsl
FILEINFO	P	FILEDATE、FILESIZE、FILESTAT、および FILETIME を含む要約情報を返します		+	-	PGAsl
FILESIZE	P	現在のファイルサイズを返します		+	-	PGAsl
FILESTAT	P	読み出し、書き込み、実行、表示、削除(rwxsd)の権限に関するファイル状態を返します		+	-	PGAsl
FILETIME	P	ファイルへの最後の書き込みアクセスの時刻を返します		+	-	PGAsl
FINEA	K	「精密イグザクトストップ」へ到達時に移動終了	M	+		PGAsl
FL	K	同期軸の速度制限	M	+		PGsl
FLIN	G	可変毎分送り速度	M	+		PGAsl
FMA	K	複数の軸送り速度	M	+		PGsl
FNORM 6)	G	DIN 66025 に準拠した標準送り速度	M	+		PGAsl
FOC	K	ノンモーダルトルク/推力の制限	s	-	+	FBSYsl
FOCOF	K	モーダルトルク/推力制限のスイッチオフ	M	-	+	FBSYsl
FOCON	K	モーダルトルク/推力制限のオン	M	-	+	FBSYsl
FOR	K	実行回数が一定のカウンタループ		+		PGAsl
FP	O	固定点:アプローチする固定点の番号	s	+		PGsl
FPO	K	多項式でプログラム指令した送り速度特性		+		PGAsl
FPR	P	回転軸識別子		+	-	PGsl
FPRAOF	P	毎回転送り速度の解除		+	-	PGsl
FPRAON	P	毎回転送り速度の起動		+	-	PGsl
FRAME	K	座標系の定義のデータタイプ		+		PGAsl
FRC	O	丸み付けと面取りの送り速度	s	+		PGsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1][2][3][4][5] 詳細は、一覧 (ページ 1059)を参照してください。						
FRCM	O	丸み付けと面取りの送り速度、モーダル	M	+		PGsl
FROM	K	条件が一度満たされると、シンクロナイズドアクションが動作中であるかぎり、動作を実行します		-	+	FBSYsl
FTOC	P	精密工具補正の変更		-	+	FBSYsl
FTOCOF [6]	G	オンライン精密工具補正のオフ	M	+		PGAsl
FTOCON	G	オンライン精密工具補正のオン	M	+		PGAsl
FXS	K	突き当て点停止のオン	M	+	+	PGsl
FXST	K	突き当て点停止のトルク制限	M	+	+	PGsl
FXSW	K	突き当て点停止の監視範囲		+	+	PGsl
FZ	K	1 刃当り送り速度	M	+		PGsl

命令 G ... L

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
G0	G	早送りによる直線補間(早送り移動)	M	+		PGsl
G1 [6]	G	送り速度による直線補間(直線補間)	M	+		PGsl
G2	G	右回りの円弧補間	M	+		PGsl
G3	G	左回りの円弧補間	M	+		PGsl
G4	G	ドウェル時間、事前設定	s	+		PGsl
G5	G	傾斜プランジ研削	s	+		PGAsl
G7	G	傾斜プランジ研削時の補正動作	s	+		PGAsl
G9	G	イグザクトストップ - 減速	s	+		PGsl
G17 [6]	G	作業平面 X/Y の選択	M	+		PGsl
G18	G	作業平面 Z/X の選択	M	+		PGsl
G19	G	作業平面 Y/Z の選択	M	+		PGsl
G25	G	ワーキングエリアリミットの下限	s	+		PGsl
G26	G	ワーキングエリアリミットの上限	s	+		PGsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
G33	G	固定リードのねじ切り	M	+		PGsl
G34	G	単調増加可変リードねじ切り	M	+		PGsl
G35	G	単調減少可変リードねじ切り	M	+		PGsl
G40 [6]	G	工具径補正のオフ	M	+		PGsl
G41	G	輪郭の左側の工具径補正	M	+		PGsl
G42	G	輪郭の右側の工具径補正	M	+		PGsl
G53	G	現在のゼロオフセットのマスク(ノンモーダル)	s	+		PGsl
G54	G	1 番目の設定可能ゼロオフセット	M	+		PGsl
G55	G	2 番目の設定可能ゼロオフセット	M	+		PGsl
G56	G	3 番目の設定可能ゼロオフセット	M	+		PGsl
G57	G	4 番目の設定可能ゼロオフセット	M	+		PGsl
G58 (840D sl)	G	アブソリュートプログラマブルゼロオフセット(荒削りオフセット)	s	+		PGsl
G58 (828D)	G	5 番目の設定可能ゼロオフセット	M	+		PGsl
G59 (840D sl)	G	追加のプログラマブルゼロオフセット(仕上げオフセット)	s	+		PGsl
G59 (828D)	G	6 番目の設定可能ゼロオフセット	M	+		PGsl
G60 [6]	G	イグザクトストップ - 減速	M	+		PGsl
G62	G	工具径補正が有効なときの内側コーナのコーナー減速(G41、G42)	M	+		PGAsl
G63	G	フローティングチャックを使用したタッピング	s	+		PGsl
G64	G	連続軌跡モード	M	+		PGsl
G70	G	ジオメトリ指定のインチ寸法(長さ)	M	+	+	PGsl
G71 [6]	G	ジオメトリ指定のメトリック寸法(長さ)	M	+	+	PGsl
G74	G	レファレンス点復帰	s	+		PGsl
G75	G	固定点アプローチ	s	+		PGsl
G90 [6]	G	アブソリュート指令	m/s	+		PGsl
G91	G	インクレメンタル指令	m/s	+		PGsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
G93	G	インバースタイム送り(/min)	M	+		PGsl
G94 [6]	G	毎分送り速度 F(mm/min または inch/min、および°/min 単位)	M	+		PGsl
G95	G	毎回転送り速度 F(mm/rev または inch/rev 単位)	M	+		PGsl
G96	G	毎回転送り速度(G95 に対して)および収束一定制御	M	+		PGsl
G97	G	毎回転送り速度および一定主軸速度(周速一定制御オフ)	M	+		PGsl
G110	G	最後のプログラム指令位置に対する極のプログラミング	s	+		PGsl
G111	G	現在のワーク座標系の原点に対する極のプログラミング	s	+		PGsl
G112	G	最後に有効な極に対する極のプログラミング	s	+		PGsl
G140 [6]	G	G41/G42 で定義した SAR アプローチ方向	M	+		PGsl
G141	G	輪郭の左側への SAR アプローチ方向	M	+		PGsl
G142	G	輪郭の右側への SAR アプローチ方向	M	+		PGsl
G143	G	接線に応じて SAR アプローチ方向を決定	M	+		PGsl
G147	G	直線による滑らかなアプローチ	s	+		PGsl
G148	G	直線による滑らかな後退	s	+		PGsl
G153	G	基本フレームを含む現在のフレームのマスク	s	+		PGsl
G247	G	4 分円による滑らかなアプローチ	s	+		PGsl
G248	G	4 分円による滑らかな後退	s	+		PGsl
G290 [6]	G	SINUMERIK モードへの切り替えのオン	M	+		FBWsl
G291	G	ISO2/3 モードへの切り替えのオン	M	+		FBWsl
G331	G	リジッドタッピング、正符号のリード、右回り	M	+		PGsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
G332	G	リジッドタッピング、負符号のリード、左回り	M	+		PGsl
G335	G	右回りでの凸ねじの旋削	M	+		PGsl
G336	G	左回りでの凸ねじの旋削	M	+		PGsl
G340 [6]	G	空間アプローチブロック(平面上と奥行きへ同時に移動(ヘリカル))	M	+		PGsl
G341	G	最初に垂直軸(z)の切り込み、その後平面内にアプローチ	M	+		PGsl
G347	G	半円による滑らかなアプローチ	s	+		PGsl
G348	G	半円による滑らかな後退	s	+		PGsl
G450 [6]	G	遷移円	M	+		PGsl
G451	G	等距離の交点	M	+		PGsl
G460 [6]	G	アプローチと後退ブロックの干渉検出の適用	M	+		PGsl
G461	G	TRC ブロックへの円弧の挿入	M	+		PGsl
G462	G	TRC ブロックへの直線の挿入	M	+		PGsl
G500 [6]	G	すべての設定可能フレームの解除、基本フレームは有効です	M	+		PGsl
G505 ... G599	G	5 ... 99 番目の設定可能なゼロオフセット	M	+		PGsl
G601 [6]	G	精密イグザクトストップでブロック切り替え	M	+		PGsl
G602	G	汎用イグザクトストップでブロック切り替え	M	+		PGsl
G603	G	IPO のブロック終点でブロック切り替え	M	+		PGsl
G621	G	すべてのコーナーのコーナー減速	M	+		PGAsl
G641	G	距離条件に応じたスムージングによる連続軌跡モード(=プログラム指令可能な丸み付け隙間)	M	+		PGsl
G642	G	定義した許容範囲内のスムージングをおこなう連続軌跡モード	M	+		PGsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
G643	G	定義した許容範囲内のスムージングをおこなう連続軌跡モード(ブロック内部)	M	+		PGsl
G644	G	最大ダイナミック応答でスムージングをおこなう連続軌跡モード	M	+		PGsl
G645	G	スムージング、および定義許容範囲内で接線方向のブロック遷移をおこなう連続軌跡モード	M	+		PGsl
G700	G	ジオメトリ仕様と加工仕様(長さ、送り速度) のインチ寸法	M	+	+	PGsl
G710 [6]	G	ジオメトリ仕様と加工仕様(長さ、送り速度)のメトリック寸法	M	+	+	PGsl
G810 [6]、...、G819	G	OEM ユーザー用に予約された G グループ		+		PGAsl
G820 [6]、...、G829	G	OEM ユーザー用に予約された G グループ		+		PGAsl
G931	G	移動時間により指定された送り速度、一定軌跡速度を解除	M	+		
G942	G	毎分送り速度と周速一定制御、または主軸速度の解除	M	+		
G952	G	毎回転送り速度と周速一定制御、または主軸速度の解除	M	+		
G961	G	毎分送り速度(G94 に対して)および周速一定制御	M	+		PGsl
G962	G	毎分送り速度と毎回転送り速度、および周速一定制御	M	+		PGsl
G971	G	毎分送り速度および一定主軸速度(周速一定制御オフ)	M	+		PGsl
G972	G	毎分送り速度または毎回転送り速度および一定主軸速度(周速一定制御オフ)	M	+		PGsl
G973	G	主軸速度制限のない毎回転送り速度および一定主軸速度(ISO モードでは、LIMS なしの G97)	M	+		PGsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
GEOAX	P	ジオメトリ軸 1 ～ 3 に新しいチャネル軸を割り当て		+	-	PGAsl
GET	P	チャネル間で有効な軸の入れ替え		+	+	PGAsl
GETACTT	F	同じ名称の工具グループから、動作中の工具を取得		+	-	FBWsl
GETACTTD	F	絶対 D 番号に関連する T 番号を取得		+	-	PGAsl
GETD	P	チャネル間で軸の直接入れ替え		+	-	PGAsl
GETDNO	F	工具(T)の刃先(CE)の D 番号を返します		+	-	PGAsl
GETEXET	P	ロードした T 番号の読み出し		+	-	FBWsl
GETFREELOC	P	当該の工具用のマガジンの空きスペースを検出		+	-	FBWsl
GETSELT	P	選択した T 番号を返します		+	-	FBWsl
GETT	F	工具名称に対応する T 番号を取得		+	-	FBWsl
GETTCOR	F	工具長または工具長成分、またはその両方の読み出し		+	-	PGAsl
GETTENV	F	T 番号、D 番号、および DL 番号の読み出し		+	-	PGAsl
GETVARAP	F	システム/ユーザー変数への読み取りアクセス権		+	-	PGAsl
GETVARDFT	F	システム/ユーザー変数の読み取り初期値		+	-	PGAsl
GETVARLIM	F	システム/ユーザー変数の読み取り制限値		+	-	PGAsl
GETVARPHU	F	システム/ユーザー変数の読み取り物理単位		+	-	PGAsl
GETVARTYP	F	システム/ユーザー変数の読み取りデータタイプ		+	-	PGAsl
GFRAME0 ... GFRAME100	G	チャネル内のデータ管理の研削フレーム<n>の起動	M	+		PGsl
GOTO	K	最初に前方に、次に後方に(最初にプログラム末尾方向、次にプログラム先頭方向に)ジャンプする命令		+		PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
GOTOB	K	後方(プログラム先頭方向)にジャンプ		+		PGAsl
GOTOC	K	GOTO と同じ。ただし、アラーム 14080「Jump destination not found」(ジャンプ先が見つかりません)はマスクされます		+		PGAsl
GOTOF	K	前方(プログラム末尾方向)にジャンプ		+		PGAsl
GOTOS	K	プログラム先頭へジャンプ		+		PGAsl
BP	K	位置属性の間接プログラミング用キーワード		+		PGAsl
GROUP_ ADDEND	C (T)	試験切削追加の終了		+		PGAsl
GROUP_BEGIN	C (T)	プログラムグループの先頭		+		PGAsl
GROUP_END	C (T)	プログラムグループの終了		+		PGAsl
GWPSOF	P	砥石周速度一定(GWPS)の選択解除	s	+	-	PGsl
GWPSON	P	砥石周速度一定制御(GWPS)の選択	s	+	-	PGsl
H...	O	PLC への補助機能出力		+	+	PGsl/FB1sl (H2)
HOLES1	C (T)	列並びの穴		+		PGAsl
HOLES2	C (T)	円弧並びの穴		+		PGAsl
I	O	補間パラメータ	s	+		PGsl
I1	O	中間点座標	s	+		PGsl
IC	K	インクレメンタル指令	s	+		PGsl
ICYCOF	P	テクノロジサイクルのすべてのブロックを、ICYCOF 指令後に 1 つの補間サイクルで処理します		+	+	FBSYsl
ICYCON	P	テクノロジサイクルの各ブロックを、ICYCON 指令後に個別の補間サイクルで処理します。		+	+	FBSYsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
ID	K	モーダルシンクロナイズドアクションの識別子	M	-	+	FBSYsl
IDS	K	モーダルの静的シンクロナイズドアクションの識別子		-	+	FBSYsl
IF	K	パートプログラム/テクノロジサイクルの条件分岐の開始		+	+	PGAsl
INDEX	F	入力文字列の文字のインデックスを定義		+	-	PGAsl
INICF	K	NEWCONF 変数の初期化		+		PGAsl
INIPO	K	電源投入時の変数の初期化		+		PGAsl
INIRE	K	リセット時の変数の初期化		+		PGAsl
INIT	P	特定のチャネルで実行する特定の NC プログラムの選択		+	-	PGAsl
INITIAL		すべての領域にわたる INI ファイルの生成		+		PGAsl
INT	K	データタイプ:符号付き整数		+		PGAsl
INTERSEC	F	2 個の輪郭要素の交点を計算		+	-	PGAsl
INVCCW	G	インボリュート曲線の起動、左回り	M	+		PGsl
INVCW	G	インボリュート曲線の起動、右回り	M	+		PGsl
INVFRAME	F	フレームから逆フレームを計算		+	-	FB1sl (K2)
IP	K	可変補間パラメータ		+		PGAsl
IPOBRKA	P	減速カーブ適用による移動条件	M	+	+	
IPOENDA	K	「補間停止」へ到達時に移動終了	M	+		PGAsl
IPTRLOCK	P	次の運転機能ブロックで検索不可プログラム区間の開始をおこないます。	M	+	-	PGAsl
IPTRUNLOCK	P	中断時に、実行中のブロックで検索不可プログラム区間の終了を設定します。	M	+	-	PGAsl
IR	O	凸ねじを旋削する際の円弧中心点の座標(X 軸)		+		PGsl
ISAXIS	F	ジオメトリ軸 1 がパラメータとして指定されているかどうかをチェック		+	-	PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
ISD	O	切り込み深さ	M	+		PGAsl
ISFILE	F	NC アプリケーションメモリにファイルがあるかどうかをチェック		+	-	PGAsl
ISNUMBER	F	入力文字列を数字に変換できるかどうかをチェック		+	-	PGAsl
ISOCALL	K	ISO 言語でプログラム指令したプログラムの間接呼び出し		+		PGAsl
ISVAR	F	NC で宣言された変数が転送パラメータに含まれるかどうかをチェック		+	-	PGAsl
J	O	補間パラメータ	s	+		PGsl
J1	O	中間点座標	s	+		PGsl
JERKA	P	プログラム指令軸に対して、マシンデータで設定した加減速動作を起動		+	-	
JERKLIM	K	最大軸加々速度の低減、または増加	M	+		PGAsl
JERKLIMA	K	最大軸加々速度の低減、または増加	M	+	+	PGAsl
JR	O	凸ねじを旋削する際の円弧中心点の座標(Y 軸)		+		PGsl
K	O	補間パラメータ	s	+		PGsl
K1	O	中間点座標	s	+		PGsl
KONT	G	工具補正時の、輪郭周りの移動	M	+		PGsl
KONTC	G	曲率の変化が連続的な多項式によるアプローチ/後退	M	+		PGsl
KONTT	G	連続の接線の多項式によるアプローチ/後退	M	+		PGsl
KR	O	凸ねじを旋削する際の円弧中心点の座標(Z 軸)		+		PGsl
L	O	サブプログラム番号	s	+	+	PGAsl
LEAD	O	リード角 1 番目の基本工具の向き 2 番目のオリエンテーション多項式	M	+		PGAsl
LEADOF	P	軸間連動機能のオフ		+	+	PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
LEADON	P	軸間連動機能のオン		+	+	PGAsl
LENTOAX	F	動作中の工具の工具長 L1、L2、および L3 の、横座標、縦座標、および垂直座標への割り当てに関する情報を提供		+	-	PGAsl
LFOF [6]	G	ねじ切りの高速リトラクトのオフ	M	+		PGsl
LFON	G	ねじ切りの高速リトラクトのオン	M	+		PGsl
LFPOS	G	POLFMASK または POLFMLIN を使用して宣言した軸の、POLF でプログラム指令したアブソリュート軸位置への後退	M	+		PGsl
LFTXT [6]	G	高速リトラクトの後退移動の平面を、軌跡タンジェントと現在の工具方向から特定します	M	+		PGsl
LFWP	G	高速リトラクトの後退移動の平面を現在の作業平面(G17/G18/G19)で特定します	M	+		PGsl
LIFTFAST	K	高速リトラクト		+		PGsl
LIMS	K	G96/G961 および G97 の速度制限	M	+		PGsl
LLI	K	変数の下限値		+		PGAsl
LN	F	自然対数		+	+	PGAsl
LOCK	P	ID によりシンクロナイズドアクションを解除 (テクノロジサイクルを停止)		-	+	FBSYsl
LONGHOLE	C (T)	長穴		+		PGAsl
LOOP	K	無限ループの開始		+		PGAsl

命令 M ... R

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
M0		プログラムストップ		+	+	PGsl
M1		オプショナルストップ		+	+	PGsl
M2		プログラムの終了、メインプログラム (M30 と同じ)		+	+	PGsl
M3		CW 主軸回転		+	+	PGsl
M4		CCW 主軸回転		+	+	PGsl
M5		主軸停止		+	+	PGsl
M6		工具交換		+	+	PGsl
M17		サブプログラム終了		+	+	PGsl
M19		主軸を SD43240 で入力された位置に位置決め		+	+	PGsl
M30		プログラムの終了、メインプログラム (M2 と同じ)		+	+	PGsl
M40		自動ギヤ切り替え		+	+	PGsl
M41 ... M45		ギヤ選択 1 ... 5		+	+	PGsl
M70		軸モードへの移行		+	+	PGsl
MASLDEF	P	マスタ/スレーブ軸グループの定義		+	+	PGAsl
MASLDEL	P	マスタ/スレーブ軸グループを連結解除し、グループ定義を解除		+	+	PGAsl
MASLOF	P	一時的に連結を解除		+	+	PGAsl
MASLOFS	P	自動スレーブ軸停止により、一時的に連結を解除		+	+	PGAsl
MASLON	P	一時的に連結を起動		+	+	PGAsl
MATCH	F	文字列中の文字列の検索		+	-	PGAsl
MAXVAL	F	2 個の変数の、大きい方の値(算術機能)		+	+	PGAsl
MCALL	K	モーダルサブプログラム呼び出し		+		PGAsl
MEAC	K	残移動距離削除なしの連続軸計測	s	+	+	PGAsl
MEAFRAME	F	計測点からのフレーム計算		+	-	PGAsl
MEAS	O	残移動距離を削除する計測	s	+		PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
MEASA	K	残移動距離を削除する軸計測	s	+	+	PGAsl
MEASURE	F	ワーク計測と工具計測の計算方式		+	-	FB1sl (M5)
MEAW	O	残移動距離を削除しない計測	s	+		PGAsl
MEAWA	K	残移動距離を削除しない軸計測	s	+	+	PGAsl
Mi	K	フレームデータへのアクセス:ミラーリング		+		PGAsl
MINDEX	F	入力文字列の文字のインデックスを定義		+	-	PGAsl
MINVAL	F	2 個の変数の、小さい方の値(算術機能)		+	+	PGAsl
MIRROR	G	プログラマブルミラーリング	s	+		PGAsl
MMC	P	パートプログラムから、対話用ウィンドウを対話形式で HMI 上に呼び出します		+	-	PGAsl
MOD	K	モジュロ除算		+		PGAsl
MODAXVAL	F	モジュロ回転軸のモジュロ位置を特定		+	-	PGAsl
MOV	K	軸位置決めの起動		-	+	FBSYsl
MOVT	O	工具方向への移動動作の終点の指定				FB1(K2)
MSG	P	プログラマブルメッセージ	M	+	-	PGsl
MVTOOL	P	工具を移動するための言語命令		+	-	FBWsl
N	O	NC 補助ブロック番号		+		PGsl
NAMETOINT	F	システム変数インデックスの特定		+		PGAsl
NC	K	データの有効範囲を指定します		+		PGAsl
NEWCONF	P	変更したマシンデータを適用(「マシンデータの有効化」に相当します)		+	-	PGAsl
NEWMT	F	新しいマルチツールの作成		+	-	FBWsl
NEWT	F	新しい工具の作成		+	-	FBWsl
NORM[6]	G	工具補正の始点と終点の標準設定	M	+		PGsl
NOT	K	論理否定		+		PGAsl
NPROT	P	機械別プロテクションゾーンのオン/オフ		+	-	PGAsl
NPROTDEF	P	機械別プロテクションゾーンの定義		+	-	PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
NUMBER	F	入力文字列を数字に変換		+	-	PGAsl
OEMIPO1	G	OEM 補間 1	M	+		PGAsl
OEMIPO2	G	OEM 補間 2	M	+		PGAsl
OF	K	CASE 分岐のキーワード		+		PGAsl
OFFN	O	プログラム指令輪郭の許容量	M	+		PGsl
OMA1	O	OEM アドレス 1	M	+		PGAsl
OMA2	O	OEM アドレス 2	M	+		PGAsl
OMA3	O	OEM アドレス 3	M	+		PGAsl
OMA4	O	OEM アドレス 4	M	+		PGAsl
OMA5	O	OEM アドレス 5	M	+		PGAsl
または	K	論理演算子、論理和		+		PGAsl
ORIAXES	G	機械軸または旋回軸の直線補間	M	+		PGAsl
ORIAXPOS	G	仮想旋回軸の回転軸位置による旋回角度	M	+		PGAsl
ORIC[6]	G	外側コーナーでの向きの変更を、挿入する円弧ブロックに重畳します	M	+		PGAsl
ORICONCCW	G	円錐面での CCW 方向の補間	M	+		PGAsl/FB3sl (F3)
ORICONCW	G	円錐面での CW 方向の補間	M	+		PGAsl/FB3sl (F4)
ORICONIO	G	中間旋回設定による円錐面の補間	M	+		PGAsl/FB3sl (F4)
ORICONTO	G	接線方向の遷移による円錐面の補間 (最終旋回)	M	+		PGAsl/FB3sl (F5)
ORICURVE	G	工具の 2 つの接点の移動を指定した旋回補間	M	+		PGAsl/FB3sl (F6)
ORID	G	円弧ブロックの前に向きの変更を実行します	M	+		PGAsl
ORIEULER[6]	G	オイラー角による旋回角度	M	+		PGAsl
ORIMKS	G	機械座標系の工具オリエンテーション	M	+		PGAsl
ORIPATH	G	軌跡に対する工具オリエンテーション	M	+		PGAsl
ORIPATHS	G	軌跡に対する工具オリエンテーション、旋回処理の不連続変化をスムージングします	M	+		PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
ORIPLANE	G	平面での補間 (ORIVECT に相当)、 大半径円弧補間	M	+		PGAsl
ORIRESET	P	3 つまでの旋回軸による工具オリエンテーションの初期設定		+	-	PGAsl
ORIROTA [6]	G	アブソリュート回転方向への回転角度	M	+		PGAsl
ORIROTC	G	軌跡タンジェントに対する接線方向の回転ベクトル	M	+		PGAsl
ORIROTR	G	旋回の開始と終了間の平面に対する回転角度	M	+		PGAsl
ORIROTT	G	配向ベクトルの変更に対する回転角度	M	+		PGAsl
ORIRPY	G	RPY 角による旋回角度(XYZ)	M	+		PGAsl
ORIRPY2	G	RPY 角による旋回角度(ZYX)	M	+		PGAsl
ORIS	O	向きの変更	M	+		PGAsl
ORISOF [6]	G	旋回処理のスムージングのオフ	M	+		PGAsl
ORISOLH	F	方向の計算		+		PGAsl
ORISON	G	旋回処理のスムージングのオン	M	+		PGAsl
ORIVECT [6]	G	大円弧補間(ORIPLANE と同じ)	M	+		PGAsl
ORIVIRT1	G	仮想旋回軸による旋回角度(定義 1)	M	+		PGAsl
ORIVIRT2	G	仮想旋回軸による旋回角度(定義 1)	M	+		PGAsl
ORIWKS [6]	G	ワーク座標系の工具オリエンテーション	M	+		PGAsl
OS	K	揺動のオン/オフ		+		PGAsl
OSB	K	揺動:起点	M	+		FB1sl (P5)
OSC	G	連続工具オリエンテーションのスムージング	M	+		PGAsl
OSCILL	K	軸:1 ～ 3 つの切り込み軸	M	+		PGAsl
OSCTRL	K	揺動オプション	M	+		PGAsl
OSD	G	セッティングデータでスムージング距離が指定された工具オリエンテーションのスムージング	M	+		PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
OSE	K	揺動の終了位置	M	+		PGAsl
OSNSC	K	揺動:スパークアウトサイクル数	M	+		PGAsl
OSOF [6]	G	工具オリエンテーションのスムージングのオフ	M	+		PGAsl
OSP1	K	揺動:左の反転点	M	+		PGAsl
OSP2	K	揺動:右の反転点	M	+		PGAsl
OSS	G	ブロック終点の工具オリエンテーションのスムージング	M	+		PGAsl
OSSE	G	ブロックの始点と終点の工具オリエンテーションのスムージング	M	+		PGAsl
OST	G	セッティングデータの角度許容範囲(°単位)の指定による工具オリエンテーションのスムージング(プログラム指令の旋回処理からの最大誤差)	M	+		PGAsl
OST1	K	揺動:左の反転点の停止点	M	+		PGAsl
OST2	K	揺動:右の反転点の停止点	M	+		PGAsl
OTOL	A	コンプレッサ機能、旋回スムージング、およびスムージングタイプの旋回許容範囲	M	+		PGAsl
OVR	K	速度オーバライド	M	+		PGAsl
OVRA	K	軸速度オーバーライド	M	+	+	PGAsl
OVRRAP	K	早送りオーバーライド	M	+		PGAsl
P	O	サブプログラム繰り返しの回数		+		PGAsl
PAROT	G	ワークにワーク座標系を配置	M	+		PGsl
PAROTOF [6]	G	ワークに対するフレーム回転を解除	M	+		PGsl
PCALL	K	絶対パスとパラメータ転送によるサブプログラム呼び出し		+		PGAsl
PDELAYOF	G	遅延のあるパンチングのオフ	M	+		PGAsl
PDELAYON [6]	G	遅延のあるパンチングのオン	M	+		PGAsl
PHI	K	円錐の方向軸を中心とした向きの回転角度		+		PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
PHU	K	変数の物理単位		+		PGAsl
PL	O	1.B スプライン:ノードクリアランス 2.多項式補間　多項式補間のパラメータ区間の長さ	s	+		PGAsl
PM	K	毎分		+		PGsl
PO	K	多項式補間の多項式係数	s	+		PGAsl
POCKET3	C (T)	長方形ポケットのフライス加工		+		PGAsl
POCKET4	C (T)	円形ポケットのフライス削り		+		PGAsl
POLF	K	高速リトラクトの後退位置	M	+		PGsl/PGAsl
POLFA	P	$AA_ESR_TRIGGER による単独軸の後退開始位置	M	+	+	PGsl
POLFMASK	P	軸間の関係なしに個別に後退用の軸を有効化	M	+	-	PGsl
POLFMLIN	P	軸間の直線補間で後退用の軸を有効化	M	+	-	PGsl
POLY	G	多項式補間	M	+		PGAsl
POLYPATH	P	AXIS と VECT の両方の軸グループに対して多項式補間を選択可能	M	+	-	PGAsl
PON	G	パンチングのオン	M	+		PGAsl
PONS	G	補間サイクルのパンチングのオン	M	+		PGAsl
POS	K	軸の位置決め		+	+	PGsl
POSA	K	ブロック境界を越える軸の位置決め		+	+	PGsl
POSM	P	マガジンの位置決め		+	-	FBWsl
POSMT	P	工具ホルダのロケーション番号へのマルチツールの位置決め		+	-	FBWsl
POSP	K	複数区間の軸位置決め(揺動)		+		PGsl
POSRANGE	F	現在補間している軸の位置指令が、事前定義された基準位置の範囲にあるかどうかを特定		+	+	FBSYsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
POT	F	二乗 (算術機能)		+	+	PGAsl
PR	K	毎回転		+		PGsl
PREPRO	PA	解析でのサブプログラムの識別		+		PGAsl
PRESETON	P	原点確立状態をリセットして現在値を設定		+	+	PGAsl
PRESETONS	P	原点確立状態をリセットして現在値を設定		+	+	PGAsl
PRIO	K	割り込み処理の優先度を設定するためのキーワード		+		PGAsl
PRLOC	K	ローカル変更後にのみ、リセット時の変数の初期化		+		PGAsl
PROC	K	プログラムの最初の命令		+		PGAsl
PROTA	P	干渉モデルの再計算の要求		+		PGAsl
PROTD	F	2 つのプロテクションゾーンの間の距離の計算		+		PGAsl
PROTS	P	プロテクションゾーンステータスの設定		+		PGAsl
psi	K	円錐の開口角度		+		PGAsl
PTP	G	ポイントツーポイント移動(PTP 移動)	M	+		PGAsl
PTPG0	G	G0 の場合はポイントツーポイント移動のみ、G0 以外の場合は軌跡移動の CP	M	+		PGAsl
PTPWOC	G	オリエンテーションの変更による補正移動なしのポイントツーポイント移動	M	+		PGAsl
PUNCHACC	P	ニブリングの、移動に応じた加減速度		+	-	PGAsl
PUTFTOC	P	並列目立ての工具仕上げオフセット		+	-	PGAsl
PUTFTOCF	P	FCTDEF で定義した並列目立て用機能に対応した工具仕上げオフセット		+	-	PGAsl
PW	O	B スプライン、点の重み	s	+		PGAsl
QU	K	高速追加 (補助)機能の出力		+		PGsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
R...	O	設定可能アドレス識別子として、および数値拡張子付き算術変数		+		PGAsl
RAC	K	ノンモーダルのアブソリュート指令の軸別半径指定	s	+		PGsl
RDISABLE	P	読み込み停止		-	+	FBSYsl
READ	P	指定したファイルの複数の行を読み出し、読み出した情報を配列に格納		+	-	PGAsl
REAL	K	データタイプ:符号付きフローティングポイント変数(実数)		+		PGAsl
REDEF	K	システム変数の再定義、ユーザー変数、および NC 言語命令		+		PGAsl
RELEASE	P	軸入れ替えのために機械軸を解放		+	+	PGAsl
REP	K	配列のすべての要素を同じ値で初期化するためのキーワード		+		PGAsl
REPEAT	K	プログラムループの繰り返し		+		PGAsl
REPEATB	K	プログラム行の繰り返し		+		PGAsl
REPOSA	G	すべての軸の直線再位置決め	s	+		PGAsl
REPOSH	G	半円による再位置決め	s	+		PGAsl
REPOSHA	G	全ジオメトリ軸の半円による再位置決め	s	+		PGAsl
REPOSL	G	直線再位置決め	s	+		PGAsl
REPOSQ	G	4 分円の再位置決め	s	+		PGAsl
REPOSQA	G	全ジオメトリ軸の 4 分円による再位置決め	s	+		PGAsl
RESETMON	P	指令値適用の言語命令		+	-	FBWsl
RET	P	サブプログラム終了		+	+	PGAsl
RETB	P	サブプログラム終了		+	+	PGAsl
RIC	K	ノンモーダルのインクレメンタル指令の軸別半径指定	s	+		PGsl
RINDEX	F	入力文字列の文字のインデックスを定義		+	-	PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、一覧 (ページ 1059)を参照してください。						
RMB	G	ブロックの始点への再位置決め	M	+		PGAsl
RMBBL	G	ブロックの始点への再位置決め	s	+		PGAsl
RME	G	ブロック終点への再位置決め	M	+		PGAsl
RMEBL	G	ブロック終点への再位置決め	s	+		PGAsl
RMI [6]	G	中断点への再位置決め	M	+		PGAsl
RMIBL [6]	G	中断点への再位置決め	s	+		PGAsl
RMN	G	最も近い軌跡点への再位置決め	M	+		PGAsl
RMNBL	G	最も近い軌跡点への再位置決め	s	+		PGAsl
RND	O	輪郭のコーナーの丸み付け	s	+		PGsl
RNDM	O	モーダル丸み付け	M	+		PGsl
ROT	G	プログラマブル座標回転	s	+		PGsl
ROTS	G	立体角によるプログラマブルフレームの回転	s	+		PGsl
ROUND	F	小数点以下の四捨五入		+	+	PGAsl
ROUNDUP	F	入力値の切り上げ		+	+	PGAsl
RP	O	極半径	m/s	+		PGsl
RPL	O	平面の回転	s	+		PGsl
RT	K	フレームデータにアクセスするためのパラメータ:回転		+		PGAsl
RTLIOF	G	直線補間のない G0(単独軸補間)	M	+		PGsl
RTLION [6]	G	直線補間による G0	M	+		PGsl

命令 S ... Z

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1][2][3][4][5] 詳細は、 一覧 (ページ 1059) を参照してください。						
S	A	主軸速度 (G4 の場合、G96/G961 の意味が異なります)	m/s	+	+	PGsl
SAVE	PA	サブプログラムの呼び出し時の情報保存の属性		+		PGAsl
SBLOF	P	シングルブロックのマスク		+	-	PGAsl
SBLON	P	シングルブロックのマスクを無効化します		+	-	PGAsl
SC	K	フレームデータにアクセスするためのパラメータ:スケーリング		+		PGAsl
SCALE	G	プログラマブルスケーリング	S	+		PGsl
SCC	K	G96/G961/G962 に対して径方向軸を選択して割り当て。軸識別子は、ジオメトリ軸、チャネル軸、または機械軸タイプの場合があります		+		PGsl
SCPARA	K	サーボパラメータセットをプログラム指令		+	+	PGAsl
SD	A	スプラインの次数	S	+		PGAsl
SET	K	配列のすべての要素を、リスト値で初期化するためのキーワード		+		PGAsl
SETAL	P	アラームの設定		+	+	PGAsl
SETDNO	F	工具(T)の刃先(CE)の D 番号の割り当て		+	-	PGAsl
SETINT	K	NC 入力がある場合に、どの割り込みルーチンを起動するかを定義		+		PGAsl
SETM	P	専用チャネルのマークの設定		+	+	PGAsl
SETMS	P	マシンデータで定義したメイン主軸に設定		+	-	PGsl
SETMS(n)	P	主軸 n をメイン主軸として設定		+		PGsl
SETMTH	P	メイン工具ホルダ番号を設定		+	-	FBWsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、 一覧 (ページ 1059) を参照してください。						
SETPIECE	P	主軸に割り当てられたすべての工具に部品番号を設定		+	-	FBWsl
SETTA	P	摩耗グループから工具を起動		+	-	FBWsl
SETTCOR	F	すべての補足条件を考慮して、工具成分を変更		+	-	PGAsl
SETTIA	P	摩耗グループから工具を解除		+	-	FBWsl
SF	A	ねじ切りの始点オフセット	M	+		PGsl
sin	F	正弦(三角関数)		+	+	PGAsl
SIRELAY	F	SIRELIN、SIRELOUT、および SIRELTIME でパラメータ設定した安全機能の起動		-	+	FBSIsl
SIRELIN	P	ファンクションブロックの入力変数を初期化		+	-	FBSIsl
SIRELOUT	P	ファンクションブロックの出力変数を初期化		+	-	FBSIsl
SIRELTIME	P	ファンクションブロックのタイマーを初期化		+	-	FBSIsl
SLOT1	C (T)	直線溝		+		PGAsl
SLOT2	C (T)	円周溝		+		PGAsl
SOFT	G	加々速度一定加減速の軌跡加減速度	M	+		PGAsl
SOFTA	P	プログラム指令軸で、加々速度制限付きの軸加減速を起動		+	-	PGAsl
SON	G	ニブリングのオン	M	+		PGAsl
SONS	G	補間サイクルのニブリングのオン	M	+		PGAsl
SPATH[6]	G	FGROUP 軸の軌跡基準は円弧長です	M	+		PGAsl
SPCOF	P	メイン主軸または主軸を位置制御から速度制御に切り替え	M	+	-	PGsl
SPCON	P	メイン主軸または主軸を速度制御から位置制御に切り替え	M	+	-	PGAsl
SPI	F	主軸番号を軸識別子に変換		+	-	PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、 一覧 (ページ 1059) を参照してください。						
SPIF1 [6]	G	パンチング/ニブリングのバイト 1 に対する 高速 NC 入力/出力	M	+		FB2sl (N4)
SPIF2	G	パンチング/ニブリングのバイト 2 に対する 高速 NC 入力/出力	M	+		FB2sl (N4)
SPLINEPATH	P	スプライングループの定義		+	-	PGAsl
SPN	A	ブロックごとの軌跡区間数	S	+		PGAsl
SPOF [6]	G	ストロークのオフ、 ニブリング、パンチングのオフ	M	+		PGAsl
SPOS	K	主軸位置決め	M	+	+	PGsl
SPOSA	K	ブロック境界を越える主軸位置決め	M	+		PGsl
SPP	A	軌跡区間の長さ	M	+		PGAsl
SPRINT	F	フォーマットされた入力文字列を返します。		+		PGAsl
SQRT	F	平方根 (算術機能)		+	+	PGAsl
SR	A	シンクロナイズドアクションの揺動後退軌跡	S	+		PGsl
SRA	K	シンクロナイズドアクション用の軸の外部入力による揺動後退軌跡	M	+		PGsl
ST	A	シンクロナイズドアクションの揺動スパークアウト時間	S	+		PGsl
STA	K	シンクロナイズドアクション用の軸の揺動スパークアウト時間	M	+		PGsl
START	P	複数のチャネルで、実行中のプログラムから選択プログラムを同時に起動		+	-	PGAsl
STARTFIFO [6]	G	実行; 全解析メモリを同時にすべて使用	M	+		PGAsl
STAT		関節継手の位置	S	+		PGAsl
STOLF	K	G0 許容範囲係数	M	+		PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、 一覧 (ページ 1059) を参照してください。						
STOPFIFO	G	加工の停止; STARTFIFO を検出するか、解析メモリがいっぱいになるか、プログラムが終了するまで、解析メモリを使用します。	M	+		PGAsl
STOPRE	P	解析されたすべてのブロックがメインランで実行されるまで先読み停止		+	-	PGAsl
STOPREOF	P	先読み停止の無効化		-	+	FBSYsl
STRING	K	データタイプ:文字列		+		PGAsl
STRINGIS	F	NC 言語と NC サイクル名称の現在の適用範囲、ユーザー変数、マクロ、およびこの命令に特有のラベル名称が存在しているか、有効か、定義済みか、または動作中かをチェックします。		+	-	PGAsl
STRLEN	F	文字列長の定義		+	-	PGAsl
SUBSTR	F	入力文字列の文字のインデックスを定義		+	-	PGAsl
SUPA	G	現在のゼロオフセット(プログラム指令オフセット、システムフレーム、ハンドルオフセット(DRF)、外部ゼロオフセット、および重畳移動を含む)のマスク	S	+		PGsl
SVC	K	工具切削速度	M	+		PGsl
SYNFCT	P	シンクロナイズドアクションの条件に応じて多項式を評価		-	+	FBSYsl
SYNR	K	実行タイミングに同期して変数を読み出します		+		PGAsl
SYNRW	K	実行タイミング同期して変数の読み出し、および書き込みをおこないます		+		PGAsl
SYNW	K	実行タイミングに同期して変数を書き込みます		+		PGAsl
t	A	工具の呼び出し (マシンデータで指定した場合にのみ交換; 指定しない場合は、M6 命令が必要)		+		PGsl
tan	F	タンジェント(三角関数)		+	+	PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、 一覧 (ページ 1059) を参照してください。						
TANG	P	法線方向制御:連結の定義		+	-	PGAsl
TANGDEL	P	法線方向制御:連結の解除		+	-	PGAsl
TANGOF	P	法線方向制御:連結の解除		+	-	PGAsl
TANGON	P	法線方向制御:連結の有効化		+	-	PGAsl
TCA (828D: _TCA)	P	工具状態を問わない工具選択/工具交換		+	-	FBWsl
TCARR	A	工具ホルダ(番号「m」)の要求		+		PGAsl
TCI	P	工具を工具バッファからマガジンへロード		+	-	FBWsl
TCOABS[6]	G	現在の工具オリエンテーションから工具長成分を特定	M	+		PGAsl
TCOFR	G	動作中のフレームの向きから工具長成分を特定	M	+		PGAsl
TCOFRX	G	工具の選択時に、動作中のフレームの工具オリエンテーションを特定、X 方向の工具点	M	+		PGAsl
TCOFRY	G	工具の選択時に、動作中のフレームの工具オリエンテーションを特定、Y 方向の工具点	M	+		PGAsl
TCOFRZ	G	工具の選択時に、動作中のフレームの工具オリエンテーションを特定、Z 方向の工具点	M	+		PGAsl
THETA	A	回転角度	S	+		PGAsl
TILT	A	傾斜角	M	+		PGAsl
TLIFT	P	法線方向制御:挿入ブロック生成の有効化		+	-	PGAsl
TML	P	マガジンロケーション番号による工具の選択		+	-	FBWsl
TMOF	P	工具監視の解除		+	-	PGAsl
TMON	P	工具監視の起動		+	-	PGAsl
TO	K	FOR カウンタループの終了値を指定		+		PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、 一覧 (ページ 1059) を参照してください。						
TOFF	K	インデックスで指定されたジオメトリ軸に平行な、有効工具長成分の方向への工具長補正	M	+		PGsl
TOFFL	K	工具長成分 L1、L2、または L3 の方向への工具長補正	M	+		PGsl
TOFFOF	P	オンライン工具長補正の解除		+	-	PGAsl
TOFFON	P	オンライン工具長補正の起動		+	-	PGAsl
TOFFR	A	工具径補正	M	+		PGsl
TOFRAME	G	フレームを回転して、WCS の Z 軸を工具オリエンテーションに平行に配置します	M	+		PGsl
TOFRAMEX	G	フレームを回転して、WCS の X 軸を工具オリエンテーションに平行に配置します	M	+		PGsl
TOFRAMEY	G	フレームを回転して、WCS の Y 軸を工具オリエンテーションに平行に配置します	M	+		PGsl
TOFRAMEZ	G	TOFRAME と同じです	M	+		PGsl
TOLOWER	F	文字列中の文字を小文字に変換		+	-	PGAsl
TOOLENV	F	メモリに格納された工具データの評価内容について現在の状態を保存		+	-	PGAsl
TOOLGNT	F	工具グループの工具数の決定		+	-	FBWsl
TOOLGT	F	工具グループの工具の T 番号を決定		+	-	FBWsl
TOROT	G	フレームを回転して、WCS の Z 軸を工具オリエンテーションに平行に配置します	M	+		PGsl
TOROTOF[6]	G	工具方向のフレーム回転のオフ	M	+		PGsl
TOROTX	G	フレームを回転して、WCS の X 軸を工具オリエンテーションに平行に配置します	M	+		PGsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、 一覧 (ページ 1059) を参照してください。						
TOROTY	G	フレームを回転して、WCS の Y 軸を工具オリエンテーションに平行に配置します	M	+		PGsl
TOROTZ	G	TOROT と同じです	M	+		PGsl
TOUPPER	F	文字列中の文字を大文字に変換		+	-	PGAsl
TOWBCS	G	基本座標系(BCS)の摩耗値	M	+		PGAsl
TOWKCS	G	キネマティックトランスフォーメーションをおこなう工具ヘッドの座標系の摩耗値(工具回転による機械座標系とは異なります)	M	+		PGAsl
TOWMCS	G	機械座標系の摩耗値	M	+		PGAsl
TOWSTD[6]	G	工具長のオフセットの初期設定値	M	+		PGAsl
TOWTCS	G	工具座標系の摩耗値(工具ホルダの工具ホルダ基準点)	M	+		PGAsl
TOWWCS	G	ワーク座標系の摩耗値	M	+		PGAsl
tr	K	フレーム変数のオフセット成分		+		PGAsl
TRAANG	P	傾斜軸座標変換		+	-	PGAsl
TRACON	P	座標変換重畳		+	-	PGAsl
TRACYL	P	円筒:円筒補間		+	-	PGAsl
TRAFOOF	P	チャネルで動作中の座標変換を解除		+	-	PGAsl
TRAFOON	P	キネマティック結合により定義された変換の起動		+	-	PGAsl
TRAILOF	P	非同期連結移動のオフ		+	+	PGAsl
TRAILON	P	非同期連結移動のオン		+	+	PGAsl
TRANS	G	アブソリュートプログラマブルゼロオフセット	S	+		PGsl
TRANSMIT	P	極座標変換(正面加工)		+	-	PGAsl
TRAORI	P	4 軸座標変換、5 軸座標変換、汎用座標変換		+	-	PGAsl
TRUE	K	論理定数:正		+		PGAsl
TRUNC	F	小数点以下の切り捨て		+	+	PGAsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、 一覧 (ページ 1059) を参照してください。						
Tu		軸角度	S	+		PGAsl
TURN	A	ヘリカルのターン数	S	+		PGsl
ULI	K	変数の上限値		+		PGAsl
UNLOCK	P	ID によるシンクロナイズドアクションが有効(テクノロジサイクルを続行)		-	+	FBSYsl
UNTIL	K	REPEAT ループ終了の条件		+		PGAsl
UPATH	G	FGROUP 軸の軌跡基準は曲線パラメータです	M	+		PGAsl
var	K	キーワード:パラメータ転送のタイプ		+		PGAsl
VELOLIM	K	最大軸速度の低減	M	+		PGAsl
VELOLIMA	K	スレーブ軸の最大軸速度の低減または増加	M	+	+	PGAsl
WAITC	P	軸/主軸に対して関連ブロック切り替えの条件が満たされるまで待機		+	-	PGAsl
WAITE	P	別のチャネルのプログラム終了を待機		+	-	PGAsl
WAITENC	P	軸位置の原点同期または復元まで待機		+	-	PGAsl
WAITM	P	指定したチャネルのマークを待機; 先行ブロックをイグザクトストップで終了		+	-	PGAsl
WAITMC	P	指定したチャネルのマークを待機; 他のチャネルがマークに到達していない場合にのみイグザクトストップ		+	-	PGAsl
WAITP	P	位置決め軸の移動終了まで待機		+	-	PGsl
WAITS	P	主軸位置への到達を待機		+	-	PGsl
WALCS0 [6]	G	ワーク座標系のワーキングエリアリミットの選択解除	M	+	-	PGsl
WALCS1	G	WCS 作業領域リミットグループ 1 が有効	M	+	-	PGsl
WALCS2	G	WCS 作業領域リミットグループ 2 が有効	M	+	-	PGsl
WALCS3	G	WCS 作業領域リミットグループ 3 が有効	M	+	-	PGsl

命令	タイプ[1]	意味	W[2]	TP[3]	SA[4]	説明の参照先[5]
[1] [2] [3] [4] [5] 詳細は、 一覧 (ページ 1059) を参照してください。						
WALCS4	G	WCS 作業領域リミットグループ 4 が有効	M	+	-	PGsl
WALCS5	G	WCS 作業領域リミットグループ 5 が有効	M	+	-	PGsl
WALCS6	G	WCS 作業領域リミットグループ 6 が有効	M	+	-	PGsl
WALCS7	G	WCS 作業領域リミットグループ 7 が有効	M	+	-	PGsl
WALCS8	G	WCS 作業領域リミットグループ 8 が有効	M	+	-	PGsl
WALCS9	G	WCS 作業領域リミットグループ 9 が有効	M	+	-	PGsl
WALCS10	G	WCS 作業領域リミットグループ 10 が有効	M	+	-	PGsl
WALIMOF	G	BCS ワーキングエリアリミットのオフ	M	+	-	PGsl
WALIMON [6]	G	BCS ワーキングエリアリミットのオン	M	+	-	PGsl
WHEN	K	条件を満たすたびに、動作を 1 回実行します。		-	+	FBSYsl
WHENEVER	K	条件を満たしている間は、各補間サイクルで動作を周期的に実行します。		-	+	FBSYsl
WHILE	K	WHILE プログラムループの開始		+		PGAsl
WRITE	P	テキストをファイルシステムに書込みます。 指定されたファイルの最後にブロックを追加します。		+	-	PGAsl
WRTPR	P	OPI 変数での文字列の書き込み (WRTPR)		+	-	PGsl
X	A	軸名称	m/s	+	+	PGsl
XOR	O	排他的論理和		+		PGAsl
Y	A	軸名称	m/s	+	+	PGsl
Z	A	軸名称	m/s	+	+	PGsl

凡例

1) 命令のタイプ:

A アドレス

値の割り当て先の識別子(たとえば、OVR=10).値の割り当てはおこなわないで、機能をオン/オフにするアドレスもあります(例えば、CPLON と CPLOF)。

C (A) AST サイクル

AST (= Automatic Servo Tuning)による自動ポスト最適化(チューニング)の予約済み NC プログラム。パラメータは、特定の最適化の状況に対応するために使用されます。これらのパラメータは呼び出し時に転送されます。

C 計測サイクル

(M) 円筒形のワークの内部の直径を決定するような、一般的に有効な専用の測定動作がプログラムされる予約 NC プログラム。パラメータは特定の測定状況に対応するために使用されます。これらのパラメータは呼び出し時に転送されます。

C (T) 加工種別のサイクル

ねじのタッピングまたはポケットのフライス削りなどの、一般的に有効な専用の加工運転がプログラム指令された予約 NC プログラム。特定の加工状況に対する調整は、サイクル呼び出し時にサイクルに転送されるパラメータによって実行されます。

F 予約機能(戻り値を提供します)

予約機能の呼び出しは、演算式のオペランドにすることができます。

G G 命令

G 命令は G グループに分けられます。グループのうち、1 つの G 命令のみを 1 ブロックにプログラム指令できます。G 命令は、モーダル(同じグループの別の命令によって取り消されるまで)、または、プログラム指令ブロックに対してのみ有効(ノンモーダル)です。

K キーワード

ブロックの構文を定義する識別子。キーワードに値は割り当てられず、キーワードで NC 機能をオン/オフすることはできません。

例:フロー制御(IF, ELSE, ENDIF, WHEN, ...)、プログラムの実行(GOTOB, GOTO, RET ...)

O 演算子

算術演算、比較演算、または論理演算の演算子。

P 予約処理(戻り値を提供しません)

PA　プログラム属性

プログラム属性は、サブプログラムの定義行の終わりに指定します。

```
PROC <プログラム名称>(...) <プログラム属性>
```

プログラム属性は、サブプログラムの実行時の動作を決定します。

2) 命令の効果:

M　モーダル

S　ノンモーダル

3) パートプログラムのプログラマブル属性

+　プログラム指令可能

-　プログラム指令不可

M　工作機械メーカによってのみプログラム指令可能

4) シンクロナイズドアクションのプログラム指令属性

+　プログラム指令可能

-　プログラム指令不可

t　テクノロジサイクルでのみプログラム指令可能

5) 命令の詳細説明があるマニュアルの参照先:

PGsl　プログラミングマニュアル基本編

PGAsl　プログラミングマニュアル上級編

BNMsl　プログラミングマニュアル、計測サイクル

BHDsl　操作マニュアル、旋削

BHFsl　操作マニュアル、フライス加工

FB1sl ( )　機能マニュアル 基本機能(括弧内は、対応する機能説明の略語の英数字です)

FB2sl ( )　機能マニュアル 上級機能(括弧内は、対応する機能説明の略語の英数字です)

FB3sl ( )　機能マニュアル 応用機能(括弧内は、対応する機能説明の略語の英数字です)

FBSIsl　機能マニュアル Safety Integrated

FBSYsl　機能マニュアル、シンクロナイズドアクション

FBWsl　機能マニュアル 工具管理機能

6) プログラム開始点での初期設定(他で何もプログラム指令していなければ、コントローラの出荷時設定値です)。

図 21-1　操作表の脚注の意味

21.2 命令:SINUMERIK 828D での適用

注記

サイクル

サイクルは、ライセンスを必要とする以下のオプションに依存する場合、「オプション」としてマーキングされます。

- 拡張テクノロジ機能(注文番号:6FC5800-0AP58-0YB0)
- 計測サイクル(注文番号:6FC5800-0AP28-0YB0)
- 計測キネマティックス(注文番号:6FC5800-0AP18-0YB0)
- SINUMERIK Grinding Advanced(注文番号:6FC5800-0AS35-0YB0)

オプション「拡張テクノロジ機能」の結果として一部の機能のみが含まれている場合は、マーキングなし

21.2.1 制御タイプ フライス削り / 旋削

命令 A ... C

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
:	●	●	●	●	●	●	●	●
*	●	●	●	●	●	●	●	●
+	●	●	●	●	●	●	●	●
-	●	●	●	●	●	●	●	●
<	●	●	●	●	●	●	●	●
<<	●	●	●	●	●	●	●	●
<=	●	●	●	●	●	●	●	●
=	●	●	●	●	●	●	●	●
>=	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
/	●	●	●	●	●	●	●	●
/0 … /7	●	●	●	●	●	●	●	●
A	●	●	●	●	●	●	●	●
A2	-	-	-	-	-	-	-	-
A3	-	-	-	-	-	-	-	-
A4	-	-	-	-	-	-	-	-
A5	-	-	-	-	-	-	-	-
A6	-	-	-	-	-	-	-	-
A7	-	-	-	-	-	-	-	-
ABS	●	●	●	●	●	●	●	●
AC	●	●	●	●	●	●	●	●
ACC	●	●	●	●	●	●	●	●
ACCLIMA	●	●	●	●	●	●	●	●
ACN	●	●	●	●	●	●	●	●
ACOS	●	●	●	●	●	●	●	●
ACP	●	●	●	●	●	●	●	●
ACTBLOCNO	●	●	●	●	●	●	●	●
ADDFRAME	●	●	●	●	●	●	●	●
ADIS	●	●	●	●	●	●	●	●
ADISPOS	●	●	●	●	●	●	●	●
ADISPOSA	●	●	●	●	●	●	●	●
ALF	●	●	●	●	●	●	●	●
AMIRROR	●	●	●	●	●	●	●	●
および	●	●	●	●	●	●	●	●
ANG	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
AP	●	●	●	●	●	●	●	●
APR	●	●	●	●	●	●	●	●
APRB	●	●	●	●	●	●	●	●
APRP	●	●	●	●	●	●	●	●
APW	●	●	●	●	●	●	●	●
APWB	●	●	●	●	●	●	●	●
APWP	●	●	●	●	●	●	●	●
APX	●	●	●	●	●	●	●	●
AR	●	●	●	●	●	●	●	●
AROT	●	●	●	●	●	●	●	●
AROTS	●	●	●	●	●	●	●	●
AS	●	●	●	●	●	●	●	●
ASCALE	●	●	●	●	●	●	●	●
ASIN	●	●	●	●	●	●	●	●
ASPLINE	○	○	○	○	○	○	○	○
ATAN2	●	●	●	●	●	●	●	●
ATOL	●	●	●	●	●	●	●	●
ATRANS	●	●	●	●	●	●	●	●
AUXFUDEL	●	●	●	●	●	●	●	●
AUXFUDELG	●	●	●	●	●	●	●	●
AUXFUMSEQ	●	●	●	●	●	●	●	●
AUXFUSYNC	●	●	●	●	●	●	●	●
AX	●	●	●	●	●	●	●	●
AXCTSWE	-	-	-	-	-	-	-	-
AXCTSWEC	-	-	-	-	-	-	-	-

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
AXCTSWED	-	-	-	-	-	-	-	-
AXIS	●	●	●	●	●	●	●	●
AXNAME	●	●	●	●	●	●	●	●
AXSTRING	●	●	●	●	●	●	●	●
AXTOCHAN	●	●	●	●	●	●	●	●
AXTOSPI	●	●	●	●	●	●	●	●
B	●	●	●	●	●	●	●	●
B2	-	-	-	-	-	-	-	-
B3	-	-	-	-	-	-	-	-
B4	-	-	-	-	-	-	-	-
B5	-	-	-	-	-	-	-	-
B6	-	-	-	-	-	-	-	-
B7	-	-	-	-	-	-	-	-
B_AND	●	●	●	●	●	●	●	●
B_OR	●	●	●	●	●	●	●	●
B_NOT	●	●	●	●	●	●	●	●
B_XOR	●	●	●	●	●	●	●	●
BAUTO	○	○	○	○	○	○	○	○
BLOCK	●	●	●	●	●	●	●	●
BLSYNC	●	●	●	●	●	●	●	●
BNAT	○	○	○	○	○	○	○	○
BOOL	●	●	●	●	●	●	●	●
BOUND	●	●	●	●	●	●	●	●
BRISK	●	●	●	●	●	●	●	●
BRISKA	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
BSPLINE	○	○	○	○	○	○	○	○
BTAN	○	○	○	○	○	○	○	○
C	●	●	●	●	●	●	●	●
C2	-	-	-	-	-	-	チャネル軸の名称	チャネル軸の名称
C3	-	-	-	-	-	-	-	-
C4	-	-	-	-	-	-	-	-
C5	-	-	-	-	-	-	-	-
C6	-	-	-	-	-	-	-	-
C7	-	-	-	-	-	-	-	-
CAC	●	●	●	●	●	●	●	●
CACN	●	●	●	●	●	●	●	●
CACP	●	●	●	●	●	●	●	●
CALCDAT	●	●	●	●	●	●	●	●
CALCPOSI	●	●	●	●	●	●	●	●
CALL	●	●	●	●	●	●	●	●
CALLPATH	●	●	●	●	●	●	●	●
CANCEL	●	●	●	●	●	●	●	●
CASE	●	●	●	●	●	●	●	●
CDC	●	●	●	●	●	●	●	●
CDOF	-	-	-	-	-	-	-	-
CDOF2	-	-	-	-	-	-	-	-
CDON	-	-	-	-	-	-	-	-
CFC	●	●	●	●	●	●	●	●
CFIN	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
CFINE	●	●	●	●	●	●	●	●
CFTCP	●	●	●	●	●	●	●	●
CHAN	●	●	●	●	●	●	●	●
CHANDATA	●	●	●	●	●	●	●	●
CHAR	●	●	●	●	●	●	●	●
CHF	●	●	●	●	●	●	●	●
CHKDM	●	●	●	●	●	●	●	●
CHKDNO	●	●	●	●	●	●	●	●
CHR	●	●	●	●	●	●	●	●
CIC	●	●	●	●	●	●	●	●
CIP	●	●	●	●	●	●	●	●
CLEARM	-	-	-	-	-	○	-	●
CLRINT	●	●	●	●	●	●	●	●
CMIRROR	●	●	●	●	●	●	●	●
COARSEA	●	●	●	●	●	●	●	●
COLLPAIR	-	-	-	-	-	-	-	-
COMPCAD	●	-	●	-	●	●	-	-
COMPCURV	●	-	●	-	●	●	-	-
COMPLETE	●	●	●	●	●	●	●	●
COMPOF	●	-	●	-	●	●	-	-
COMPON	●	-	●	-	●	●	-	-
COMPSURF	-	-	○	-	○	○	-	-
CONTDCON	●	●	●	●	●	●	●	●
CONTPRON	●	●	●	●	●	●	●	●
CORROF	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
CORRTRAFO	-	-	-	-	-	-	-	-
cos	●	●	●	●	●	●	●	●
COUPDEF	-	○	-	○	-	-	○	○
COUPDEL	-	○	-	○	-	-	○	○
COUPOF	-	○	-	○	-	-	○	○
COUPOFS	-	○	-	○	-	-	○	○
COUPON	-	○	-	○	-	-	○	○
COUPONC	-	○	-	○	-	-	○	○
COUPRES	-	○	-	○	-	-	○	○
CP	●	●	●	●	●	●	●	●
CPBC	○	○	○	○	○	○	○	○
CPDEF	○	○	○	○	○	○	○	○
CPDEL	○	○	○	○	○	○	○	○
CPFMOF	○	○	○	○	○	○	○	○
CPFMON	○	○	○	○	○	○	○	○
CPFMSON	○	○	○	○	○	○	○	○
CPFPOS	○	○	○	○	○	○	○	○
CPFRS	○	○	○	○	○	○	○	○
CPLA	○	○	○	○	○	○	○	○
CPLCTID	○	○	○	○	○	○	○	○
CPLDEF	○	○	○	○	○	○	○	○
CPLDEL	○	○	○	○	○	○	○	○
CPLDEN	○	○	○	○	○	○	○	○
CPLINSC	○	○	○	○	○	○	○	○
CPLINTR	○	○	○	○	○	○	○	○

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
CPLNUM	○	○	○	○	○	○	○	○
CPLOF	○	○	○	○	○	○	○	○
CPLON	○	○	○	○	○	○	○	○
CPLOUTSC	○	○	○	○	○	○	○	○
CPLOUTTR	○	○	○	○	○	○	○	○
CPLPOS	○	○	○	○	○	○	○	○
CPLSETVAL	○	○	○	○	○	○	○	○
CPMALARM	○	○	○	○	○	○	○	○
CPMBRAKE	○	○	○	○	○	○	○	○
CPMPRT	○	○	○	○	○	○	○	○
CPMRESET	○	○	○	○	○	○	○	○
CPMSTART	○	○	○	○	○	○	○	○
CPMVDI	○	○	○	○	○	○	○	○
CPOF	○	○	○	○	○	○	○	○
CPON	○	○	○	○	○	○	○	○
CPRECOF	●	●	●	●	●	●	●	●
CPRECON	●	●	●	●	●	●	●	●
CPRES	○	○	○	○	○	○	○	○
CPROT	●	●	●	●	●	●	●	●
CPROTDEF	●	●	●	●	●	●	●	●
CPSETTYPE	○	○	○	○	○	○	○	○
CPSYNCOP	○	○	○	○	○	○	○	○
CPSYNCOP2	○	○	○	○	○	○	○	○
CPSYNCOV	○	○	○	○	○	○	○	○
CPSYNFIP	○	○	○	○	○	○	○	○

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
CPSYNFIP2	○	○	○	○	○	○	○	○
CPSYNFIV	○	○	○	○	○	○	○	○
CR	●	●	●	●	●	●	●	●
CROT	●	●	●	●	●	●	●	●
CROTS	●	●	●	●	●	●	●	●
CRPL	●	●	●	●	●	●	●	●
CSCALE	●	●	●	●	●	●	●	●
CSPLINE	○	○	○	○	○	○	○	○
CT	●	●	●	●	●	●	●	●
CTAB	-	-	-	-	-	-	-	-
CTABDEF	-	-	-	-	-	-	-	-
CTABDEL	-	-	-	-	-	-	-	-
CTABEND	-	-	-	-	-	-	-	-
CTABEXISTS	-	-	-	-	-	-	-	-
CTABFNO	-	-	-	-	-	-	-	-
CTABFPOL	-	-	-	-	-	-	-	-
CTABFSEG	-	-	-	-	-	-	-	-
CTABID	-	-	-	-	-	-	-	-
CTABINV	-	-	-	-	-	-	-	-
CTABISLOCK	-	-	-	-	-	-	-	-
CTABLOCK	-	-	-	-	-	-	-	-
CTABMEMTYP	-	-	-	-	-	-	-	-
CTABMPOL	-	-	-	-	-	-	-	-
CTABMSEG	-	-	-	-	-	-	-	-

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
CTABNO	-	-	-	-	-	-	-	-
CTABNOMEM	-	-	-	-	-	-	-	-
CTABPERIOD	-	-	-	-	-	-	-	-
CTABPOL	-	-	-	-	-	-	-	-
CTABPOLID	-	-	-	-	-	-	-	-
CTABSEG	-	-	-	-	-	-	-	-
CTABSEGID	-	-	-	-	-	-	-	-
CTABSEV	-	-	-	-	-	-	-	-
CTABSSV	-	-	-	-	-	-	-	-
CTABTEP	-	-	-	-	-	-	-	-
CTABTEV	-	-	-	-	-	-	-	-
CTABTMAX	-	-	-	-	-	-	-	-
CTABTMIN	-	-	-	-	-	-	-	-
CTABTSP	-	-	-	-	-	-	-	-
CTABTSV	-	-	-	-	-	-	-	-
CTABUNLOCK	-	-	-	-	-	-	-	-
CTOL	●	●	●	●	●	●	●	●
CTRANS	●	●	●	●	●	●	●	●
CUT2D	●	●	●	●	●	●	●	●
CUT2DD	●	●	●	●	●	●	●	●
CUT2DF	●	●	●	●	●	●	●	●
CUT2DFD	●	●	●	●	●	●	●	●
CUT3DC	-	-	-	-	-	-	-	-
CUT3DCC	-	-	-	-	-	-	-	-
CUT3DCCD	-	-	-	-	-	-	-	-

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
CUT3DCD	-	-	-	-	-	-	-	-
CUT3DF	-	-	-	-	-	-	-	-
CUT3DFD	-	-	-	-	-	-	-	-
CUT3DFF	-	-	-	-	-	-	-	-
CUT3DFS	-	-	-	-	-	-	-	-
CUTCONOF	●	●	●	●	●	●	●	●
CUTCONON	●	●	●	●	●	●	●	●
CUTMOD	●	●	●	●	●	●	●	●
CUTMODK	-	-	-	-	-	-	-	-
CYCLE60	○	○	●	●	●	●	●	●
CYCLE61	●	●	●	●	●	●	●	●
CYCLE62	●	●	●	●	●	●	●	●
CYCLE63	○	○	●	●	●	●	●	●
CYCLE64	○	○	●	●	●	●	●	●
CYCLE70	○	○	●	●	●	●	●	●
CYCLE72	●	●	●	●	●	●	●	●
CYCLE76	●	●	●	●	●	●	●	●
CYCLE77	●	●	●	●	●	●	●	●
CYCLE78	○	○	●	●	●	●	●	●
CYCLE79	○	○	●	●	●	●	●	●
CYCLE81	●	●	●	●	●	●	●	●
CYCLE82	●	●	●	●	●	●	●	●
CYCLE83	●	●	●	●	●	●	●	●
CYCLE84	●	●	●	●	●	●	●	●
CYCLE85	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
CYCLE86	●	●	●	●	●	●	●	●
CYCLE92	●	●	●	●	●	●	●	●
CYCLE95	●	●	●	●	●	●	●	●
CYCLE98	●	●	●	●	●	●	●	●
CYCLE99	●	●	●	●	●	●	●	●
CYCLE150	○	○	○	○	○	○	○	○
CYCLE435	-	-	-	-	-	-	-	-
CYCLE495	-	-	-	-	-	-	-	-
CYCLE750	●	●	●	●	●	●	●	●
CYCLE751	●	●	●	●	●	●	●	●
CYCLE752	●	●	●	●	●	●	●	●
CYCLE753	●	●	●	●	●	●	●	●
CYCLE754	●	●	●	●	●	●	●	●
CYCLE755	●	●	●	●	●	●	●	●
CYCLE756	●	●	●	●	●	●	●	●
CYCLE757	●	●	●	●	●	●	●	●
CYCLE758	●	●	●	●	●	●	●	●
CYCLE759	●	●	●	●	●	●	●	●
CYCLE800	-	-	●	●	●	●	●	●
CYCLE801	●	●	●	●	●	●	●	●
CYCLE802	●	●	●	●	●	●	●	●
CYCLE830	○	○	●	●	●	●	●	●
CYCLE832	●	●	●	●	●	●	●	●
CYCLE840	●	●	●	●	●	●	●	●
CYCLE899	○	○	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
CYCLE930	●	●	●	●	●	●	●	●
CYCLE940	●	●	●	●	●	●	●	●
CYCLE951	●	●	●	●	●	●	●	●
CYCLE952	○	○	●	●	●	●	●	●
CYCLE961	○	○	○	○	○	○	○	○
CYCLE971	○	○	○	○	○	○	○	○
CYCLE973	○	○	○	○	○	○	○	○
CYCLE974	○	○	○	○	○	○	○	○
CYCLE976	○	○	○	○	○	○	○	○
CYCLE977	○	○	○	○	○	○	○	○
CYCLE978	○	○	○	○	○	○	○	○
CYCLE979	○	○	○	○	○	○	○	○
CYCLE982	○	○	○	○	○	○	○	○
CYCLE994	○	○	○	○	○	○	○	○
CYCLE995	○	○	○	○	○	○	○	○
CYCLE996	○	-	○	-	○	○	-	-
CYCLE997	○	○	○	○	○	○	○	○
CYCLE998	○	○	○	○	○	○	○	○
CYCLE4071	-	-	-	-	-	-	-	-
CYCLE4072	-	-	-	-	-	-	-	-
CYCLE4073	-	-	-	-	-	-	-	-
CYCLE4074	-	-	-	-	-	-	-	-
CYCLE4075	-	-	-	-	-	-	-	-
CYCLE4077	-	-	-	-	-	-	-	-

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
CYCLE4078	-	-	-	-	-	-	-	-
CYCLE4079	-	-	-	-	-	-	-	-

命令 D ... F

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
D	●	●	●	●	●	●	●	●
D0	●	●	●	●	●	●	●	●
DAC	●	●	●	●	●	●	●	●
DC	●	●	●	●	●	●	●	●
DCI	●	●	●	●	●	●	●	●
DCM	●	●	●	●	●	●	●	●
DCU	●	●	●	●	●	●	●	●
DEF	●	●	●	●	●	●	●	●
DEFINE	●	●	●	●	●	●	●	●
DEFAULT	●	●	●	●	●	●	●	●
DELAYFSTON	●	●	●	●	●	●	●	●
DELAYFSTOF	●	●	●	●	●	●	●	●
DELDL	●	●	●	●	●	●	●	●
DELDTG	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
DELETE	●	●	●	●	●	●	●	●
DELMLOWNER	●	●	●	●	●	●	●	●
DELMLRES	●	●	●	●	●	●	●	●
DELMT	●	●	●	●	●	●	●	●
DELOBJ	-	-	-	-	-	-	-	-
DELT	●	●	●	●	●	●	●	●
DELTC	●	●	●	●	●	●	●	●
DELTOOLENV	●	●	●	●	●	●	●	●
DIACYCOFA	●	●	●	●	●	●	●	●
DIAM90	●	●	●	●	●	●	●	●
DIAM90A	●	●	●	●	●	●	●	●
DIAMCHAN	●	●	●	●	●	●	●	●
DIAMCHANA	●	●	●	●	●	●	●	●
DIAMCYCOF	●	●	●	●	●	●	●	●
DIAMOF	●	●	●	●	●	●	●	●
DIAMOFA	●	●	●	●	●	●	●	●
DIAMON	●	●	●	●	●	●	●	●
DIAMONA	●	●	●	●	●	●	●	●
DIC	●	●	●	●	●	●	●	●
DILF	●	●	●	●	●	●	●	●
DISABLE	●	●	●	●	●	●	●	●
DISC	●	●	●	●	●	●	●	●
DISCL	●	●	●	●	●	●	●	●
DISPLOF	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
DISPLON	●	●	●	●	●	●	●	●
DISPR	●	●	●	●	●	●	●	●
DISR	●	●	●	●	●	●	●	●
DISRP	●	●	●	●	●	●	●	●
DITE	●	●	●	●	●	●	●	●
DITS	●	●	●	●	●	●	●	●
DIV	●	●	●	●	●	●	●	●
DI	-	-	-	-	-	-	-	-
DO	●	●	●	●	●	●	●	●
DRFOF	●	●	●	●	●	●	●	●
DRIVE	●	●	●	●	●	●	●	●
DRIVEA	●	●	●	●	●	●	●	●
DYNFINISH	●	●	●	●	●	●	●	●
DYNNORM	●	●	●	●	●	●	●	●
DYNPOS	●	●	●	●	●	●	●	●
DYNROUGH	●	●	●	●	●	●	●	●
DYNSEMIFIN	●	●	●	●	●	●	●	●
DZERO	●	●	●	●	●	●	●	●
EAUTO	○	○	○	○	○	○	○	○
EGDEF	-	○	-	○	-	-	○	○
EGDEL	-	○	-	○	-	-	○	○
EGOFC	-	○	-	○	-	-	○	○
EGOFS	-	○	-	○	-	-	○	○
EGON	-	○	-	○	-	-	○	○
EGONSYN	-	○	-	○	-	-	○	○

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
EGONSYNE	-	○	-	○	-	-	○	○
ELSE	●	●	●	●	●	●	●	●
ENABLE	●	●	●	●	●	●	●	●
ENAT	○	○	○	○	○	○	○	○
ENDFOR	●	●	●	●	●	●	●	●
ENDIF	●	●	●	●	●	●	●	●
ENDLABEL	●	●	●	●	●	●	●	●
ENDLOOP	●	●	●	●	●	●	●	●
ENDPROC	●	●	●	●	●	●	●	●
ENDWHILE	●	●	●	●	●	●	●	●
ESRR	○	○	○	○	○	○	○	○
ESRS	○	○	○	○	○	○	○	○
ETAN	○	○	○	○	○	○	○	○
EVERY	●	●	●	●	●	●	●	●
EX	●	●	●	●	●	●	●	●
EXECSTRING	●	●	●	●	●	●	●	●
EXECTAB	●	●	●	●	●	●	●	●
EXECUTE	●	●	●	●	●	●	●	●
EXP	●	●	●	●	●	●	●	●
EXTCALL	●	●	●	●	●	●	●	●
EXTCLOSE	●	●	●	●	●	●	●	●
EXTERN	●	●	●	●	●	●	●	●
EXTOPEN	●	●	●	●	●	●	●	●
F	●	●	●	●	●	●	●	●
FA	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
FAD	●	●	●	●	●	●	●	●
FALSE	●	●	●	●	●	●	●	●
FB	●	●	●	●	●	●	●	●
FCTDEF	●	●	●	●	●	●	●	●
FCUB	●	●	●	●	●	●	●	●
FD	●	●	●	●	●	●	●	●
FDA	●	●	●	●	●	●	●	●
FENDNORM	●	●	●	●	●	●	●	●
FFWOF	●	●	●	●	●	●	●	●
FFWON	●	●	●	●	●	●	●	●
FGREF	●	●	●	●	●	●	●	●
FGROUP	●	●	●	●	●	●	●	●
FI	●	●	●	●	●	●	●	●
FIFOCTRL	●	●	●	●	●	●	●	●
FILEDATE	●	●	●	●	●	●	●	●
FILEINFO	●	●	●	●	●	●	●	●
FILESIZE	●	●	●	●	●	●	●	●
FILESTAT	●	●	●	●	●	●	●	●
FILETIME	●	●	●	●	●	●	●	●
FINEA	●	●	●	●	●	●	●	●
FL	●	●	●	●	●	●	●	●
FLIN	●	●	●	●	●	●	●	●
FMA	●	●	●	●	●	●	●	●
FNORM	●	●	●	●	●	●	●	●
FOCOF	○	○	○	○	○	○	○	○

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
FOCON	○	○	○	○	○	○	○	○
FOR	●	●	●	●	●	●	●	●
FP	●	●	●	●	●	●	●	●
FPO	-	-	-	-	-	-	-	-
FPR	●	●	●	●	●	●	●	●
FPRAOF	●	●	●	●	●	●	●	●
FPRAON	●	●	●	●	●	●	●	●
FRAME	●	●	●	●	●	●	●	●
FRC	●	●	●	●	●	●	●	●
FRCM	●	●	●	●	●	●	●	●
FROM	●	●	●	●	●	●	●	●
FTOC	●	●	●	●	●	●	●	●
FTOCOF	●	●	●	●	●	●	●	●
FTOCON	●	●	●	●	●	●	●	●
FXS	●	●	●	●	●	●	●	●
FXST	●	●	●	●	●	●	●	●
FXSW	●	●	●	●	●	●	●	●
FZ	●	●	●	●	●	●	●	●

命令 G ... L

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
G0	●	●	●	●	●	●	●	●
G1	●	●	●	●	●	●	●	●
G2	●	●	●	●	●	●	●	●
G3	●	●	●	●	●	●	●	●
G4	●	●	●	●	●	●	●	●
G5	●	●	●	●	●	●	●	●
G7	●	●	●	●	●	●	●	●
G9	●	●	●	●	●	●	●	●
G17	●	●	●	●	●	●	●	●
G18	●	●	●	●	●	●	●	●
G19	●	●	●	●	●	●	●	●
G25	●	●	●	●	●	●	●	●
G26	●	●	●	●	●	●	●	●
G33	●	●	●	●	●	●	●	●
G34	●	●	●	●	●	●	●	●
G35	●	●	●	●	●	●	●	●
G40	●	●	●	●	●	●	●	●
G41	●	●	●	●	●	●	●	●
G42	●	●	●	●	●	●	●	●
G53	●	●	●	●	●	●	●	●
G54	●	●	●	●	●	●	●	●
G55	●	●	●	●	●	●	●	●
G56	●	●	●	●	●	●	●	●
G57	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
G58	→ G505							
G59	→ G506							
G60	●	●	●	●	●	●	●	●
G62	●	●	●	●	●	●	●	●
G63	●	●	●	●	●	●	●	●
G64	●	●	●	●	●	●	●	●
G70	●	●	●	●	●	●	●	●
G71	●	●	●	●	●	●	●	●
G74	●	●	●	●	●	●	●	●
G75	●	●	●	●	●	●	●	●
G90	●	●	●	●	●	●	●	●
G91	●	●	●	●	●	●	●	●
G93	●	●	●	●	●	●	●	●
G94	●	●	●	●	●	●	●	●
G95	●	●	●	●	●	●	●	●
G96	●	●	●	●	●	●	●	●
G97	●	●	●	●	●	●	●	●
G110	●	●	●	●	●	●	●	●
G111	●	●	●	●	●	●	●	●
G112	●	●	●	●	●	●	●	●
G140	●	●	●	●	●	●	●	●
G141	●	●	●	●	●	●	●	●
G142	●	●	●	●	●	●	●	●
G143	●	●	●	●	●	●	●	●
G147	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
G148	●	●	●	●	●	●	●	●
G153	●	●	●	●	●	●	●	●
G247	●	●	●	●	●	●	●	●
G248	●	●	●	●	●	●	●	●
G290	●	●	●	●	●	●	●	●
G291	●	●	●	●	●	●	●	●
G331	●	●	●	●	●	●	●	●
G332	●	●	●	●	●	●	●	●
G335	●	●	●	●	●	●	●	●
G336	●	●	●	●	●	●	●	●
G340	●	●	●	●	●	●	●	●
G341	●	●	●	●	●	●	●	●
G347	●	●	●	●	●	●	●	●
G348	●	●	●	●	●	●	●	●
G450	●	●	●	●	●	●	●	●
G451	●	●	●	●	●	●	●	●
G460	●	●	●	●	●	●	●	●
G461	●	●	●	●	●	●	●	●
G462	●	●	●	●	●	●	●	●
G500	●	●	●	●	●	●	●	●
G505 ～ G599	●	●	●	●	●	●	●	●
G601	●	●	●	●	●	●	●	●
G602	●	●	●	●	●	●	●	●
G603	●	●	●	●	●	●	●	●
G621	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
G641	●	●	●	●	●	●	●	●
G642	●	●	●	●	●	●	●	●
G643	●	●	●	●	●	●	●	●
G644	●	●	●	●	●	●	●	●
G645	●	●	●	●	●	●	●	●
G700	●	●	●	●	●	●	●	●
G710	●	●	●	●	●	●	●	●
G810 ～ G819	-	-	-	-	-	-	-	-
G820 ～ G829	-	-	-	-	-	-	-	-
G931	●	●	●	●	●	●	●	●
G942	●	●	●	●	●	●	●	●
G952	●	●	●	●	●	●	●	●
G961	●	●	●	●	●	●	●	●
G962	●	●	●	●	●	●	●	●
G971	●	●	●	●	●	●	●	●
G972	●	●	●	●	●	●	●	●
G973	●	●	●	●	●	●	●	●
GEOAX	●	●	●	●	●	●	●	●
GET	●	●	●	●	●	●	●	●
GETACTT	●	●	●	●	●	●	●	●
GETACTTD	●	●	●	●	●	●	●	●
GETD	-	-	-	-	-	○	-	●
GETDNO	●	●	●	●	●	●	●	●
GETEXET	●	●	●	●	●	●	●	●
GETFREELOC	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
GETSELT	●	●	●	●	●	●	●	●
GETT	●	●	●	●	●	●	●	●
GETTCOR	●	●	●	●	●	●	●	●
GETTENV	●	●	●	●	●	●	●	●
GETVARAP	●	●	●	●	●	●	●	●
GETVARDFT	●	●	●	●	●	●	●	●
GETVARLIM	●	●	●	●	●	●	●	●
GETVARPHU	●	●	●	●	●	●	●	●
GETVARTYP	●	●	●	●	●	●	●	●
GFRAME0 ... GFRAME100	-	-	-	-	-	-	-	-
GOTO	●	●	●	●	●	●	●	●
GOTOB	●	●	●	●	●	●	●	●
GOTOC	●	●	●	●	●	●	●	●
GOTOF	●	●	●	●	●	●	●	●
GOTOS	●	●	●	●	●	●	●	●
BP	●	●	●	●	●	●	●	●
GWPSOF	●	●	●	●	●	●	●	●
GROUP_ ADDEND	●	●	●	●	●	●	●	●
GROUP_BEGIN	●	●	●	●	●	●	●	●
GROUP_END	●	●	●	●	●	●	●	●
GWPSON	●	●	●	●	●	●	●	●
H...	●	●	●	●	●	●	●	●
HOLES1	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
HOLES2	●	●	●	●	●	●	●	●
I	●	●	●	●	●	●	●	●
I1	●	●	●	●	●	●	●	●
IC	●	●	●	●	●	●	●	●
ICYCOF	●	●	●	●	●	●	●	●
ICYCON	●	●	●	●	●	●	●	●
ID	●	●	●	●	●	●	●	●
IDS	●	●	●	●	●	●	●	●
IF	●	●	●	●	●	●	●	●
INDEX	●	●	●	●	●	●	●	●
INIPO	●	●	●	●	●	●	●	●
INIRE	●	●	●	●	●	●	●	●
INICF	●	●	●	●	●	●	●	●
INIT	-	-	-	-	-	○	-	●
INITIAL								
INT	●	●	●	●	●	●	●	●
INTERSEC	●	●	●	●	●	●	●	●
INVCCW	-	-	-	-	-	-	-	-
INVCW	-	-	-	-	-	-	-	-
INVFRAME	●	●	●	●	●	●	●	●
IP	●	●	●	●	●	●	●	●
IPOBRKA	●	●	●	●	●	●	●	●
IPOENDA	●	●	●	●	●	●	●	●
IPTRLOCK	●	●	●	●	●	●	●	●
IPTRUNLOCK	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
IR	●	●	●	●	●	●	●	●
ISAXIS	●	●	●	●	●	●	●	●
ISD	-	-	-	-	-	-	-	-
ISFILE	●	●	●	●	●	●	●	●
ISNUMBER	●	●	●	●	●	●	●	●
ISOCALL	●	●	●	●	●	●	●	●
ISVAR	●	●	●	●	●	●	●	●
J	●	●	●	●	●	●	●	●
J1	●	●	●	●	●	●	●	●
JERKA	●	●	●	●	●	●	●	●
JERKLIM	●	●	●	●	●	●	●	●
JERKLIMA	●	●	●	●	●	●	●	●
JR	●	●	●	●	●	●	●	●
K	●	●	●	●	●	●	●	●
K1	●	●	●	●	●	●	●	●
KONT	●	●	●	●	●	●	●	●
KONTC	●	●	●	●	●	●	●	●
KONTT	●	●	●	●	●	●	●	●
KR	●	●	●	●	●	●	●	●
L	●	●	●	●	●	●	●	●
LEAD 工具オリエンテーション 旋回多項式	- -	- -	- -	- -	- -	- -	- -	- -
LEADOF	-	-	-	●	-	-	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
LEADON	-	-	-	●	-	-	●	●
LENTOAX	●	●	●	●	●	●	●	●
LFOF	●	●	●	●	●	●	●	●
LFON	●	●	●	●	●	●	●	●
LFPOS	●	●	●	●	●	●	●	●
LFTXT	●	●	●	●	●	●	●	●
LFWP	●	●	●	●	●	●	●	●
LIFTFAST	●	●	●	●	●	●	●	●
LIMS	●	●	●	●	●	●	●	●
LLI	●	●	●	●	●	●	●	●
LN	●	●	●	●	●	●	●	●
LOCK	●	●	●	●	●	●	●	●
LONGHOLE	●	●	●	●	●	●	●	●
LOOP	●	●	●	●	●	●	●	●

命令 M ... R

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
M0	●	●	●	●	●	●	●	●
M1	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
M2	●	●	●	●	●	●	●	●
M3	●	●	●	●	●	●	●	●
M4	●	●	●	●	●	●	●	●
M5	●	●	●	●	●	●	●	●
M6	●	●	●	●	●	●	●	●
M17	●	●	●	●	●	●	●	●
M19	●	●	●	●	●	●	●	●
M30	●	●	●	●	●	●	●	●
M40	●	●	●	●	●	●	●	●
M41 ... M45	●	●	●	●	●	●	●	●
M70	●	●	●	●	●	●	●	●
MASLDEF	-	-	-	-	-	-	-	-
MASLDEL	-	-	-	-	-	-	-	-
MASLOF	-	-	-	-	-	-	-	-
MASLOFS	-	-	-	-	-	-	-	-
MASLON	-	-	-	-	-	-	-	-
MATCH	●	●	●	●	●	●	●	●
MAXVAL	●	●	●	●	●	●	●	●
MCALL	●	●	●	●	●	●	●	●
MEAC	-	-	-	-	-	-	-	-
MEAFRAME	●	●	●	●	●	●	●	●
MEAS	●	●	●	●	●	●	●	●
MEASA	-	-	-	-	-	-	-	-
MEASURE	●	●	●	●	●	●	●	●
MEAW	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
MEAWA	-	-	-	-	-	-	-	-
Mi	●	●	●	●	●	●	●	●
MINDEX	●	●	●	●	●	●	●	●
MINVAL	●	●	●	●	●	●	●	●
MIRROR	●	●	●	●	●	●	●	●
MMC	●	●	●	●	●	●	●	●
MOD	●	●	●	●	●	●	●	●
MODAXVAL	●	●	●	●	●	●	●	●
MOV	●	●	●	●	●	●	●	●
MOVT	●	●	●	●	●	●	●	●
MSG	●	●	●	●	●	●	●	●
MVTOOL	●	●	●	●	●	●	●	●
N	●	●	●	●	●	●	●	●
NAMETOINT	●	●	●	●	●	●	●	●
NC	●	●	●	●	●	●	●	●
NEWCONF	●	●	●	●	●	●	●	●
NEWMT	●	●	●	●	●	●	●	●
NEWT	●	●	●	●	●	●	●	●
NORM	●	●	●	●	●	●	●	●
NOT	●	●	●	●	●	●	●	●
NPROT	●	●	●	●	●	●	●	●
NPROTDEF	●	●	●	●	●	●	●	●
NUMBER	●	●	●	●	●	●	●	●
OEMIPO1	-	-	-	-	-	-	-	-
OEMIPO2	-	-	-	-	-	-	-	-

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
OF	●	●	●	●	●	●	●	●
OFFN	●	●	●	●	●	●	●	●
OMA1	-	-	-	-	-	-	-	-
OMA2	-	-	-	-	-	-	-	-
OMA3	-	-	-	-	-	-	-	-
OMA4	-	-	-	-	-	-	-	-
OMA5	-	-	-	-	-	-	-	-
OR	●	●	●	●	●	●	●	●
ORIAXES	-	-	-	-	-	-	-	-
ORIAXPOS	-	-	-	-	-	-	-	-
ORIC	-	-	-	-	-	-	-	-
ORICONCCW	-	-	-	-	-	-	-	-
ORICONCW	-	-	-	-	-	-	-	-
ORICONIO	-	-	-	-	-	-	-	-
ORICONTO	-	-	-	-	-	-	-	-
ORICURVE	-	-	-	-	-	-	-	-
ORID	-	-	-	-	-	-	-	-
ORIEULER	-	-	-	-	-	-	-	-
ORIMKS	-	-	-	-	-	-	-	-
ORIPATH	-	-	-	-	-	-	-	-
ORIPATHS	-	-	-	-	-	-	-	-
ORIPLANE	-	-	-	-	-	-	-	-
ORIRESET	-	-	-	-	-	-	-	-
ORIROTA	-	-	-	-	-	-	-	-
ORIROTC	-	-	-	-	-	-	-	-

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
ORIROTR	-	-	-	-	-	-	-	-
ORIROTT	-	-	-	-	-	-	-	-
ORIRPY	-	-	-	-	-	-	-	-
ORIRPY2	-	-	-	-	-	-	-	-
ORIS	-	-	-	-	-	-	-	-
ORISOF	-	-	-	-	-	-	-	-
ORISOLH	-	-	-	-	-	-	-	-
ORISON	-	-	-	-	-	-	-	-
ORIVECT	-	-	-	-	-	-	-	-
ORIVIRT1	-	-	-	-	-	-	-	-
ORIVIRT2	-	-	-	-	-	-	-	-
ORIWKS	-	-	-	-	-	-	-	-
OS	-	-	-	-	-	-	-	-
OSB	-	-	-	-	-	-	-	-
OSC	-	-	-	-	-	-	-	-
OSCILL	-	-	-	-	-	-	-	-
OSCTRL	-	-	-	-	-	-	-	-
OSD	-	-	-	-	-	-	-	-
OSE	-	-	-	-	-	-	-	-
OSNSC	●	●	●	●	●	●	●	●
OSOF	-	-	-	-	-	-	-	-
OSP1	●	●	●	●	●	●	●	●
OSP2	●	●	●	●	●	●	●	●
OSS	-	-	-	-	-	-	-	-
OSSE	-	-	-	-	-	-	-	-

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
OST	-	-	-	-	-	-	-	-
OST1	●	●	●	●	●	●	●	●
OST2	●	●	●	●	●	●	●	●
OTOL	●	●	●	●	●	●	●	●
OVR	●	●	●	●	●	●	●	●
OVRA	●	●	●	●	●	●	●	●
OVRRAP	●	●	●	●	●	●	●	●
P	●	●	●	●	●	●	●	●
PAROT	●	●	●	●	●	●	●	●
PAROTOF	●	●	●	●	●	●	●	●
PCALL	●	●	●	●	●	●	●	●
PDELAYOF	-	-	-	-	-	-	-	-
PDELAYON	-	-	-	-	-	-	-	-
PHI	-	-	-	-	-	-	-	-
PHU	●	●	●	●	●	●	●	●
PL	-	-	-	-	-	-	-	-
PM	●	●	●	●	●	●	●	●
PO	-	-	-	-	-	-	-	-
POCKET3	●	●	●	●	●	●	●	●
POCKET4	●	●	●	●	●	●	●	●
POLF	●	●	●	●	●	●	●	●
POLFA	●	●	●	●	●	●	●	●
POLFMASK	●	●	●	●	●	●	●	●
POLFMLIN	●	●	●	●	●	●	●	●
POLY	-	-	-	-	-	-	-	-

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
POLYPATH	-	-	-	-	-	-	-	-
PON	-	-	-	-	-	-	-	-
PONS	-	-	-	-	-	-	-	-
POS	●	●	●	●	●	●	●	●
POSA	●	●	●	●	●	●	●	●
POSM	●	●	●	●	●	●	●	●
POSMT	●	●	●	●	●	●	●	●
POSP	●	●	●	●	●	●	●	●
POSRANGE	●	●	●	●	●	●	●	●
POT	●	●	●	●	●	●	●	●
PR	●	●	●	●	●	●	●	●
PREPRO	●	●	●	●	●	●	●	●
PRESETON	●	●	●	●	●	●	●	●
PRESETONS	●	●	●	●	●	●	●	●
PRIO	●	●	●	●	●	●	●	●
PRLOC	●	●	●	●	●	●	●	●
PROC	●	●	●	●	●	●	●	●
PROTA	●	●	●	●	●	●	●	●
PROTD	●	●	●	●	●	●	●	●
PROTS	●	●	●	●	●	●	●	●
psi	-	-	-	-	-	-	-	-
PTP	●	●	●	●	●	●	●	●
PTPG0	●	●	●	●	●	●	●	●
PTPWOC	●	●	●	●	●	●	●	●
PUNCHACC	-	-	-	-	-	-	-	-

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
PUTFTOC	●	●	●	●	●	●	●	●
PUTFTOCF	●	●	●	●	●	●	●	●
PW	○	○	○	○	○	○	○	○
QU	●	●	●	●	●	●	●	●
R...	●	●	●	●	●	●	●	●
RAC	●	●	●	●	●	●	●	●
RDISABLE	●	●	●	●	●	●	●	●
READ	●	●	●	●	●	●	●	●
REAL	●	●	●	●	●	●	●	●
RELEASE	●	●	●	●	●	●	●	●
REP	●	●	●	●	●	●	●	●
REPEAT	●	●	●	●	●	●	●	●
REPEATB	●	●	●	●	●	●	●	●
REPOSA	●	●	●	●	●	●	●	●
REPOSH	●	●	●	●	●	●	●	●
REPOSHA	●	●	●	●	●	●	●	●
REPOSL	●	●	●	●	●	●	●	●
REPOSQ	●	●	●	●	●	●	●	●
REPOSQA	●	●	●	●	●	●	●	●
RESETMON	●	●	●	●	●	●	●	●
RET	●	●	●	●	●	●	●	●
RETB	●	●	●	●	●	●	●	●
RIC	●	●	●	●	●	●	●	●
RINDEX	●	●	●	●	●	●	●	●
RMB	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
RME	●	●	●	●	●	●	●	●
RMI	●	●	●	●	●	●	●	●
RMN	●	●	●	●	●	●	●	●
RND	●	●	●	●	●	●	●	●
RNDM	●	●	●	●	●	●	●	●
ROT	●	●	●	●	●	●	●	●
ROTS	●	●	●	●	●	●	●	●
ROUND	●	●	●	●	●	●	●	●
ROUNDUP	●	●	●	●	●	●	●	●
RP	●	●	●	●	●	●	●	●
RPL	●	●	●	●	●	●	●	●
RT	●	●	●	●	●	●	●	●
RTLIOF	●	●	●	●	●	●	●	●
RTLION	●	●	●	●	●	●	●	●

命令 S ... Z

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
S	●	●	●	●	●	●	●	●
SAVE	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
SBLOF	●	●	●	●	●	●	●	●
SBLON	●	●	●	●	●	●	●	●
SC	●	●	●	●	●	●	●	●
SCALE	●	●	●	●	●	●	●	●
SCC	●	●	●	●	●	●	●	●
SCPARA	●	●	●	●	●	●	●	●
SD	○	○	○	○	○	○	○	○
SET	●	●	●	●	●	●	●	●
SETAL	●	●	●	●	●	●	●	●
SETDNO	●	●	●	●	●	●	●	●
SETINT	●	●	●	●	●	●	●	●
SETM	-	-	-	-	-	○	-	●
SETMS	●	●	●	●	●	●	●	●
SETMS(n)	●	●	●	●	●	●	●	●
SETMTH	●	●	●	●	●	●	●	●
SETPIECE	●	●	●	●	●	●	●	●
SETTA	●	●	●	●	●	●	●	●
SETTCOR	●	●	●	●	●	●	●	●
SETTIA	●	●	●	●	●	●	●	●
SF	●	●	●	●	●	●	●	●
sin	●	●	●	●	●	●	●	●
SIRELAY	-	-	-	-	-	-	-	-
SIRELIN	-	-	-	-	-	-	-	-
SIRELOUT	-	-	-	-	-	-	-	-
SIRELTIME	-	-	-	-	-	-	-	-

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
SLOT1	●	●	●	●	●	●	●	●
SLOT2	●	●	●	●	●	●	●	●
SOFT	●	●	●	●	●	●	●	●
SOFTA	●	●	●	●	●	●	●	●
SON	-	-	-	-	-	-	-	-
SONS	-	-	-	-	-	-	-	-
SPATH	●	●	●	●	●	●	●	●
SPCOF	●	●	●	●	●	●	●	●
SPCON	●	●	●	●	●	●	●	●
SPI	●	●	●	●	●	●	●	●
SPIF1	-	-	-	-	-	-	-	-
SPIF2	-	-	-	-	-	-	-	-
SPLINEPATH	○	○	○	○	○	○	○	○
SPN; ＳＰＮ	-	-	-	-	-	-	-	-
SPOF	-	-	-	-	-	-	-	-
SPOS	●	●	●	●	●	●	●	●
SPOSA	●	●	●	●	●	●	●	●
SPP	-	-	-	-	-	-	-	-
SPRINT	●	●	●	●	●	●	●	●
SQRT	●	●	●	●	●	●	●	●
SR	●	●	●	●	●	●	●	●
SRA	●	●	●	●	●	●	●	●
ST	●	●	●	●	●	●	●	●
STA	●	●	●	●	●	●	●	●
START	-	-	-	-	-	○	-	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
STARTFIFO	●	●	●	●	●	●	●	●
STAT	●	●	●	●	●	●	●	●
STOLF	●	●	●	●	●	●	●	●
STOPFIFO	●	●	●	●	●	●	●	●
STOPRE	●	●	●	●	●	●	●	●
STOPREOF	●	●	●	●	●	●	●	●
STRING	●	●	●	●	●	●	●	●
STRINGFELD	●	●	●	●	●	●	●	●
STRINGIS	●	●	●	●	●	●	●	●
STRLEN	●	●	●	●	●	●	●	●
SUBSTR	●	●	●	●	●	●	●	●
SUPA	●	●	●	●	●	●	●	●
SVC	●	●	●	●	●	●	●	●
SYNFCT	●	●	●	●	●	●	●	●
SYNR	●	●	●	●	●	●	●	●
SYNRW	●	●	●	●	●	●	●	●
SYNW	●	●	●	●	●	●	●	●
T	●	●	●	●	●	●	●	●
TAN	●	●	●	●	●	●	●	●
TANG	-	-	-	-	-	-	-	-
TANGDEL	-	-	-	-	-	-	-	-
TANGOF	-	-	-	-	-	-	-	-
TANGON	-	-	-	-	-	-	-	-
TCA (828D: _TCA)	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
TCARR	●	-	●	-	●	●	-	-
TCI	●	●	●	●	●	●	●	●
TCOABS	●	-	●	-	●	●	-	-
TCOFR	●	-	●	-	●	●	-	-
TCOFRX	●	-	●	-	●	●	-	-
TCOFRY	●	-	●	-	●	●	-	-
TCOFRZ	●	-	●	-	●	●	-	-
THETA	-	-	-	-	-	-	-	-
TILT	-	-	-	-	-	-	-	-
TLIFT	-	-	-	-	-	-	-	-
TML	●	●	●	●	●	●	●	●
TMOF	●	●	●	●	●	●	●	●
TMON	●	●	●	●	●	●	●	●
TO	●	●	●	●	●	●	●	●
TOFF	●	●	●	●	●	●	●	●
TOFFL	●	●	●	●	●	●	●	●
TOFFOF	●	●	●	●	●	●	●	●
TOFFON	●	●	●	●	●	●	●	●
TOFFR	●	●	●	●	●	●	●	●
TOFRAME	●	●	●	●	●	●	●	●
TOFRAMEX	●	●	●	●	●	●	●	●
TOFRAMEY	●	●	●	●	●	●	●	●
TOFRAMEZ	●	●	●	●	●	●	●	●
TOLOWER	●	●	●	●	●	●	●	●
TOOLENV	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
TOOLGNT	●	●	●	●	●	●	●	●
TOOLGT	●	●	●	●	●	●	●	●
TOROT	●	●	●	●	●	●	●	●
TOROTOF	●	●	●	●	●	●	●	●
TOROTX	●	●	●	●	●	●	●	●
TOROTY	●	●	●	●	●	●	●	●
TOROTZ	●	●	●	●	●	●	●	●
TOUPPER	●	●	●	●	●	●	●	●
TOWBCS	●	-	●	-	●	●	-	-
TOWKCS	●	-	●	-	●	●	-	-
TOWMCS	●	-	●	-	●	●	-	-
TOWSTD	●	-	●	-	●	●	-	-
TOWTCS	●	-	●	-	●	●	-	-
TOWWCS	●	-	●	-	●	●	-	-
TR	●	●	●	●	●	●	●	●
TRAANG	-	-	-	-	-	-	○	○
TRACON	-	-	-	-	-	-	○	○
TRACYL	○	○	○	○	○	○	○	○
TRAFOOF	●	●	●	●	●	●	●	●
TRAFOON	-	-	-	-	-	-	-	-
TRAILOF	●	●	●	●	●	●	●	●
TRAILON	●	●	●	●	●	●	●	●
TRANS	●	●	●	●	●	●	●	●
TRANSMIT	○	○	○	○	○	○	○	○
TRAORI	-	-	-	-	-	-	-	-

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
TRUE	●	●	●	●	●	●	●	●
TRUNC	●	●	●	●	●	●	●	●
TU	●	●	●	●	●	●	●	●
TURN	●	●	●	●	●	●	●	●
ULI	●	●	●	●	●	●	●	●
UNLOCK	●	●	●	●	●	●	●	●
UNTIL	●	●	●	●	●	●	●	●
UPATH	●	●	●	●	●	●	●	●
var	●	●	●	●	●	●	●	●
VELOLIM	●	●	●	●	●	●	●	●
VELOLIMA	●	●	●	●	●	●	●	●
WAITC	●	●	●	●	●	●	●	●
WAITE	-	-	-	-	-	○	-	●
WAITENC	●	●	●	●	●	●	●	●
WAITM	-	-	-	-	-	○	-	●
WAITMC	-	-	-	-	-	○	-	●
WAITP	●	●	●	●	●	●	●	●
WAITS	●	●	●	●	●	●	●	●
WALCS0	●	●	●	●	●	●	●	●
WALCS1	●	●	●	●	●	●	●	●
WALCS2	●	●	●	●	●	●	●	●
WALCS3	●	●	●	●	●	●	●	●
WALCS4	●	●	●	●	●	●	●	●
WALCS5	●	●	●	●	●	●	●	●
WALCS6	●	●	●	●	●	●	●	●

命令	SINUMERIK 828D							
● 標準 ○ オプション - なし	SW24x(5) CNC-SW フライス削り エクスポート (me42)	SW24x(5) CNC-SW 旋削 エクスポート (te42)	SW26x(3) CNC-SW フライス削り エクスポート (me62)	SW26x(3) CNC-SW 旋削 エクスポート (te62)	SW28x(2) CNC-SW フライス削り エクスポート (me821)	SW28x(1) CNC-SW フライス削り 高度なエクスポート (me822)	SW28x(2) CNC-SW 旋削 エクスポート (te821)	SW28x(1) CNC-SW 旋削 高度なエクスポート (te822)
WALCS7	●	●	●	●	●	●	●	●
WALCS8	●	●	●	●	●	●	●	●
WALCS9	●	●	●	●	●	●	●	●
WALCS10	●	●	●	●	●	●	●	●
WALIMOF	●	●	●	●	●	●	●	●
WALIMON	●	●	●	●	●	●	●	●
WHEN	●	●	●	●	●	●	●	●
WHENEVER	●	●	●	●	●	●	●	●
WHILE	●	●	●	●	●	●	●	●
WORKPIECE	●	●	●	●	●	●	●	●
WRITE	●	●	●	●	●	●	●	●
WRTPR	●	●	●	●	●	●	●	●
X	●	●	●	●	●	●	●	●
XOR	●	●	●	●	●	●	●	●
Y	●	●	●	●	●	●	●	●
Z	●	●	●	●	●	●	●	●

21.2.2 制御タイプ 研削

命令 A ... C

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
:	●	●	●	●	●	●
*	●	●	●	●	●	●
+	●	●	●	●	●	●
-	●	●	●	●	●	●
<	●	●	●	●	●	●
<<	●	●	●	●	●	●
<=	●	●	●	●	●	●
=	●	●	●	●	●	●
>=	●	●	●	●	●	●
/	●	●	●	●	●	●
/0 … /7	●	●	●	●	●	●
A	●	●	●	●	●	●
A2	-	-	-	-	-	-
A3	-	-	-	-	-	-
A4	-	-	-	-	-	-
A5	-	-	-	-	-	-
A6	-	-	-	-	-	-
A7	-	-	-	-	-	-
ABS	●	●	●	●	●	●
AC	●	●	●	●	●	●
ACC	●	●	●	●	●	●
ACCLIMA	●	●	●	●	●	●
ACN	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
ACOS	●	●	●	●	●	●
ACP	●	●	●	●	●	●
ACTBLOCNO	●	●	●	●	●	●
ADDFRAME	●	●	●	●	●	●
ADIS	●	●	●	●	●	●
ADISPOS	●	●	●	●	●	●
ADISPOSA	●	●	●	●	●	●
ALF	●	●	●	●	●	●
AMIRROR	●	●	●	●	●	●
および	●	●	●	●	●	●
ANG	●	●	●	●	●	●
AP	●	●	●	●	●	●
APR	●	●	●	●	●	●
APRB	●	●	●	●	●	●
APRP	●	●	●	●	●	●
APW	●	●	●	●	●	●
APWB	●	●	●	●	●	●
APWP	●	●	●	●	●	●
APX	●	●	●	●	●	●
AR	●	●	●	●	●	●
AROT	●	●	●	●	●	●
AROTS	●	●	●	●	●	●
AS	●	●	●	●	●	●
ASCALE	●	●	●	●	●	●
ASIN	●	●	●	●	●	●
ASPLINE	○	○	○	○	○	○
ATAN2	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
ATOL	●	●	●	●	●	●
ATRANS	●	●	●	●	●	●
AUXFUDEL	●	●	●	●	●	●
AUXFUDELG	●	●	●	●	●	●
AUXFUMSEQ	●	●	●	●	●	●
AUXFUSYNC	●	●	●	●	●	●
AX	●	●	●	●	●	●
AXCTSWE	-	-	-	-	-	-
AXCTSWEC	-	-	-	-	-	-
AXCTSWED	-	-	-	-	-	-
AXIS	●	●	●	●	●	●
AXNAME	●	●	●	●	●	●
AXSTRING	●	●	●	●	●	●
AXTOCHAN	●	●	●	●	●	●
AXTOSPI	●	●	●	●	●	●
B	●	●	●	●	●	●
B2	-	-	-	-	-	-
B3	-	-	-	-	-	-
B4	-	-	-	-	-	-
B5	-	-	-	-	-	-
B6	-	-	-	-	-	-
B7	-	-	-	-	-	-
B_AND	●	●	●	●	●	●
B_OR	●	●	●	●	●	●
B_NOT	●	●	●	●	●	●
B_XOR	●	●	●	●	●	●
BAUTO	○	○	○	○	○	○

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
BLOCK	●	●	●	●	●	●
BLSYNC	●	●	●	●	●	●
BNAT	○	○	○	○	○	○
BOOL	●	●	●	●	●	●
BOUND	●	●	●	●	●	●
BRISK	●	●	●	●	●	●
BRISKA	●	●	●	●	●	●
BSPLINE	○	○	○	○	○	○
BTAN	○	○	○	○	○	○
C	●	●	●	●	●	●
C2	-	-	チャネル軸の名称	-	-	-
C3	-	-	-	-	-	-
C4	-	-	-	-	-	-
C5	-	-	-	-	-	-
C6	-	-	-	-	-	-
C7	-	-	-	-	-	-
CAC	●	●	●	●	●	●
CACN	●	●	●	●	●	●
CACP	●	●	●	●	●	●
CALCDAT	●	●	●	●	●	●
CALCPOSI	●	●	●	●	●	●
CALL	●	●	●	●	●	●
CALLPATH	●	●	●	●	●	●
CANCEL	●	●	●	●	●	●
CASE	●	●	●	●	●	●
CDC	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
CDOF	-	-	-	-	-	-
CDOF2	-	-	-	-	-	-
CDON	-	-	-	-	-	-
CFC	●	●	●	●	●	●
CFIN	●	●	●	●	●	●
CFINE	●	●	●	●	●	●
CFTCP	●	●	●	●	●	●
CHAN	●	●	●	●	●	●
CHANDATA	●	●	●	●	●	●
CHAR	●	●	●	●	●	●
CHF	●	●	●	●	●	●
CHKDM	●	●	●	●	●	●
CHKDNO	●	●	●	●	●	●
CHR	●	●	●	●	●	●
CIC	●	●	●	●	●	●
CIP	●	●	●	●	●	●
CLEARM	-	-	●	-	-	●
CLRINT	●	●	●	●	●	●
CMIRROR	●	●	●	●	●	●
COARSEA	●	●	●	●	●	●
COLLPAIR	-	-	-	-	-	-
COMPCAD	●	●	●	●	●	●
COMPCURV	●	●	●	●	●	●
COMPLETE	●	●	●	●	●	●
COMPOF	●	●	●	●	●	●
COMPON	●	●	●	●	●	●
COMPSURF	-	-	-	-	-	-

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
CONTDCON	●	●	●	●	●	●
CONTPRON	●	●	●	●	●	●
CORROF	●	●	●	●	●	●
CORRTRAFO	-	-	-	-	-	-
cos	●	●	●	●	●	●
COUPDEF	○	○	○	○	○	○
COUPDEL	○	○	○	○	○	○
COUPOF	○	○	○	○	○	○
COUPOFS	○	○	○	○	○	○
COUPON	○	○	○	○	○	○
COUPONC	○	○	○	○	○	○
COUPRES	○	○	○	○	○	○
CP	●	●	●	●	●	●
CPBC	○	○	○	○	○	○
CPDEF	○	○	○	○	○	○
CPDEL	○	○	○	○	○	○
CPFMOF	○	○	○	○	○	○
CPFMON	○	○	○	○	○	○
CPFMSON	○	○	○	○	○	○
CPFPOS	○	○	○	○	○	○
CPFRS	○	○	○	○	○	○
CPLA	○	○	○	○	○	○
CPLCTID	○	○	○	○	○	○
CPLDEF	○	○	○	○	○	○
CPLDEL	○	○	○	○	○	○
CPLDEN	○	○	○	○	○	○
CPLINSC	○	○	○	○	○	○

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
CPLINTR	○	○	○	○	○	○
CPLNUM	○	○	○	○	○	○
CPLOF	○	○	○	○	○	○
CPLON	○	○	○	○	○	○
CPLOUTSC	○	○	○	○	○	○
CPLOUTTR	○	○	○	○	○	○
CPLPOS	○	○	○	○	○	○
CPLSETVAL	○	○	○	○	○	○
CPMALARM	○	○	○	○	○	○
CPMBRAKE	○	○	○	○	○	○
CPMPRT	○	○	○	○	○	○
CPMRESET	○	○	○	○	○	○
CPMSTART	○	○	○	○	○	○
CPMVDI	○	○	○	○	○	○
CPOF	○	○	○	○	○	○
CPON	○	○	○	○	○	○
CPRECOF	●	●	●	●	●	●
CPRECON	●	●	●	●	●	●
CPRES	○	○	○	○	○	○
CPROT	●	●	●	●	●	●
CPROTDEF	●	●	●	●	●	●
CPSETTYPE	○	○	○	○	○	○
CPSYNCOP	○	○	○	○	○	○
CPSYNCOP2	○	○	○	○	○	○
CPSYNCOV	○	○	○	○	○	○
CPSYNFIP	○	○	○	○	○	○
CPSYNFIP2	○	○	○	○	○	○

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
CPSYNFIV	○	○	○	○	○	○
CR	●	●	●	●	●	●
CROT	●	●	●	●	●	●
CROTS	●	●	●	●	●	●
CRPL	●	●	●	●	●	●
CSCALE	●	●	●	●	●	●
CSPLINE	○	○	○	○	○	○
CT	●	●	●	●	●	●
CTAB	-	-	-	-	-	-
CTABDEF	-	-	-	-	-	-
CTABDEL	-	-	-	-	-	-
CTABEND	-	-	-	-	-	-
CTABEXISTS	-	-	-	-	-	-
CTABFNO	-	-	-	-	-	-
CTABFPOL	-	-	-	-	-	-
CTABFSEG	-	-	-	-	-	-
CTABID	-	-	-	-	-	-
CTABINV	-	-	-	-	-	-
CTABISLOCK	-	-	-	-	-	-
CTABLOCK	-	-	-	-	-	-
CTABMEMTYP	-	-	-	-	-	-
CTABMPOL	-	-	-	-	-	-
CTABMSEG	-	-	-	-	-	-
CTABNO	-	-	-	-	-	-
CTABNOMEM	-	-	-	-	-	-
CTABPERIOD	-	-	-	-	-	-
CTABPOL	-	-	-	-	-	-

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
CTABPOLID	-	-	-	-	-	-
CTABSEG	-	-	-	-	-	-
CTABSEGID	-	-	-	-	-	-
CTABSEV	-	-	-	-	-	-
CTABSSV	-	-	-	-	-	-
CTABTEP	-	-	-	-	-	-
CTABTEV	-	-	-	-	-	-
CTABTMAX	-	-	-	-	-	-
CTABTMIN	-	-	-	-	-	-
CTABTSP	-	-	-	-	-	-
CTABTSV	-	-	-	-	-	-
CTABUNLOCK	-	-	-	-	-	-
CTOL	●	●	●	●	●	●
CTRANS	●	●	●	●	●	●
CUT2D	●	●	●	●	●	●
CUT2DD	●	●	●	●	●	●
CUT2DF	●	●	●	●	●	●
CUT2DFD	●	●	●	●	●	●
CUT3DC	-	-	-	-	-	-
CUT3DCC	-	-	-	-	-	-
CUT3DCCD	-	-	-	-	-	-
CUT3DCD	-	-	-	-	-	-
CUT3DF	-	-	-	-	-	-
CUT3DFD	-	-	-	-	-	-
CUT3DFF	-	-	-	-	-	-
CUT3DFS	-	-	-	-	-	-
CUTCONOF	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
CUTCONON	●	●	●	●	●	●
CUTMOD	●	●	●	●	●	●
CUTMODK	-	-	-	-	-	-
CYCLE60	-	-	-	-	-	-
CYCLE61	-	-	-	-	-	-
CYCLE62	●	●	●	●	●	●
CYCLE63	-	-	-	-	-	-
CYCLE64	-	-	-	-	-	-
CYCLE70	-	-	-	-	-	-
CYCLE72	-	-	-	-	-	-
CYCLE76	-	-	-	-	-	-
CYCLE77	-	-	-	-	-	-
CYCLE78	-	-	-	-	-	-
CYCLE79	-	-	-	-	-	-
CYCLE81	-	-	-	-	-	-
CYCLE82	-	-	-	-	-	-
CYCLE83	-	-	-	-	-	-
CYCLE84	-	-	-	-	-	-
CYCLE85	-	-	-	-	-	-
CYCLE86	-	-	-	-	-	-
CYCLE92	-	-	-	-	-	-
CYCLE95	-	-	-	-	-	-
CYCLE98	-	-	-	-	-	-
CYCLE99	-	-	-	-	-	-
CYCLE150	-	-	-	-	-	-
CYCLE435	○	○	○	○	○	○
CYCLE495	○	○	○	○	○	○

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
CYCLE750	●	●	●	●	●	●
CYCLE751	●	●	●	●	●	●
CYCLE752	●	●	●	●	●	●
CYCLE753	●	●	●	●	●	●
CYCLE754	●	●	●	●	●	●
CYCLE755	●	●	●	●	●	●
CYCLE756	●	●	●	●	●	●
CYCLE757	●	●	●	●	●	●
CYCLE758	●	●	●	●	●	●
CYCLE759	●	●	●	●	●	●
CYCLE800	○	○	○	○	○	○
CYCLE801	-	-	-	-	-	-
CYCLE802	-	-	-	-	-	-
CYCLE830	-	-	-	-	-	-
CYCLE832	●	●	●	●	●	●
CYCLE840	-	-	-	-	-	-
CYCLE899	-	-	-	-	-	-
CYCLE930	-	-	-	-	-	-
CYCLE940	-	-	-	-	-	-
CYCLE951	-	-	-	-	-	-
CYCLE952	-	-	-	-	-	-
CYCLE961	-	-	-	-	-	-
CYCLE971	-	-	-	-	-	-
CYCLE973	-	-	-	-	-	-
CYCLE974	-	-	-	-	-	-
CYCLE976	-	-	-	-	-	-
CYCLE977	-	-	-	-	-	-

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
CYCLE978	-	-	-	-	-	-
CYCLE979	-	-	-	-	-	-
CYCLE982	-	-	-	-	-	-
CYCLE994	-	-	-	-	-	-
CYCLE995	-	-	-	-	-	-
CYCLE996	-	-	-	-	-	-
CYCLE997	-	-	-	-	-	-
CYCLE998	-	-	-	-	-	-
CYCLE4071	●	●	●	●	●	●
CYCLE4072	●	●	●	●	●	●
CYCLE4073	●	●	●	●	●	●
CYCLE4074	●	●	●	●	●	●
CYCLE4075	●	●	●	●	●	●
CYCLE4077	●	●	●	●	●	●
CYCLE4078	●	●	●	●	●	●
CYCLE4079	●	●	●	●	●	●

命令 D ... F

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
D	●	●	●	●	●	●
D0	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
DAC	●	●	●	●	●	●
DC	●	●	●	●	●	●
DCI	●	●	●	●	●	●
DCM	●	●	●	●	●	●
DCU	●	●	●	●	●	●
DEF	●	●	●	●	●	●
DEFINE	●	●	●	●	●	●
DEFAULT	●	●	●	●	●	●
DELAYFSTON	●	●	●	●	●	●
DELAYFSTOF	●	●	●	●	●	●
DELDL	●	●	●	●	●	●
DELDTG	●	●	●	●	●	●
DELETE	●	●	●	●	●	●
DELMLOWNER	●	●	●	●	●	●
DELMLRES	●	●	●	●	●	●
DELMT	-	-	-	-	-	-
DELOBJ	-	-	-	-	-	-
DELT	●	●	●	●	●	●
DELTC	●	●	●	●	●	●
DELTOOLENV	●	●	●	●	●	●
DIACYCOFA	●	●	●	●	●	●
DIAM90	●	●	●	●	●	●
DIAM90A	●	●	●	●	●	●
DIAMCHAN	●	●	●	●	●	●
DIAMCHANA	●	●	●	●	●	●
DIAMCYCOF	●	●	●	●	●	●
DIAMOF	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
DIAMOFA	●	●	●	●	●	●
DIAMON	●	●	●	●	●	●
DIAMONA	●	●	●	●	●	●
DIC	●	●	●	●	●	●
DILF	●	●	●	●	●	●
DISABLE	●	●	●	●	●	●
DISC	●	●	●	●	●	●
DISCL	●	●	●	●	●	●
DISPLOF	●	●	●	●	●	●
DISPLON	●	●	●	●	●	●
DISPR	●	●	●	●	●	●
DISR	●	●	●	●	●	●
DISRP	●	●	●	●	●	●
DITE	●	●	●	●	●	●
DITS	●	●	●	●	●	●
DIV	●	●	●	●	●	●
DI	-	-	-	-	-	-
DO	●	●	●	●	●	●
DRFOF	●	●	●	●	●	●
DRIVE	●	●	●	●	●	●
DRIVEA	●	●	●	●	●	●
DYNFINISH	●	●	●	●	●	●
DYNNORM	●	●	●	●	●	●
DYNPOS	●	●	●	●	●	●
DYNROUGH	●	●	●	●	●	●
DYNSEMIFIN	●	●	●	●	●	●
DZERO	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
EAUTO	○	○	○	○	○	○
EGDEF	○	○	○	○	○	○
EGDEL	○	○	○	○	○	○
EGOFC	○	○	○	○	○	○
EGOFS	○	○	○	○	○	○
EGON	○	○	○	○	○	○
EGONSYN	○	○	○	○	○	○
EGONSYNE	○	○	○	○	○	○
ELSE	●	●	●	●	●	●
ENABLE	●	●	●	●	●	●
ENAT	○	○	○	○	○	○
ENDFOR	●	●	●	●	●	●
ENDIF	●	●	●	●	●	●
ENDLABEL	●	●	●	●	●	●
ENDLOOP	●	●	●	●	●	●
ENDPROC	●	●	●	●	●	●
ENDWHILE	●	●	●	●	●	●
ESRR	○	○	○	○	○	○
ESRS	○	○	○	○	○	○
ETAN	○	○	○	○	○	○
EVERY	●	●	●	●	●	●
EX	●	●	●	●	●	●
EXECSTRING	●	●	●	●	●	●
EXECTAB	●	●	●	●	●	●
EXECUTE	●	●	●	●	●	●
EXP	●	●	●	●	●	●
EXTCALL	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
EXTCLOSE	●	●	●	●	●	●
EXTERN	●	●	●	●	●	●
EXTOPEN	●	●	●	●	●	●
F	●	●	●	●	●	●
FA	●	●	●	●	●	●
FAD	●	●	●	●	●	●
FALSE	●	●	●	●	●	●
FB	●	●	●	●	●	●
FCTDEF	●	●	●	●	●	●
FCUB	●	●	●	●	●	●
FD	●	●	●	●	●	●
FDA	●	●	●	●	●	●
FENDNORM	●	●	●	●	●	●
FFWOF	●	●	●	●	●	●
FFWON	●	●	●	●	●	●
FGREF	●	●	●	●	●	●
FGROUP	●	●	●	●	●	●
FI	●	●	●	●	●	●
FIFOCTRL	●	●	●	●	●	●
FILEDATE	●	●	●	●	●	●
FILEINFO	●	●	●	●	●	●
FILESIZE	●	●	●	●	●	●
FILESTAT	●	●	●	●	●	●
FILETIME	●	●	●	●	●	●
FINEA	●	●	●	●	●	●
FL	●	●	●	●	●	●
FLIN	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
FMA	●	●	●	●	●	●
FNORM	●	●	●	●	●	●
FOCOF	○	○	○	○	○	○
FOCON	○	○	○	○	○	○
FOR	●	●	●	●	●	●
FP	●	●	●	●	●	●
FPO	-	-	-	-	-	-
FPR	●	●	●	●	●	●
FPRAOF	●	●	●	●	●	●
FPRAON	●	●	●	●	●	●
FRAME	●	●	●	●	●	●
FRC	●	●	●	●	●	●
FRCM	●	●	●	●	●	●
FROM	●	●	●	●	●	●
FTOC	●	●	●	●	●	●
FTOCOF	●	●	●	●	●	●
FTOCON	●	●	●	●	●	●
FXS	●	●	●	●	●	●
FXST	●	●	●	●	●	●
FXSW	●	●	●	●	●	●
FZ	●	●	●	●	●	●

命令 G ... L

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
G0	●	●	●	●	●	●
G1	●	●	●	●	●	●
G2	●	●	●	●	●	●
G3	●	●	●	●	●	●
G4	●	●	●	●	●	●
G5	●	●	●	●	●	●
G7	●	●	●	●	●	●
G9	●	●	●	●	●	●
G17	●	●	●	●	●	●
G18	●	●	●	●	●	●
G19	●	●	●	●	●	●
G25	●	●	●	●	●	●
G26	●	●	●	●	●	●
G33	●	●	●	●	●	●
G34	●	●	●	●	●	●
G35	●	●	●	●	●	●
G40	●	●	●	●	●	●
G41	●	●	●	●	●	●
G42	●	●	●	●	●	●
G53	●	●	●	●	●	●
G54	●	●	●	●	●	●
G55	●	●	●	●	●	●
G56	●	●	●	●	●	●
G57	●	●	●	●	●	●
G58	→ G505					

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
G59	→ G506					
G60	●	●	●	●	●	●
G62	●	●	●	●	●	●
G63	●	●	●	●	●	●
G64	●	●	●	●	●	●
G70	●	●	●	●	●	●
G71	●	●	●	●	●	●
G74	●	●	●	●	●	●
G75	●	●	●	●	●	●
G90	●	●	●	●	●	●
G91	●	●	●	●	●	●
G93	●	●	●	●	●	●
G94	●	●	●	●	●	●
G95	●	●	●	●	●	●
G96	●	●	●	●	●	●
G97	●	●	●	●	●	●
G110	●	●	●	●	●	●
G111	●	●	●	●	●	●
G112	●	●	●	●	●	●
G140	●	●	●	●	●	●
G141	●	●	●	●	●	●
G142	●	●	●	●	●	●
G143	●	●	●	●	●	●
G147	●	●	●	●	●	●
G148	●	●	●	●	●	●
G153	●	●	●	●	●	●
G247	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
G248	●	●	●	●	●	●
G290	●	●	●	●	●	●
G291	-	-	-	-	-	-
G331	●	●	●	●	●	●
G332	●	●	●	●	●	●
G335	●	●	●	●	●	●
G336	●	●	●	●	●	●
G340	●	●	●	●	●	●
G341	●	●	●	●	●	●
G347	●	●	●	●	●	●
G348	●	●	●	●	●	●
G450	●	●	●	●	●	●
G451	●	●	●	●	●	●
G460	●	●	●	●	●	●
G461	●	●	●	●	●	●
G462	●	●	●	●	●	●
G500	●	●	●	●	●	●
G505 ～ G599	●	●	●	●	●	●
G601	●	●	●	●	●	●
G602	●	●	●	●	●	●
G603	●	●	●	●	●	●
G621	●	●	●	●	●	●
G641	●	●	●	●	●	●
G642	●	●	●	●	●	●
G643	●	●	●	●	●	●
G644	●	●	●	●	●	●
G645	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
G700	●	●	●	●	●	●
G710	●	●	●	●	●	●
G810 ～ G819	-	-	-	-	-	-
G820 ～ G829	-	-	-	-	-	-
G931	●	●	●	●	●	●
G942	●	●	●	●	●	●
G952	●	●	●	●	●	●
G961	●	●	●	●	●	●
G962	●	●	●	●	●	●
G971	●	●	●	●	●	●
G972	●	●	●	●	●	●
G973	●	●	●	●	●	●
GEOAX	●	●	●	●	●	●
GET	●	●	●	●	●	●
GETACTT	●	●	●	●	●	●
GETACTTD	●	●	●	●	●	●
GETD	-	-	●	-	-	●
GETDNO	●	●	●	●	●	●
GETEXET	●	●	●	●	●	●
GETFREELOC	●	●	●	●	●	●
GETSELT	●	●	●	●	●	●
GETT	●	●	●	●	●	●
GETTCOR	●	●	●	●	●	●
GETTENV	●	●	●	●	●	●
GETVARAP	●	●	●	●	●	●
GETVARDFT	●	●	●	●	●	●
GETVARLIM	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
GETVARPHU	●	●	●	●	●	●
GETVARTYP	●	●	●	●	●	●
GFRAME0 ... GFRAME100	< 50	< 100	< 100	< 50	< 100	< 100
GOTO	●	●	●	●	●	●
GOTOB	●	●	●	●	●	●
GOTOC	●	●	●	●	●	●
GOTOF	●	●	●	●	●	●
GOTOS	●	●	●	●	●	●
BP	●	●	●	●	●	●
GWPSOF	●	●	●	●	●	●
GROUP_ADDEND	●	●	●	●	●	●
GROUP_BEGIN	●	●	●	●	●	●
GROUP_END	●	●	●	●	●	●
GWPSON	●	●	●	●	●	●
H...	●	●	●	●	●	●
HOLES1	-	-	-	-	-	-
HOLES2	-	-	-	-	-	-
I	●	●	●	●	●	●
I1	●	●	●	●	●	●
IC	●	●	●	●	●	●
ICYCOF	●	●	●	●	●	●
ICYCON	●	●	●	●	●	●
ID	●	●	●	●	●	●
IDS	●	●	●	●	●	●
IF	●	●	●	●	●	●
INDEX	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
INIPO	●	●	●	●	●	●
INIRE	●	●	●	●	●	●
INICF	●	●	●	●	●	●
INIT	-	-	●	-	-	●
INITIAL						
INT	●	●	●	●	●	●
INTERSEC	●	●	●	●	●	●
INVCCW	-	-	-	-	-	-
INVCW	-	-	-	-	-	-
INVFRAME	●	●	●	●	●	●
IP	●	●	●	●	●	●
IPOBRKA	●	●	●	●	●	●
IPOENDA	●	●	●	●	●	●
IPTRLOCK	●	●	●	●	●	●
IPTRUNLOCK	●	●	●	●	●	●
IR	●	●	●	●	●	●
ISAXIS	●	●	●	●	●	●
ISD	-	-	-	-	-	-
ISFILE	●	●	●	●	●	●
ISNUMBER	●	●	●	●	●	●
ISOCALL	-	-	-	-	-	-
ISVAR	●	●	●	●	●	●
J	●	●	●	●	●	●
J1	●	●	●	●	●	●
JERKA	●	●	●	●	●	●
JERKLIM	●	●	●	●	●	●
JERKLIMA	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
JR	●	●	●	●	●	●
K	●	●	●	●	●	●
K1	●	●	●	●	●	●
KONT	●	●	●	●	●	●
KONTC	●	●	●	●	●	●
KONTT	●	●	●	●	●	●
KR	●	●	●	●	●	●
L	●	●	●	●	●	●
LEAD 工具オリエンテーション 旋回多項式	- -	- -	- -	- -	- -	- -
LEADOF	-	-	-	-	-	-
LEADON	-	-	-	-	-	-
LENTOAX	●	●	●	●	●	●
LFOF	●	●	●	●	●	●
LFON	●	●	●	●	●	●
LFPOS	●	●	●	●	●	●
LFTXT	●	●	●	●	●	●
LFWP	●	●	●	●	●	●
LIFTFAST	●	●	●	●	●	●
LIMS	●	●	●	●	●	●
LLI	●	●	●	●	●	●
LN	●	●	●	●	●	●
LOCK	●	●	●	●	●	●
LONGHOLE	-	-	-	-	-	-
LOOP	●	●	●	●	●	●

命令 M ... R

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
M0	●	●	●	●	●	●
M1	●	●	●	●	●	●
M2	●	●	●	●	●	●
M3	●	●	●	●	●	●
M4	●	●	●	●	●	●
M5	●	●	●	●	●	●
M6	●	●	●	●	●	●
M17	●	●	●	●	●	●
M19	●	●	●	●	●	●
M30	●	●	●	●	●	●
M40	●	●	●	●	●	●
M41 ... M45	●	●	●	●	●	●
M70	●	●	●	●	●	●
MASLDEF	-	-	-	-	-	-
MASLDEL	-	-	-	-	-	-
MASLOF	-	-	-	-	-	-
MASLOFS	-	-	-	-	-	-
MASLON	-	-	-	-	-	-
MATCH	●	●	●	●	●	●
MAXVAL	●	●	●	●	●	●
MCALL	●	●	●	●	●	●
MEAC	-	○	○	-	○	○
MEAFRAME	●	●	●	●	●	●
MEAS	●	●	●	●	●	●
MEASA	-	○	○	-	○	○

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
MEASURE	●	●	●	●	●	●
MEAW	●	●	●	●	●	●
MEAWA	-	○	○	-	○	○
Mi	●	●	●	●	●	●
MINDEX	●	●	●	●	●	●
MINVAL	●	●	●	●	●	●
MIRROR	●	●	●	●	●	●
MMC	●	●	●	●	●	●
MOD	●	●	●	●	●	●
MODAXVAL	●	●	●	●	●	●
MOV	●	●	●	●	●	●
MOVT	●	●	●	●	●	●
MSG	●	●	●	●	●	●
MVTOOL	●	●	●	●	●	●
N	●	●	●	●	●	●
NAMETOINT	●	●	●	●	●	●
NCK	●	●	●	●	●	●
NEWCONF	●	●	●	●	●	●
NEWMT	●	●	●	●	●	●
NEWT	-	-	-	-	-	-
NORM	●	●	●	●	●	●
NOT	●	●	●	●	●	●
NPROT	●	●	●	●	●	●
NPROTDEF	●	●	●	●	●	●
NUMBER	●	●	●	●	●	●
OEMIPO1	-	-	-	-	-	-
OEMIPO2	-	-	-	-	-	-

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
OF	●	●	●	●	●	●
OFFN	●	●	●	●	●	●
OMA1	-	-	-	-	-	-
OMA2	-	-	-	-	-	-
OMA3	-	-	-	-	-	-
OMA4	-	-	-	-	-	-
OMA5	-	-	-	-	-	-
OR	●	●	●	●	●	●
ORIAXES	-	-	-	-	-	-
ORIAXPOS	-	-	-	-	-	-
ORIC	-	-	-	-	-	-
ORICONCCW	-	-	-	-	-	-
ORICONCW	-	-	-	-	-	-
ORICONIO	-	-	-	-	-	-
ORICONTO	-	-	-	-	-	-
ORICURVE	-	-	-	-	-	-
ORID	-	-	-	-	-	-
ORIEULER	-	-	-	-	-	-
ORIMKS	-	-	-	-	-	-
ORIPATH	-	-	-	-	-	-
ORIPATHS	-	-	-	-	-	-
ORIPLANE	-	-	-	-	-	-
ORIRESET	-	-	-	-	-	-
ORIROTA	-	-	-	-	-	-
ORIROTC	-	-	-	-	-	-
ORIROTR	-	-	-	-	-	-
ORIROTT	-	-	-	-	-	-

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
ORIRPY	-	-	-	-	-	-
ORIRPY2	-	-	-	-	-	-
ORIS	-	-	-	-	-	-
ORISOF	-	-	-	-	-	-
ORISOLH	-	-	-	-	-	-
ORISON	-	-	-	-	-	-
ORIVECT	-	-	-	-	-	-
ORIVIRT1	-	-	-	-	-	-
ORIVIRT2	-	-	-	-	-	-
ORIWKS	-	-	-	-	-	-
OS	○	○	○	○	○	○
OSB	○	○	○	○	○	○
OSC	-	-	-	-	-	-
OSCILL	○	○	○	○	○	○
OSCTRL	○	○	○	○	○	○
OSD	-	-	-	-	-	-
OSE	○	○	○	○	○	○
OSNSC	●	●	●	●	●	●
OSOF	-	-	-	-	-	-
OSP1	●	●	●	●	●	●
OSP2	●	●	●	●	●	●
OSS	-	-	-	-	-	-
OSSE	-	-	-	-	-	-
OST	-	-	-	-	-	-
OST1	●	●	●	●	●	●
OST2	●	●	●	●	●	●
OTOL	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
OVR	●	●	●	●	●	●
OVRA	●	●	●	●	●	●
OVRRAP	●	●	●	●	●	●
P	●	●	●	●	●	●
PAROT	●	●	●	●	●	●
PAROTOF	●	●	●	●	●	●
PCALL	●	●	●	●	●	●
PDELAYOF	-	-	-	-	-	-
PDELAYON	-	-	-	-	-	-
PHI	-	-	-	-	-	-
PHU	●	●	●	●	●	●
PL	-	-	-	-	-	-
PM	●	●	●	●	●	●
PO	-	-	-	-	-	-
POCKET3	-	-	-	-	-	-
POCKET4	-	-	-	-	-	-
POLF	●	●	●	●	●	●
POLFA	●	●	●	●	●	●
POLFMASK	●	●	●	●	●	●
POLFMLIN	●	●	●	●	●	●
POLY	-	-	-	-	-	-
POLYPATH	-	-	-	-	-	-
PON	-	-	-	-	-	-
PONS	-	-	-	-	-	-
POS	●	●	●	●	●	●
POSA	●	●	●	●	●	●
POSM	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
POSMT	-	-	-	-	-	-
POSP	●	●	●	●	●	●
POSRANGE	●	●	●	●	●	●
POT	●	●	●	●	●	●
PR	●	●	●	●	●	●
PREPRO	●	●	●	●	●	●
PRESETON	●	●	●	●	●	●
PRESETONS	●	●	●	●	●	●
PRIO	●	●	●	●	●	●
PRLOC	●	●	●	●	●	●
PROC	●	●	●	●	●	●
PROTA	●	●	●	●	●	●
PROTD	●	●	●	●	●	●
PROTS	●	●	●	●	●	●
psi	-	-	-	-	-	-
PTP	●	●	●	●	●	●
PTPG0	●	●	●	●	●	●
PTPWOC	●	●	●	●	●	●
PUNCHACC	-	-	-	-	-	-
PUTFTOC	●	●	●	●	●	●
PUTFTOCF	●	●	●	●	●	●
PW	○	○	○	○	○	○
QU	●	●	●	●	●	●
R...	●	●	●	●	●	●
RAC	●	●	●	●	●	●
RDISABLE	●	●	●	●	●	●
READ	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
REAL	●	●	●	●	●	●
RELEASE	●	●	●	●	●	●
REP	●	●	●	●	●	●
REPEAT	●	●	●	●	●	●
REPEATB	●	●	●	●	●	●
REPOSA	●	●	●	●	●	●
REPOSH	●	●	●	●	●	●
REPOSHA	●	●	●	●	●	●
REPOSL	●	●	●	●	●	●
REPOSQ	●	●	●	●	●	●
REPOSQA	●	●	●	●	●	●
RESETMON	●	●	●	●	●	●
RET	●	●	●	●	●	●
RETB	●	●	●	●	●	●
RIC	●	●	●	●	●	●
RINDEX	●	●	●	●	●	●
RMB	●	●	●	●	●	●
RME	●	●	●	●	●	●
RMI	●	●	●	●	●	●
RMN	●	●	●	●	●	●
RND	●	●	●	●	●	●
RNDM	●	●	●	●	●	●
ROT	●	●	●	●	●	●
ROTS	●	●	●	●	●	●
ROUND	●	●	●	●	●	●
ROUNDUP	●	●	●	●	●	●
RP	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
RPL	●	●	●	●	●	●
RT	●	●	●	●	●	●
RTLIOF	●	●	●	●	●	●
RTLION	●	●	●	●	●	●

命令 S ... Z

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
S	●	●	●	●	●	●
SAVE	●	●	●	●	●	●
SBLOF	●	●	●	●	●	●
SBLON	●	●	●	●	●	●
SC	●	●	●	●	●	●
SCALE	●	●	●	●	●	●
SCC	●	●	●	●	●	●
SCPARA	●	●	●	●	●	●
SD	○	○	○	○	○	○
SET	●	●	●	●	●	●
SETAL	●	●	●	●	●	●
SETDNO	●	●	●	●	●	●
SETINT	●	●	●	●	●	●
SETM	-	-	●	-	-	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
SETMS	●	●	●	●	●	●
SETMS(n)	●	●	●	●	●	●
SETMTH	●	●	●	●	●	●
SETPIECE	●	●	●	●	●	●
SETTA	●	●	●	●	●	●
SETTCOR	●	●	●	●	●	●
SETTIA	●	●	●	●	●	●
SF	●	●	●	●	●	●
sin	●	●	●	●	●	●
SIRELAY	-	-	-	-	-	-
SIRELIN	-	-	-	-	-	-
SIRELOUT	-	-	-	-	-	-
SIRELTIME	-	-	-	-	-	-
SLOT1	-	-	-	-	-	-
SLOT2	-	-	-	-	-	-
SOFT	●	●	●	●	●	●
SOFTA	●	●	●	●	●	●
SON	-	-	-	-	-	-
SONS	-	-	-	-	-	-
SPATH	●	●	●	●	●	●
SPCOF	●	●	●	●	●	●
SPCON	●	●	●	●	●	●
SPI	●	●	●	●	●	●
SPIF1	-	-	-	-	-	-
SPIF2	-	-	-	-	-	-
SPLINEPATH	○	○	○	○	○	○
SPN; ＳＰＮ	-	-	-	-	-	-

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
SPOF	-	-	-	-	-	-
SPOS	●	●	●	●	●	●
SPOSA	●	●	●	●	●	●
SPP	-	-	-	-	-	-
SPRINT	●	●	●	●	●	●
SQRT	●	●	●	●	●	●
SR	●	●	●	●	●	●
SRA	●	●	●	●	●	●
ST	●	●	●	●	●	●
STA	●	●	●	●	●	●
START	-	-	●	-	-	●
STARTFIFO	●	●	●	●	●	●
STAT	●	●	●	●	●	●
STOLF	●	●	●	●	●	●
STOPFIFO	●	●	●	●	●	●
STOPRE	●	●	●	●	●	●
STOPREOF	●	●	●	●	●	●
STRING	●	●	●	●	●	●
STRINGFELD	●	●	●	●	●	●
STRINGIS	●	●	●	●	●	●
STRLEN	●	●	●	●	●	●
SUBSTR	●	●	●	●	●	●
SUPA	●	●	●	●	●	●
SVC	●	●	●	●	●	●
SYNFCT	●	●	●	●	●	●
SYNR	●	●	●	●	●	●
SYNRW	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
SYNW	●	●	●	●	●	●
T	●	●	●	●	●	●
TAN	●	●	●	●	●	●
TANG	○	○	○	○	○	○
TANGDEL	○	○	○	○	○	○
TANGOF	○	○	○	○	○	○
TANGON	○	○	○	○	○	○
TCA (828D: _TCA)	●	●	●	●	●	●
TCARR	●	●	●	●	●	●
TCI	●	●	●	●	●	●
TCOABS	●	●	●	●	●	●
TCOFR	●	●	●	●	●	●
TCOFRX	●	●	●	●	●	●
TCOFRY	●	●	●	●	●	●
TCOFRZ	●	●	●	●	●	●
THETA	-	-	-	-	-	-
TILT	-	-	-	-	-	-
TLIFT	○	○	○	○	○	○
TML	●	●	●	●	●	●
TMOF	●	●	●	●	●	●
TMON	●	●	●	●	●	●
TO	●	●	●	●	●	●
TOFF	●	●	●	●	●	●
TOFFL	●	●	●	●	●	●
TOFFOF	●	●	●	●	●	●
TOFFON	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
TOFFR	●	●	●	●	●	●
TOFRAME	●	●	●	●	●	●
TOFRAMEX	●	●	●	●	●	●
TOFRAMEY	●	●	●	●	●	●
TOFRAMEZ	●	●	●	●	●	●
TOLOWER	●	●	●	●	●	●
TOOLENV	●	●	●	●	●	●
TOOLGNT	●	●	●	●	●	●
TOOLGT	●	●	●	●	●	●
TOROT	●	●	●	●	●	●
TOROTOF	●	●	●	●	●	●
TOROTX	●	●	●	●	●	●
TOROTY	●	●	●	●	●	●
TOROTZ	●	●	●	●	●	●
TOUPPER	●	●	●	●	●	●
TOWBCS	●	●	●	●	●	●
TOWKCS	●	●	●	●	●	●
TOWMCS	●	●	●	●	●	●
TOWSTD	●	●	●	●	●	●
TOWTCS	●	●	●	●	●	●
TOWWCS	●	●	●	●	●	●
TR	●	●	●	●	●	●
TRAANG	○	○	○	-	-	-
TRACON	○	○	○	-	-	-
TRACYL	○	○	○	○	○	○
TRAFOOF	●	●	●	●	●	●
TRAFOON	-	-	-	-	-	-

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
TRAILOF	●	●	●	●	●	●
TRAILON	●	●	●	●	●	●
TRANS	●	●	●	●	●	●
TRANSMIT	○	○	○	○	○	○
TRAORI	-	-	-	-	-	-
TRUE	●	●	●	●	●	●
TRUNC	●	●	●	●	●	●
TU	●	●	●	●	●	●
TURN	●	●	●	●	●	●
ULI	●	●	●	●	●	●
UNLOCK	●	●	●	●	●	●
UNTIL	●	●	●	●	●	●
UPATH	●	●	●	●	●	●
var	●	●	●	●	●	●
VELOLIM	●	●	●	●	●	●
VELOLIMA	●	●	●	●	●	●
WAITC	●	●	●	●	●	●
WAITE	-	-	●	-	-	●
WAITENC	●	●	●	●	●	●
WAITM	-	-	●	-	-	●
WAITMC	-	-	●	-	-	●
WAITP	●	●	●	●	●	●
WAITS	●	●	●	●	●	●
WALCS0	●	●	●	●	●	●
WALCS1	●	●	●	●	●	●
WALCS2	●	●	●	●	●	●
WALCS3	●	●	●	●	●	●

命令	SINUMERIK 828D					
● 標準 ○ オプション - なし	SW24x(5) CNC-SW G-Tech エクスポート (gce42)	SW26x(3) CNC-SW G-Tech エクスポート (gce62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gce82)	SW24x(5) CNC-SW G-Tech エクスポート (gse42)	SW26x(3) CNC-SW G-Tech エクスポート (gse62)	SW28x(1) CNC-SW G-Tech 高度なエクスポート (gse82)
WALCS4	●	●	●	●	●	●
WALCS5	●	●	●	●	●	●
WALCS6	●	●	●	●	●	●
WALCS7	●	●	●	●	●	●
WALCS8	●	●	●	●	●	●
WALCS9	●	●	●	●	●	●
WALCS10	●	●	●	●	●	●
WALIMOF	●	●	●	●	●	●
WALIMON	●	●	●	●	●	●
WHEN	●	●	●	●	●	●
WHENEVER	●	●	●	●	●	●
WHILE	●	●	●	●	●	●
WORKPIECE	●	●	●	●	●	●
WRITE	●	●	●	●	●	●
WRTPR	●	●	●	●	●	●
X	●	●	●	●	●	●
XOR	●	●	●	●	●	●
Y	●	●	●	●	●	●
Z	●	●	●	●	●	●

21.3 HMI で現在設定されている言語

下の表に、操作画面で使用できるすべての言語を記載します。

現在設定されている言語を、以下のシステム変数を使用してパートプログラムとシンクロナイズドアクションで問い合わせることができます。

$AN_LANGUAGE_ON_HMI = <value>

<value>	言語	言語コード
1	ドイツ語(ドイツ)	GER
2	フランス語	FRA
3	英語(英国)	ENG
4	スペイン語	ESP
6	イタリア語	ITA
7	オランダ語	NLD
8	中国語(簡体)	CHS
9	スウェーデン語	SVE
18	ハンガリー語	HUN
19	フィンランド語	FIN
28	チェコ語	CSY
50	ポルトガル語(ブラジル)	PTB
53	ポーランド語	PLK
55	デンマーク語	DAN
57	ロシア語	RUS
68	スロバキア語	SKY
72	ルーマニア語	ROM
80	中国語(繁体)	CHT
85	韓国語	KOR
87	日本語	JPN
89	トルコ語	TRK

注記

$AN_LANGUAGE_ON_HMI は以下の場合に更新されます。

- システムの起動後
- NC および/または PLC のリセット後
- M2N の範囲内で別の NC への切り替え後
- HMI での言語の切り替え後

付録 A

A.1 略語の一覧

A	
O	出力(Output)
ADI4	4 軸アナログドライブインタフェース(Analog Drive Interface for 4 axes)
AC	適応制御(Adaptive Control)
ALM	アクティブラインモジュール(Active Line Module)
ARM	回転型誘導電動機
AS	オートメーションシステム(Automation System)
ASCII	American Standard Code for Information Interchange:情報交換のための米国標準コード
ASIC	Application-Specific Integrated Circuit:ユーザー専用の特定の用途のための集積回路
ASUB	非同期サブプログラム(Asynchronous SUBprogram)
AUXFU	補助機能(AUXiliary FUnction)
STL	ステートメントリスト(Statement List)
UP	ユーザープログラム(User Program)

B	
OP	動作モード
BAG	モードグループ
BCD	2 進化 10 進数(Binary Coded Decimals):バイナリコードで表現された 10 進数
BERO	近接スイッチの商標
BI	バイネクタ入力(Binector Input)
BICO	バイネクタコネクタ(BInector COnnector)
BIN	バイナリファイル(BINary files)
BIOS	基本的な入出力をおこなうシステム(Basic Input Output System)
BCS	基本座標系(Basic Coordinate System)

B	
BO	バイネクタ出力(Binector Output)
OPI	操作パネルインタフェース(Operator Panel Interface)

C	
CAD	コンピュータ支援設計(Computer-Aided Design)
CAM	コンピュータ支援製造(Computer-Aided Manufacturing)
CC	コンパイルサイクル(Compile Cycle)
CEC	交差誤差補正(Cross Error Compensation)
CI	コネクタ入力(Connector Input)
CF カード	コンパクトフラッシュカード(Compact Flash Card)
CNC	コンピュータによる数値制御装置(Computerized Numerical Control)
CO	コネクタ出力(Connector Output)
CoL	ライセンス証明書(Certificate of License)
COM	通信(COMmunication)
CPA	コンパイラプロジェクトデータ(Compiler Projecting Data):コンパイラの設定データ
CRT	ブラウン管(Cathode Ray Tube):受像管
CSB	Central Service Board:PLC モジュール
CU	コントロールユニット(Control Unit)
CP	通信プロセッサ(Communication Processor)
CPU	中央演算処理装置(Central Processing Unit)
CR	キャリッジリターン(Carriage Return)
CTS	Clear To Send:シリアルデータインタフェースへの送信準備完了信号
CUTCOM	工具径補正(CUTter radius COMpensation)

D	
DAC	ディジタル/アナログコンバータ(Digital-to-Analog Converter)
DB	データブロック(Data Block)(PLC)
DBB	データブロックバイト(Data Block Byte)(PLC)
DBD	データブロックダブルワード(Data Block Double word)(PLC)
DBW	データブロックワード(Data Block Word)(PLC)

D	
DBX	データブロックビット(Data block bit)(PLC)
DDE	ダイナミックデータ交信(Dynamic Data Exchange)
DDS	ドライブデータセット(Drive Data Set)
DIN	ドイツ工業規格(Deutsche Industrie Norm)
DIO	データ入力/出力(Data Input/Output):データ送受信表示
DIR	ディレクトリ(DIRectory)
DLL	ダイナミックリンクライブラリ(Dynamic Link Library)
DO	ドライブオブジェクト(Drive Object)
DPM	デュアルポートメモリ(Dual Port Memory)
DPR	デュアルポート RAM(Dual Port RAM)
DRAM	ダイナミックメモリ(非保持)
DRF	差動レゾルバ機能(Differential Resolver Function)(手動パルス発生器)
DRIVE-CLiQ	IQ によるドライブコンポーネントリンク(Drive Component Link with IQ)
DRY	ドライラン(DRY run):ドライラン送り速度
DSB	デコードシングルブロック(Decoding Single Block)
DSC	ダイナミックサーボ制御(Dynamic Servo Control) / ダイナミックスティフネスコントロール
DW	データワード(Data Word)
DWORD	ダブルワード(Double Word)(現在は 32 ビット)

E	
I	入力(Input)
EES	外部記憶からの実行(Execution from External Storage)
I/O	入力/出力(Input/Output)
ENC	エンコーダ(ENCorder):フィードバックエンコーダ
EFP	小型 I/O モジュール(PLC I/O モジュール)
ESD	静電気により破損するおそれのある部品(Electrostatic Sensitive Devices)
EMC	電磁両立性(ElectroMagnetic Compatibility)
EN	欧州統一規格
ENC	エンコーダ(ENCorder):フィードバックエンコーダ
EnDat	エンコーダインタフェース(Encoder interface)

E	
EPROM	消去可能なプログラマブル ROM(Erasable Programmable Read Only Memory)
ePS Network Services	インターネットベースのリモート機械保守のサービス
EQN	1 回転当たり 2048 のサイン信号を出力する絶対値エンコーダの名称
ES	エンジニアリングシステム(Engineering System)
ESR	停止延長と退避(Extended Stop and Retract)
ETC	ETC キー「>」: 同じメニュー内でのソフトキーバーの拡張

F	
FB	ファンクションブロック(Function Block)(PLC)
FC	ファンクションコール(Function Call):ファンクションブロック(PLC)
FEPROM	フラッシュ EPROM(Flash EPROM):読み取りと書き込みメモリ
FIFO	First In First Out:アドレス指定なしで動作し、データが格納された順序で読み取られるメモリ
FIPO	ファインインタポレーションの補間器(Fine InterPOlator)
FPU	浮動小数点演算ユニット(Floating Point Unit)
CRC	工具径補正(Cutter Radius Compensation)
FST	送り停止(Feed STop)
FBD	ファンクションブロック図(Function Block Diagram)(PLC プログラミング方式)
FW	ファームウェア(FirmWare)

G	
GC	グローバル制御(Global Control) (PROFIBUS:ブロードキャストメッセージ)
GDIR	グローバルパートプログラムメモリ
GEO	ジオメトリ(GEOmetry)、例: ジオメトリ軸
GIA	ギヤ補間データ(Gear Interpolation dAta)
GND	基準電位
BP	基本プログラム(Basic Program)(PLC)
GS	ギヤ選択(Gear Stage)

G	
GSD	PROFIBUS スレーブを記述するためのデバイスマスタファイル
GSDML	GSD ファイルを作成するための XML ベース記述言語(Generic Station Description Markup Language)
GUD	グローバルユーザーデータ(Global User Data)

H	
HEX	16 進数を示す略語
AuxF	補助機能(Auxiliary Function)
HLA	油圧リニアドライブ(Hydraulic Linear Drive)
HMI	マンマシンインタフェース(Human Machine Interface):SINUMERIK 操作画面
MSD	メイン主軸ドライブ(Main Spindle Drive)
HW	ハードウェア(HardWare)

I	
IBN	セットアップ
ICA	補間型補正(Interpolatory Compensation)
IM	インタフェースモジュール: 内部接続モジュール(Interface Module)
IMR	データ受信用のインタフェースモジュール(Interface Module Receive)
IMS	データ送信用のインタフェースモジュール(Interface Module Send)
INC	インクレメント(INCrement)
INI	初期化データ(INItializing Data)
IPO	補間器(InterPOlator)
ISA	コンピュータのバス規格(Industry Standard Architecture)
ISO	国際標準化機構(International Standardization Organization)

J	
JOG	ジョグ(JOGging):セットアップモード

K	
K_v	制御ループのゲイン係数
K_p	比例ゲイン
$K_Ü$	伝達比
LAD	ラダー図(Ladder Diagram)(PLC プログラミング方式)

L	
LAI	論理機械軸イメージ(Logic Machine Axis Image)
LAN	ローカルエリアネットワーク(Local Area Network)
LCD	液晶ディスプレイ(Liquid Crystal Display)
LED	発光ダイオード(Light Emitting Diode)
LF	ラインフィード(Line Feed)
PMS	位置検出器(Position Measuring System)
LR	位置コントローラ
LSB	最下位ビット(Least Significant Bit)
LUD	ローカルユーザーデータ(Local User Data)

M	
MAC	メディアアクセス制御(Media Access Control)
MAIN	メインプログラム(MAIN program)(OB1、PLC)
MB	メガバイト(MegaByte)
MCI	モーションコントロールインタフェース(Motion Control Interface)
MCIS	モーションコントロール情報システム(Motion Control Information System)
MCP	機械操作パネル(Machine Control Panel)
MD	マシンデータ(Machine Data)
MDI	手動データ入力(Manual Data Input)
MDS	モータデータセット(Motor Data Set)
MSGW	メッセージワード(MeSsaGe Word)
MCS	機械座標系(Machine Coordinate System)
MM	モータモジュール(Motor Module)

M	
MPF	メインプログラムファイル(Main Program File):メインプログラム(NC)
MCP	機械操作パネル(Machine Control Panel)

N	
NC	数値制御移動範囲などのブロック解析をおこなう数値制御
NCU	数値制御装置(Numerical Control Unit):NC ハードウェアユニット
NRK	NC のオペレーティングシステムの名称
IS	インタフェース信号(Interface Signal)
NURBS	非一様有理 B スプライン(Non-Uniform Rational B-Spline)
WO	ゼロオフセット(Work Offset)
NX	数値拡張(Numerical eXtension):軸の拡張基板

O	
OB	PLC のオーガニゼーションブロック(Organization Block in the PLC)
OEM	独自製品製造業者(Original Equipment Manufacturer)
OP	操作パネル(Operator Panel)
OPI	Operator Panel Interface:操作パネルへの接続用インタフェース
OPT	オプション(OPTion)
OLP	光リンクプラグ(Optical Link Plug):光ファイババスコネクタ
OSI	開放型システム間相互接続(Open Systems Interconnection):コンピュータ通信の標準規格

P	
PIQ	プロセスイメージ出力
PII	プロセスイメージ入力(Process Image Input)
PC	パーソナルコンピュータ(Personal Computer)
PCIN	制御装置とのデータ交信用ソフトウェアの名称
PCMCIA	パーソナルコンピュータメモリカード国際協会(Personal Computer Memory Card International Association): プラグインメモリカードの標準規格
PCU	PC ユニット(PC Unit):コンピュータ装置

P	
PG	プログラミング装置(Programming device)
PKE	パラメータ識別(Parameter identification):PIV の一部
PIV	パラメータ識別子(Parameter Identification Value):PPO のパラメータ設定部分
PLC	プログラマブル論理制御(Programmable Logic Control):カスタマイズ制御
PN	PROFINET
PNO	PROFIBUS 協会
PO	電源投入(POWER ON)
POU	プログラム構成単位(Program Organization Unit)
POS	位置/位置決め(Position/positioning)
POSMO A	POSitioning MOtor Actuator:位置決めモータ
POSMO CA	POSitioning MOtor Compact AC:位置決めユニットとプログラムメモリだけでなく、内蔵の電源および制御モジュール、つまり AC 電源装置を備えた一体型ドライブユニット
POSMO CD	POSitioning MOtor Compact DC:POSMO CA と似ているが DC 電源装置を搭載
POSMO SI	POSitioning MOtor Servo Integrated:位置決めモータ、DC 電源装置
PPO	パラメータプロセスデータオブジェクト(Parameter Process data Object):PROFIBUS DP 伝送用のサイクリックデータメッセージと「可変速ドライブ」プロファイル
PPU	パネルプロセスユニット(Panel Processing Unit): パネルベースの CNC のセントラルハードウェア、例: SINUMERIK 828D
PROFIBUS	PROcess FIeld BUS:シリアルデータバス
PRT	プログラムテスト(PRogram Test)
PSW	プログラムコントロールワード
PTP	ポイントツーポイント(Point-To-Point)
PUD	プログラムグローバルユーザーデータ(Program global User Data):プログラムグローバルのユーザー変数
PZD	プロセスデータ(Process data):PPO のプロセスデータ部分

Q	
QEC	象限突起補償(Quadrant Error Compensation)

R	
RAM	ランダムアクセスメモリ(Random Access Memory):読み取り/書き込みメモリ
REF	レファレンス点復帰機能(REFerence point approach function)
REPOS	再位置決め機能(REPOSition function)
RISC	縮小命令セットコンピュータ(Reduced Instruction Set Computer):小規模命令セットと高速で命令を処理する能力を備えたタイプのプロセッサ
ROV	早送りオーバライド(Rapid OVerride):入力補正
RP	R 変数(R Parameter)、算術変数、予約ユーザー変数
RPA	R 変数用の NCK 上のメモリ領域 R 変数番号用の NC 上のメモリ領域
RPY	ロールピッチヨー(Roll Pitch Yaw):座標系の回転タイプ
RTLI	早送り動作中の直線補間(Rapid Traverse Linear Interpolation)
RTS	Request To Send:シリアルデータインタフェースの制御信号
RTCP	リアルタイム制御プロトコル(Real Time Control Protocol)

S	
SA	シンクロナイズドアクション(Synchronized Action)
SBC	安全ブレーキ制御(Safe Brake Control)
SBL	シングルブロック(Single BLock)
SBR	サブプログラム(Subroutine)(PLC)
SD	セッティングデータ(Setting Data)
SDB	システムデータブロック(System Data Block)
SEA	Setting Data Active:セッティングデータの識別子(ファイルタイプ)
SERUPRO	プログラムテストによる検索実行ブロック検索、プログラムテスト
SFB	システムファンクションブロック(System Function Block)
SFC	システムファンクションコール(System Function Call)
SGE	安全関連入力
SGA	安全関連出力
SH	安全停止
SIM	シングルインラインモジュール(Single Inline Module)
SK	ソフトキー(SoftKey)

S	
SKP	スキップ(Skip):パートプログラムブロックをスキップするための機能
SLM	同期リニアモータ(Synchronous Linear Motor)
SM	ステッピングモータ(Stepper Motor)
SMC	取り付けセンサモジュールキャビネット(Sensor Module Cabinet Mounted)
SME	外部取り付けセンサモジュール外部(Sensor Module Externally Mounted)
SMI	内蔵センサモジュール(Sensor Module Integrated)
SPF	Sub Routine File:サブプログラム(NC)
PLC	プログラマブルロジックコントローラ(Programmable Logic Controller)
SRAM	スタテック RAM(Static RAM) (不揮発性)
TNRC	ノーズ R 補正(Tool Nose Radius Compensation)
SRM	回転型同期モータ(Synchronous Rotary Motor)
LEC	ピッチ誤差補正(Leadscrew Error Compensation)
SSI	同期シリアルインタフェース(Serial Synchronous Interface)
SSL	ブロック検索
STW	コントロールワード
GWPS	砥石周速制御(Grinding Wheel Peripheral Speed)
SW	ソフトウェア(SoftWare)
SYF	システムファイル(System Files)
SYNACT	シンクロナイズドアクション(SYNchronized ACTion)

T	
TB	端子基板(Terminal Board) (SINAMICS)
TCP	工具中心点(Tool Center Point):工具先端
TCP/IP	伝送制御プロトコル(Transport Control Protocol) /インターネットプロトコル(Internet Protocol)
TCU	シンクライアントユニット(Thin Client Unit)
TEA	Testing Data Active:マシンデータの識別子
TIA	統合オートメーション(Totally Integrated Automation)
TM	ターミナルモジュール(Terminal Module) (SINAMICS)
TO	工具オフセット(Tool Offset)
TOA	Tool Offset Active:工具オフセットの識別子(ファイルタイプ)

T	
TRANSMIT	TRANSform Milling Into Turning:施盤でのフライス削り運転用の座標変換
TTL	トランジスタ-トランジスタロジック(Transistor-Transistor Logic)(インタフェースタイプ)
TZ	テクノロジサイクル

U	
UFR	ユーザフレーム(User FRame):ゼロオフセット
SR	サブプログラム
USB	ユニバーサルシリアルバス(Universal Serial Bus)
UPS	無停電電源装置(Uninterruptible Power Supply)

V	
VDI	NC と PLC の間の内部通信インタフェース
VDI	ドイツエンジニア協会(Verein Deutscher Ingenieure)
VDE	ドイツ電気技術者協会(Verband Deutscher Elektrotechniker)
VI	電圧入力(Voltage Input)
VO	電圧出力(Voltage Output)
FDD	送り用ドライブ(Feed Drive)

W	
SAR	滑らかなアプローチと後退(Smooth Approach and Retraction)
WCS	ワーク座標系(Workpiece Coordinate System)
T	工具(Tool)
TLC	工具長補正(Tool Lenght Compensation)
WOP	現場指向プログラミング(Workshop-Oriented Programming)
WPD	ワークディレクトリ(WorkPiece Directory)
TRC	工具径補正(Tool Radius Compensation)
T	工具(Tool)
TO	工具オフセット(Tool Offset)

W	
TM	工具管理(Tool Management)
TC	工具交換(Tool change)

X	
XML	拡張マークアップ言語(eXtensible Markup Language)

Z	
WOA	ゼロオフセットの識別子(Work Offset Active)
ZSW	ステータスワード(ドライブの)

A.2 本書の概要

共通のマニュアル

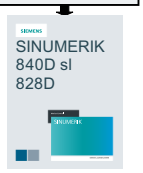
パンフレット
- SINUMERIK 840D sl
– SINUMERIK 828D
– SINUMERIK 828D BASIC

カタログNC 62
SINUMERIK 840D sl

カタログNC 82
SINUMERIK 828D

カタログPM 21
SIMOTION、
SINAMICS S120

システムマニュアル
インテリジェント機械
工具の操作



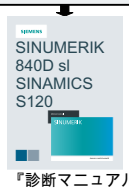
システムマニュアル
Ctrl-Energy

ユーザーマニュアル

操作マニュアル
– 汎用
– 旋削
– フライス加工
– 研削

プログラミングマニュアル
– 基本編
– 上級編
– 計測サイクル

プログラミングマニュアル
– 旋盤用Gコード
– マシニングセンタ用Gコード



『診断マニュアル』
アラーム

『診断マニュアル』
アラーム

メーカ/サービスマニュアル

製品マニュアル
– NCU
– オペレータコンポーネントとネットワーク設定
– ADI4

製品マニュアル
試運転マニュアル
サービスマニュアル

試運転マニュアル
– CNC:NCK、PLC、ドライブ
– ベースソフトウェアおよびオペレーティングソフトウェア

リストマニュアル
– マシンデータ
– インタフェース信号
– 変数

リストマニュアル
– マシンデータ
– インタフェース信号
– パラメータ
– 変数

システムマニュアル
ガイドライン
コンフィグレーションの実行
機械

メーカ/サービスマニュアル

機能マニュアル
– 基本機能
– 上級機能
– 応用機能
– シンクロナイズドアクション
– ISO系言語

機能マニュアル
工具管理機能

機能マニュアル
Safety Integrated

機能マニュアル
Safety Integrated

設定マニュアル
– EMC対策設置指針
– 工業安全性

情報/トレーニング

トレーニングマニュアル
– 簡単なフライス削りを効率化します
– 簡単な旋削を効率化します

マニュアル
ツールと
金型加工

電子マニュアル

DOConCD

Industry Online
サポート(SIOS)

Industry Mall

用語集

CNC

→ NC を参照してください。

コンピュータによる数値制御(Computerized Numerical Control):→ NC、→ PLC、HMI、→ COM のコンポーネントが含まれます。

CNC

→ NC を参照してください。

コンピュータによる数値制御(Computerized Numerical Control):→ NC、→ PLC、HMI、→ COM のコンポーネントが含まれます。

COM

通信の実行と調整用の NC コンポーネントです。

CPU

中央処理装置。 → PLC を参照してください。

CU

伝達比

C スプライン

C スプラインは最もよく知られていて広く使用されているスプラインです。補間点での遷移は、接線と曲率が連続的に変化するようにおこなわれます。3 次の多項式が使用されます。

C 軸

工具主軸の制御された回転移動および位置決め移動の中心となる軸。

DRF

差動レゾルバ機能(Differential Resolver Function):自動モードで手動パルス発生器と組み合わせてインクレメンタルゼロオフセットを生成する NC 機能です。

HIGHSTEP

AS300/AS400 システムの → PLC のプログラミングオプションの要約です。

HW Config

S7 プロジェクト内のハードウェアコンポーネントの構成とパラメータ設定用 SIMATIC S7 ツール

JOG

コントローラの運転モード(セットアップモード):機械は JOG モードで段取りができます。個々の軸と主軸を、方向キーによって JOG モードで移動できます。JOG モードには次の追加機能があります: → レファレンス点復帰、→再位置決め、および→プリセット(現在値の設定)。

KV

制御ループ内のサーボゲイン係数、制御変数です。

MDI

コントローラの運転モードです。手動データ入力。MDI モードでは、メインプログラムやサブプログラムと関係しない個別のブログラムブロックまたはブロック列を入力でき、その後にすぐに NC スタートキーの作動により実行できます。

NC

→ パートプログラムを実行し、工作機械の動作を制御する→ CNC の数値制御(Numerical Control)コンポーネントです。

NRK

数値ロボットカーネル(→ NC のオペレーティングシステム)

NURBS

コントローラ内のモーションコントロールと軌跡補間は、NURBS (**N**on **U**niform **R**ational **B**-**S**plines: 非一様有理 B スプライン)に基づいておこなわれます。これは、すべての内部補間に同じ処理を提供します。

OEM

個々のソリューション(OEM アプリケーション)を実現するための適用範囲は、独自のユーザーインタフェースの作成やテクノロジ固有機能のコントローラへの統合をおこないたい工作機械メーカに合わせて提供されています。

PCIN データ送受信プログラム

PCIN は、CNC ユーザーデータ(たとえば、パートプログラム、工具オフセットなど)をシリアルインタフェースを介して送受信するためのユーティリティプログラムです。PCIN プログラムは、標準の工業用 PC 上の MS-DOS で実行できます。

PLC

Programmable **L**ogic **C**ontroller (プログラマブルコントローラ): →プログラマブルロジックコントローラ。 NC のコンポーネント: 工作機械の論理制御を処理するためのプログラマブルコントローラ。

PLC プログラミング

STEP 7 ソフトウェアを使用して、PLC をプログラム指令します。 STEP 7 プログラミングソフトウェアは、**WINDOWS** 標準オペレーティングシステムの下で事項され、改良された拡張機能とともに STEP 5 プログラミング機能を含んでいます。

PLC プログラムメモリ

SINUMERIK 840D sl: PLC ユーザープログラム、ユーザーデータおよび PLC 基本プログラムは、まとめて PLC ユーザーメモリに格納されています。

RS-232-C

データ入出力用のシリアルインタフェース。 加工プログラムとメーカーデータ、ユーザーデータは、このインタフェースを介してロード、保存できます。

R 変数

プログラムにおける任意の目的のためにパートプログラムプログラマが設定または確認することができる算術変数。

TOA ユニット

各 TOA 領域は複数の TOA ユニットを持つことができます。 可能な TOA ユニットの数は、有効なチャネルの最大数によって制限されます。 TOA ユニットには、正確に 1 つのデータブロックと 1 つのマガジンデータブロックが含まれます。 また、TOA ユニットには工具ホルダデータブロック(オプション)を含めることもできます。

TOA 領域

TOA 領域には、すべての工具とマガジンのデータが含まれます。 初期設定で、データのアクセスに関してこの領域は→ チャネル領域と一致します。 ただし、マシンデータを使用して、複数のチャネルが 1 つの TOA ユニットを共有するように指定して、共通工具管理データをこれらのチャネルで使用可能にすることができます。

WinSCP

WinSCP は、ファイルを転送するために無料で使用できる Windows 用オープンソースプログラムです。

アーカイビング(Archiving)

外部メモリ機器にファイルとディレクトリから読み取ります。

アドレス

アドレスは、入力や出力などの特定のオペランドまたはオペランド範囲の識別子です。

アブソリュート指令

軸移動の移動先は、現在有効な座標系の原点を基準とする寸法によって定義されます。
→ インクレメンタル指令を参照してください。

アラーム

すべての → メッセージとアラームは、日付と時刻および解除条件に対応するシンボルとともに、操作パネルにプレーンテキストで表示されます。アラームとメッセージは別々に表示されます。

1. パートプログラムのアラームとメッセージ:
   アラームとメッセージは、パートプログラムから直接プレーンテキストで表示できます。
2. PLC からのアラームとメッセージ
   機械のアラームとメッセージは、PLC プログラムからプレーンテキストで表示できます。
   このために追加のファンクションブロックパッケージは必要ありません。

イグザクトストップ

イグザクトストップ命令がプログラム指令されている場合、ブロックに指定された位置は正確に、また必要であれば低速で位置決めされます。移動時間を短縮するため、早送りと送りに → イグザクトストップ範囲が定義されています。

イグザクトストップ範囲

すべての軌跡軸がイグザクトストップ範囲に達すると、制御装置は正確な終点に到達したものとして動作します。→ パートプログラムは次のブロックに進行します。

インクレメンタル指令

インクレメンタル指令:軸移動の目標は、移動する距離とすでに到達した点を基準とする方向によって定義されます。→ アブソリュート指令を参照してください。

インクレメント

インクレメント数に基づく移動軌跡長さの指定。 インクレント数は、セッティングデータとして格納するか、または適切にラベル付けされたキー(つまり、10、100、1000、10000)によって選択できます。

インチ単位系

インチおよびインチの小数で距離を定義する単位系。

インバースタイム送り

また、軸動作に対して、送り速度ではなく、ブロックの軌跡が移動するのに必要な時間をプログラム指令できます(G93)。

エディタ

エディタを使用すると、プログラム/テキスト/プログラムブロックを作成、編集、拡張、結合およびインポートできます。

オーバライド

特定のワークや材料に対して、プログラム指令送り速度や回転数を最適化するために、ユーザーによるオーバーライドを可能にする手動またはプログラム指令可能な介入機能。

オフセットメモリ

工具オフセットデータが格納されるコントローラ内のデータ領域です。

キースイッチ

→機械操作パネルのスイッチには、コントローラのオペレーティングシステムによって機能が割り当てられた 4 つの位置があります。 キースイッチには、3 つの異なる色のキーがあり、指定した位置で取り外すことができます。

キーワード

パートプログラムに対してプログラミング言語で定義された意味を持つ特定表記によるワード。

サイクル

→ワークに対して繰り返しおこなわれる加工運転を実行するための保護されたサブプログラム。

サブプログラム

「サブプログラム」という用語は、パートプログラムがメインプログラムとサブプログラムに厳密に分類されたときに生まれました。 今日の SINUMERIK NC 言語では、このように厳密には分類されていません。 原則として、すべてのパートプログラムまたはすべてのサイクルは、別のパートプログラム内でサブプログラムとして呼び出すことができます。 呼び出された後、次のプログラムレベル(x+1) (サブプログラムレベル (x+1))で実行されます。

サブブロック

たとえば番号などの順序に関する情報を含む「N」が頭に付くブロックです。

ジオメトリ

→ ワーク座標系での → ワークの記述です。

ジオメトリ軸

ジオメトリ軸は、→ パートプログラムで、ワークのジオメトリがプログラム指令される 2 次元または 3 次元の → ワーク座標系を形成します。

システムメモリ

システムメモリは、次のデータが格納されている CPU 内のメモリです。

- オペレーティングシステムに必要なデータ
- 演算タイマ、カウンタ、マーカ

システム変数

パートプログラムのプログラマによる入力が何もなくても存在する変数。 これは、**$**の文字が頭に付いた、データタイプと変数名称によって定義されます。 参照→ユーザー定義変数。

シンクロナイズドアクション

1. 補助機能出力
   ワーク加工中、技術的な機能(補助機能)を CNC プログラムから PLC に出力できます。 たとえば、次の補助機能を使用して、クイル、グラブ、クランピングチャックなどの工作機械の追加機器が制御されます。
2. 高速補助機能出力
   高速動作が必要なスイッチ機能のために、補助機能の応答時間を短縮して、加工処理での不要な待機時間を回避することができます。

スケーリング

軸別の尺度を変更するフレームの成分。

スプライン補間

スプライン補間を使用すると、コントローラは、設定輪郭のごくわずかな指定補間点から滑らかな曲線特性を生成できます。

セッティングデータ

工作機械の特性を NC に伝える、システムソフトウェアを通じて定義されたデータ。

ゼロオフセット

既存の原点と → フレームを基準として座標系の新しい基準点を指定します。

1. 設定可能
   任意の数の設定可能ゼロオフセットを CNC 軸毎に使用できます。オフセット - G 命令で選択された - が交互に有効になります。
2. 外部
   ワークの位置を定義するすべてのオフセットだけでなく、外部ゼロオフセットをハンドル(DRF オフセット)によって、または PLC から重畳することができます。
3. プログラム指令可能
   ゼロオフセットは、TRANS 命令を使用して、すべての軌跡軸と位置決め軸に対してプログラム指令できます。

ソフトウェアリミットスイッチ

ソフトウェアリミットスイッチは、軸の移動範囲を制限して、ハードウェアリミットスイッチでのスライドの急停止を防止します。 2 つの値のペアを軸毎に指定して、PLC によって別々に起動できます。

ソフトキー

その名称が画面の操作エリアに表示されるキー。 表示されるソフトキーの選択は、運転状況に随時、適用されます。 自由に割り当てられるファンクションキー(ソフトキー)には、ソフトウェアで定義された機能が割り当てられます。

チャネル

チャネルは、→ パートプログラムを他のチャネルと関係なく処理できるという特徴をもっています。チャネルは、そこに割り当てられた軸と主軸のみを制御します。別のチャネルで実行されるパートプログラムとは、→ 同期によって協調できます。

データブロック

1. → HIGHSTEP プログラムがアクセスできる → PLC のデータユニットです。
2. → NC のデータユニット:データブロックには、グローバルユーザーデータのデータ定義が含まれます。このデータは定義後そのまま初期化することができます。

データワード

→ データブロック内の 2 バイトデータユニットです。

テキストエディタ

参照→エディタ

トータルリセット

全体リセットのとき、CPU の次のメモリが削除されます。

- →作業メモリ
- ロードメモリの読み取り/書き込み領域
- →システムメモリ
- →バックアップメモリ

ドライブ

ドライブは、NC の設定に基づいて速度とトルクの制御を実行する CNC のユニットです。

ネットワーク

ネットワークとは、→ 接続ケーブルを介した、複数の S7-300 および、その他の端末機器(プログラミング機器など)のつながりのことをいいます。データ交信は、接続された機器の間でネットワークを通じておこなわれます。

ノーズ R 補正

輪郭プログラミングでは、工具が位置決めされるものと想定します。 実際にはそうならないため、使用される工具の曲率半径をコントローラに伝えて考慮に入れる必要があります。 曲率中心は、曲率半径によってオフセットされて、輪郭に等距離で維持されます。

パートプログラム

特定のワークを作成するために一体となって動作する NC に対する一連の命令。 同様に、この用語は特定の素材に対する特定の加工運転の実行にも適用されます。

パートプログラムブロック

ラインフィードによって仕切られるパートプログラムの部分。 次の 2 つのタイプがあります。 →メインブロックと→サブブロック

パートプログラム管理

パートプログラム管理はワークを基準に構成できます。 ユーザーメモリのサイズは、プログラムの数と、管理可能なデータ量を決定します。 各ファイル(プログラムとデータ)には、最大 24 文字の英数字からなる名称を指定できます。

バックアップバッテリ

バックアップバッテリによって、CPU のユーザープログラム格納されて停電から確実に防護され、指定されたデータ領域とビットメモリ、タイマおよびカウンタが確実に保持されます。

バックラッシ補正

機構上の機械バックラッシ、たとえばボールネジの反転時のバックラッシなどを補正します。バックラッシ補正は軸毎に別々に入力できます。

ピッチ誤差補正

送りに関与する送りねじの機械的な誤差に対する補正。 コントローラは、保存された誤差値を補正に使用します。

ブート

電源投入後のシステムプログラムのロードです。

フレーム

フレームは、1 つの直交座標系を別の座標系に変換する演算規則です。フレームには次の成分が含まれます: →ゼロオフセット、→座標回転、→スケーリング、→ミラーリング。

フローティングチャックなしのタッピング

この機能を使用すると、フローティングチャックなしでねじをタッピングできます。回転軸と穴あけ軸としての主軸の補間方法を使用することによって、たとえば止まり穴ねじのように、ねじを最終穴あけ深さまで正確に切削できます(必要条件:主軸の軸運転)。

プログラマブルコントローラ(Programmable logic controller)

プログラマブルコントローラ(PLC)は電子コントローラであり、その機能はコントロールユニットにプログラムとして格納されています。 つまり、機器のレイアウトと配線は、コントローラの機能に依存しません。 プログラマブルコントローラは、コンピュータと同じ構成です。これは、メモリを備えた CPU(中央モジュール)、入出力モジュールおよび内部バスシステムからなります。 周辺機器とプログラミング言語は、コントローラの機能の要求事項と一致します。

プログラマブルフレーム

プログラマブルフレームを使用すると、パートプログラムの実行中、新しい座標系出力点の動的定義が可能になります。 新しいフレームを使用する絶対定義と、既存の起点を基準とする追加定義が区別されます。

プログラマブル作業領域リミット

工具の移動空間を、プログラム指令範囲によって定義された空間に制限します。

プログラミングキー

→パートプログラム用プログラミング言語で定義された意味を持つ文字と文字列。

プログラムブロック

プログラムブロックには、→パートプログラムのメインプログラムとサブプログラムが含まれています。

プログラムレベル

チャネル内で開始されたパートプログラムは、プログラムレベル 0 (メインプログラムレベル)でメインプログラムとして実行されます。 メインプログラムで呼び出されたすべてのパートプログラムは、それ自体のプログラムレベル 1 ... n でサブプログラムとして実行されます。

ブロック

「ブロック」は、プログラムの作成と処理に必要なすべてのファイルを示す用語です。

ブロック検索

デバッグ目的で、またはプログラム中止に続いて、「ブロック検索」機能を使用して、プログラムを開始または再開するパートプログラム内の任意の位置を選択できます。

プロテクションゾーン

工具先端を通過させてはならない作業領域内の 3 次元領域。

ヘリカル補間

ヘリカル補間機能は、成形フライスカッタを使用する、おねじ、および、めねじの加工と、潤滑溝のフライス加工に適しています。

ヘリカルは 2 つの移動で構成されます。

- 1 つの平面での円移動
- この平面に垂直な直線移動

ボーレート

データ送受信速度(bps)。

マクロ機能

1 つの識別子による一連の命令のグループです。この識別子は、プログラムのなかで一連の統合された命令を表します。

マスタ軸

マスタ軸は→ガントリ軸で、オペレータとプログラムの視点からみると存在し、そのため標準的な NC 軸のように影響を受けます。

ミラーリング

ミラーリングは、軸に関して輪郭の座標値の符号を反転します。 一度に複数の軸に関してミラーリングをおこなうことができます。

メインプログラム

「メインプログラム」という用語は、パートプログラムがメインプログラムと→ サブプログラムに厳密に分類されたときに生まれました。今日の SINUMERIK NC 言語では、このように厳密には分類されていません。原則として、チャネル内のすべてのパートプログラムを選択して開始できます。パートプログラムは→ プログラムレベル 0 (メインプログラムレベル)で実行されます。また、パートプログラムまたはサブプログラムである→ サイクルをメインプログラムで呼び出すことができます。

メインブロック

→ パートプログラム内の動作シーケンスを開始するためのすべての情報が含まれた、先頭に「:」の付いているブロック。

メッセージ

パートプログラムでプログラム指令されたすべてのメッセージおよびシステムによって検出されたすべての → アラームは、日付と時刻および解除条件に対応するシンボルとともに、操作パネルにプレーンテキストで表示されます。アラームとメッセージは別々に表示されます。

メトリック単位系

基本単位系:長さの場合は、たとえば mm(ミリメートル)、m(メートル)です。

モード

SINUMERIK コントロールシステムの運転仕様です。次のモードが定義されています。→ ジョグ → MDI、→ 自動。

モードグループ

技術的に関連する軸と主軸は、1 つのモードグループに組み合わせることができます。モードグループの軸/主軸は、1 つまたは複数の → チャネルによって制御できます。同じ → モードタイプが、常に、そのモードグループのチャネルに割り当てられます。

ユーザーインタフェース

ユーザーインタフェース(UI)は、画面による CNC 用の表示媒体です。これは、水平と垂直ソフトキーが特徴です。

ユーザープログラム

S7-300 オートメーションシステムのユーザープログラムは、プログラミング言語 STEP 7 を使用して作成されます。ユーザープログラムはモジュール構造で、個々のブロックから構成されます。

基本ブロックタイプは次のとおりです。

- コードブロック
  これらのブロックには、STEP 7 の命令が含まれます。
- データブロック
  これらのブロックには、STEP 7 プログラムの定数と変数が含まれます。

ユーザーメモリ

パートプログラム、サブプログラム、コメント、工具オフセット、ゼロオフセット/フレーム、さらにチャネル、およびプログラムユーザーデータなどのすべてのプログラムとデータを共有 CNC ユーザーメモリに格納できます。

ユーザー定義変数

ユーザーは、→ パートプログラムまたはデータブロック(グローバルユーザーデータ)で任意の目的で独自の変数を宣言できます。定義には、データタイプ指定と変数名称が含まれます。→ システム変数を参照してください。

レファレンス点

機械軸の検出器が基準とする工作機械の位置。

ロードメモリ

ロードメモリは、 → PLC の → CPU 314 の→ 作業メモリと同じです。

ワーキングエリアリミット

ワーキングエリアリミットによって、軸の移動範囲をリミットスイッチに加えてさらに制限することができます。軸あたりに 1 つの値ペアを使用して、保護する作業領域を記述することができます。

ワーク

工作機械によって作成/加工される部品。

ワーク原点

ワーク原点は、ワーク座標系の起点です。 これは、機械原点との距離に関連して定義されます。

ワーク座標系

ワーク座標系は、ワーク原点に、その起点があります。 ワーク座標系でプログラム指令された加工運転で、寸法と方向はこの座標系を基準とします。

ワーク輪郭

作成または加工されるワークの指令輪郭。

安全機能(Safety functions)

コントローラは、→CNC、→PLC および機械の故障を適切に検出する常に有効な監視機能を備えているため、ワーク、工具または機械の損傷の大部分は回避されます。 故障の際は、加工運転は中断されてドライブは停止します。 誤動作の原因が記録されて、アラームとして出力されます。 同時に、PLC には CNC アラームがトリガされたことが通知されます。

位置決め軸

工作機械での補助移動(工具マガジン、パレット搬送など)をおこなう軸。 位置決め軸とは、軌跡軸と補間しない軸のことです。

移動範囲

直線軸の最大許容移動範囲は、±9 桁です。 アブソリュート値は、選択された最小入力単位と位置制御分解能および基本単位(インチまたはメートル)によって決まります。

円弧補間

→ 工具は輪郭上の指定された点の間の円の上を指定された送り速度で移動し、それによってワークが加工されます。

加々速度一定加減速

機構部を保護しながら同時に機械の加減速応答を最適化するために、パートプログラムで不連続な加減速と連続的な加減速(加々速度一定)を切り替えることができます。

加工チャネル

チャネル構造は、たとえばローディングクレーンを加工と同時に移動するなどの並列動作処理によってアイドル時間を短縮するために使用できます。ここで、CNC チャネルは、解読、ブロック解析および補間において、別の CNC コントロールシステムとみなす必要があります。

回転軸

回転軸は、定義された角度へのワークまたは工具の回転を実現します。

外部ゼロオフセット

→ PLC によって指定されたゼロオフセットです。

割り込みルーチン

割り込みルーチンは加工処理中にイベント(外部信号)によって起動できる特殊な→ サブプログラムです。現在動作中のパートプログラムブロックが中断されて、中断点での軸の位置が自動的に保存されます。

割り出し軸

割り出し軸は、インデックスグリッドに対応する角度までワークまたは工具を回転させます。 グリッドに到達すると、割り出し軸は「インポジション」になります。

完成品の輪郭

完成ワークの輪郭です。→ 素材を参照してください。

基本座標系

座標変換によって機械座標系へ投影される直交座標系です。

プログラマは → パートプログラムで基本座標系の軸名称を使用します。基本座標系は、→ 座標変換が有効ではない場合 → 機械座標系に平行して存在します。→ 軸名称に違いがあります。

基本軸

その指令値または現在位置が、補正値の計算の基礎を形成する軸です。

機械原点

(得られた)すべての検出位置へ復帰できる工作機械の固定点です。

機械固定点

工作機械によって一義的に定義される点、たとえば機械レファレンス点です。

機械座標系

工作機械の軸に関連する座標系です。

機械軸

工作機械に物理的に存在する軸です。

機械操作パネル

キー、ロータリスイッチなどの操作部品と LED などの単純な表示器を備えた工作機械上の操作パネルです。これを使用して、PLC を介して工作機械を直接操作します。

軌跡誤差監視

追従誤差は、輪郭精度の尺度として定義可能な許容誤差範囲内で監視されます。許容できないほど大きな追従誤差があると、たとえばドライブが過負荷になる場合があります。このような場合はアラームが出力されて、軸が停止します。

軌跡軸

軌跡軸には、→ 補間器によって起動、加速、停止および終点への到達が同時におこなわれるような方法で制御される → チャネルのすべての機械軸が含まれます。

軌跡送り速度

軌跡送り速度は → 軌跡軸に作用します。関連する → ジオメトリ軸の送り速度のジオメトリック合計を表しています。

軌跡速度

プログラム指令可能な最大軌跡速度は、最小入力単位によって決まります。たとえば、最小単位 0.1mm の場合、プログラム指令可能な最大軌跡速度は 1000m/min になります。

曲率

輪郭の曲率 k は輪郭点に沿った円弧半径 r の逆数(k = 1/r)です。

極座標

平面上の点の位置を、基点からの距離と、定義済み軸での半径ベクトルによって形成された角度によって定義する座標系。

傾斜面加工

「傾斜面加工」機能を使用して、機械の座標平面にないワーク面に対する穴あけ加工とフライス加工運転を簡単におこなうことができます。

固定点アプローチ

工作機械は、工具交換位置、ロードポイント、パレット交換位置などの固定点に、定義された方法で移動できます。これらの点の座標はコントローラ内に格納されています。コントローラは、可能な場合はいつも、関連する軸を → 早送りで移動します。

工具

加工を実行する工作機械の作用部分(旋削工具、フライス工具、ドリル、レーザービームなど)。

工具径補正

必要なワーク輪郭を直接プログラム指令するには、コントローラは、使用される工具の半径を考慮に入れて、プログラム指令輪郭に一定の距離を置く軌跡を移動させる必要があります(G41/G42)。

工具補正

軌跡計算における工具寸法の考慮。

高級 CNC 言語

高級言語は、NC プログラム、→ シンクロナイズドアクション、および→ サイクルを書き込むために使用されます。これは次のものを提供します:制御構造 → ユーザー定義変数、→システム変数、→マクロプログラミング。

高速ディジタル入/出力

ディジタル入力部を使用して、たとえば高速 CNC プログラムルーチン(割り込みルーチン)を起動できます。 高速のプログラム駆動スイッチ機能を、ディジタル CNC 出力で起動できます。

座標回転

ある角度での座標系の回転を定義するフレームの成分。

座標系

→ 機械座標系と → ワーク座標系を参照してください。

座標変換

軸の追加またはアブソリュートゼロオフセット。

作業メモリ

作業メモリは、アプリケーションプログラムを処理するときにプロセッサがアクセスする → CPU の RAM です。

作業領域

工作機械の物理的な構成を考慮して、工具先端が移動できる 3 次元の領域です。→ プロテクションゾーンを参照してください。

事前一致

軌跡残距離が終了位置の指定可能な範囲に等しい量に近づくと、ブロック切り替えがおこなわれます。

自動

コントローラの運転モード(DIN に準拠したブロックシーケンス運転):→ パートプログラムが選択されて連続的に実行される NC システムの運転モードです。

識別子

DIN 66025 に従って、ワードは、変数(算術変数、システム変数、ユーザー変数)、サブプログラム、キーワードおよび複数のアドレス文字を持つワードの識別子(名称)を使用して補完されます。これらの補完は、ブロック形式に関してワードと同じ意味を持ちます。識別子は一義的にしてください。異なる複数の対象には同じ識別子を使用できません。

軸

対象機能に従って、CNC 軸は次のように分割されます。

- 軸:軌跡軸の補間
- 補機軸:軸別の送り速度による非補間の軸送りと軸の位置決め。たとえば、工具供給や工具マガジンなど、補機軸は実際の加工にはかかわりません。

軸アドレス

→ 軸名称を参照してください。

軸名称

明確な識別を確実におこなうために、すべてのチャネルとコントロールシステムの→ 機械軸をチャネルとコントロールシステム内で一義的な名称で指定してください。→ ジオメトリ軸は X、Y、Z と呼ばれます。ジオメトリ軸 → を中心に回転する回転軸は、A、B、C と呼ばれます。

主軸オリエンテーション

指定された角度でワーク主軸を停止します。たとえば、特定の位置で追加の加工を実行するためです。

周辺機器

I/O モジュールは、CPU とプロセスの間の接続をおこないます。

I/O モジュールは次のとおりです。

- →ディジタル入/出力モジュール
- →アナログ入/出力モジュール
- →シミュレータモジュール

象限突起補償

象限切り替え時の輪郭誤差は、案内面での摩擦条件の変化の結果として生じますが、象限突起補償によって、ほとんどすべて取り除けます。 突起誤差補正のパラメータ設定は、真円度テストによっておこなわれます。

診断

1. コントローラの操作エリアです。
2. コントローラには、サービスのためのテスト機能と自己診断プログラムの両方があります。状態、アラームとサービス表示

寸法指定、メトリックとインチ

位置値とピッチ値は、加工プログラムでインチ単位でプログラム指令できます。プログラム指令寸法(G70/G71)に関係なく、コントローラは基本単位系に設定されます。

制限速度

最大/最小(主軸)速度:主軸の最大速度は、マシンデータ、→ PLC、または → セッティングデータを指定することによって制限できます。

接地

接地は、誤動作時でも危険な接触電圧によって活線状態にならない、機器のリンクされた非活性部分の総体としてとられます。

先読み

先読み機能を使用して、割り当て可能な数の移動ブロックを先読みすることによって、最適な加工速度を実現します。

素材

加工される前のワーク。

早送り

軸の最大移動速度。たとえば、早送りは、工具が停止位置から → ワーク輪郭にアプローチするとき、または工具がワーク輪郭から後退するときに使用されます。早送り速度は、マシンデータ単位を使用して機械ごとに設定されます。

送り速度オーバライド

プログラム指令速度は、機械操作パネルを介して、または PLC からおこなわれた、現在速度設定によってオーバライドされます(0 ～ 200%)。 送り速度は、加工プログラムでプログラム可能なパーセンテージ係数(1 ～ 200%)によって修正できます。

速度制御

ブロックあたりの移動量が非常にわずかな場合に、許容できる移動速度を実現するために、複数のブロックわたる先行解析(→ 先読み)を指定できます。

多項式補間

多項式補間を使用すると、さまざまな曲線特性を生成できます。たとえば、**直線、放物線、指数関数**などです(SINUMERIK 840D sl)。

中間ブロック

選択した工具補正(G41/G42)による移動は、一定の中間ブロック(補正平面で軸移動のないブロック)によって中断できます。これにより、工具補正を引き続き正しく実施できます。 コントローラが先読みする中間ブロックの許容数は、システムパラメータで設定できます。

直線軸

回転軸とは異なり、直線軸は直線を描きます。

直線補間

工具は、ワークを加工する間、終点に向かって直線に沿って移動します。

動的フィードフォワード制御

追従誤差による輪郭の不正確さは、動的加速度依存フィードフォワード制御を使用して現実的に、取り除くことができます。 この結果、高い軌跡速度でも優れた加工精度を達成できます。 フィードフォワード制御は、パートプログラムによって軸毎に選択と解除できます。

同期

特定の加工時点で異なるチャネルの順序調整をするためのパートプログラム内の命令。

同期軸

同期軸は → ガントリ軸で、その指令位置が → マスタ軸の移動から連続的に得られるため、マスタ軸と同期して移動します。プログラマとオペレータの観点から見ると、同期軸は「存在しません」。

同期軸

同期軸は、その軌跡を移動するために、ジオメトリ軸の軌跡を移動するのと同じ時間が掛かります。

非同期サブプログラム(Asynchronous SUBprogram)

割り込み信号(たとえば、「高速 NC 入力」信号)を使用して、現在のプログラム状態と非同期で(無関係に)起動できるパートプログラムです。

標準サイクル

標準サイクルは、下記の用途で頻繁に繰り返される加工運転のために設けられています。

- 穴あけ/フライス加工用
- 旋削加工用

使用可能なサイクルは、[プログラム]操作エリアの[サイクルサポート]メニューに表示されています。 必要な加工サイクルを選択すると、値の割り当ての必要なパラメータがプレーンテキストで表示されます。

変数定義

変数定義には、データタイプと変数名称指定が含まれます。 変数名称を使用して、変数の値にアクセスできます。

補間器

パートプログラムで指定された最終位置の情報に基づいて、個々の軸で実行される移動の中間値を定義する → NC の論理ユニットです。

補間型補正

機械の機械的な誤差を、→ ピッチ誤差、真直度補正、直角度補正、熱変位補正などの補間型補正機能により補正します。

補助機能

補助機能を使用すると、→ パートプログラムは→ パラメータを→ PLC に伝送でき、それによって、工作機械メーカによって定義された動作を起動します。

補正テーブル

補間点を含むテーブルです。これは、基準軸上の指定された位置に対する補正軸の補正値を提供します。

補正軸

補正値によって補正された指令値または現在値を持つ軸です。

補正値

エンコーダによって測定された軸位置と、プログラム指令された目標の軸位置との差です。

輪郭

→ ワークの輪郭

輪郭からの高速リトラクト

割り込みが発生すると、CNC 加工プログラムによって動作を開始して、現在加工中のワーク輪郭から工具を素早く退避させることができます。 退避角度と退避距離はパラメータ設定することもできます。 割り込みルーチンは以下の高速リトラクトでも実行することができます。

連続軌跡モード

連続軌跡モードの目的は、パートプログラムブロック境界での → 軌跡軸の実質的な減速を回避して、可能な限り同じ軌跡速度に近い速度で次のブロックに移ることにあります。

索引

-

- 試し切り追加終了 - GROUP_ADDEND
    外部プログラミング, 1003

$

$AA_ATOL, 656
$AA_COUP_ACT
    軸間連動機能, 692
    連結移動時, 664
$AA_ESR_ENABLE, 802
$AA_LEAD_SP, 692
$AA_LEAD_SV, 692
$AC_ACT_PROG_NET_TIME, 788
$AC_ACTUAL_PARTS, 792
$AC_AXCTSWA, 770
$AC_AXCTSWE, 770
$AC_CTOL, 656
$AC_CUT_INV, 566
$AC_CUTMOD, 565
$AC_CUTMOD_ANG, 565
$AC_CUTMODK, 566
$AC_CUTTING_TIME, 788
$AC_CYCLE_TIME, 788
$AC_DELAYFST, 636
$AC_ESR_TRIGGER, 802
$AC_OLD_PROG_NET_TIME, 788
$AC_OLD_PROG_NET_TIME_COUNT, 788
$AC_OPERATING_TIME, 788
$AC_OTOL, 656
$AC_PROG_NET_TIME_TRIGGER, 789
$AC_REPOS_PATH_MODE, 646
$AC_REQUIRED_PARTS, 792
$AC_SMAXVELO, 651
$AC_SMAXVELO_INFO, 651
$AC_SPECIAL_PARTS, 792
$AC_TOTAL_PARTS, 792
$AC_TRAFO_CORR_ELEM_P, 483
$AC_TRAFO_CORR_ELEM_T, 483
$AC_TRAFO_ORIAX_LOC, 484
$AN_AXCTAS, 770
$AN_AXCTSWA, 770
$AN_ESR_TRIGGER, 802
$AN_LANGUAGE_ON_HMI, 1141
$AN_POWERON_TIME, 787
$AN_SETUP_TIME, 787
$NT_CLOSE_CHAIN_T, 483
$NT_CNTRL, 483
$NT_CORR_ELEM_P, 483
$NT_CORR_ELEM_T, 483
$NT_NAME, 473
$NT_ROT_AX_NAME, 561
$NT_TRAFO_INDEX, 473
$P_ACTBFRAME, 372
$P_AD, 565
$P_BFRAME, 372
$P_CHBFRAME, 372
$P_CHBFRMASK, 373
$P_CTOL, 657
$P_CUT_INV, 566
$P_CUTMOD, 565
$P_CUTMOD_ANG, 565
$P_CUTMOD_ERR, 567
$P_CUTMODK, 566
$P_DELAYFST, 636
$P_IFRAME, 374
$P_IS_EES_PATH, 264
$P_NCBFRAME, 372
$P_NCBFRMASK, 373
$P_ORI_DIFF, 555
$P_ORI_POS, 554
$P_ORI_SOL, 556
$P_ORI_STAT, 559
$P_OTOL, 657
$P_PATH, 263
$P_PFRAME, 374
$P_PROG, 263
$P_PROGPATH, 263
$P_SIM, 337
$P_STACK, 263
$P_SUBPAR, 190
$P_TOOLENV, 577
$P_TOOLENVN, 577
$PA_ATOL, 657
$SA_LEAD_TYPE, 691
$SC_CONTPREC, 629
$SC_MINFEED, 629
$SN_PA_ACTIV_IMMED, 277
$TC_CARR1...14, 535
$TC_CARR18...23, 535
$TC_CARR18[m], 540
$TC_DP1 ... 25, 485
$TC_ECPxy, 490
$TC_SCPxy, 490

*

* (算術機能), 86

/

/ (算術機能), 86

+

+ (算術機能), 86

<

< (比較演算子), 88
<< (結合演算子), 100
<= (関係演算子), 88
<> (比較演算子), 88

=

== (比較演算子), 88

>

> (比較演算子), 88
>= (関係演算子), 88

0

0 文字, 97

A

ABS, 86
ACCLIMA, 624
ACOS, 86
ACTBLOCNO, 203
ACTFRAME, 345
ADISPOSA, 340
ALF
　輪郭からの高速リトラクト, 156
APR, 52
APRB, 52
APRP, 52
APW, 52
APWB, 52
APWP, 52
AS, 246
ASIN, 86
ASPLINE, 295
ATAN2, 86
ATOL, 653
AV, 701
AX, 761
AXCTSWE, 769
AXCTSWEC, 769
AXCTSWED, 769
AXIS, 33
AXNAME, 98
AXSTRING, 761
AXTOCHAN, 165
AXTOSPI, 761
A スプライン, 302

B

B_AND, 88
B_NOT, 88
B_OR, 88
B_XOR, 88
BAUTO, 295
BFRAME, 345
BLOCK, 231
BLSYNC, 149
BNAT, 295
BOOL, 33
BOUND, 93
BP, 81
BRISK, 621
BRISKA, 621
BSPLINE, 295
BTAN, 295
B スプライン, 303

C

CAC, 293
CACN, 293
CACP, 293
CALL, 230
CALLPATH, 235
CASE, 121
CDC, 293
CFINE, 357
CHAN, 33
CHANDATA, 265
CHAR, 33

CHKDNO, 530
CIC, 293
CLEARM, 140
CLRINT, 152
COARSE, 701
COARSEA, 340
COLLPAIR, 466
COMPCAD, 309
COMPCURV, 309
COMPLETE, 265
COMPOF, 309
COMPON, 309
COMPSURF, 309
CONTDCON, 827
CONTPRON, 820
CORRTRAFO, 475
COS, 86
COUPDEF, 701
COUPDEL, 701
COUPOF, 701
COUPOFS, 701
COUPON, 701
COUPONC, 701
COUPRES, 701
CP, 441
CPBC, 715
CPDEF, 713
CPDEL, 713
CPFMOF, 718
CPFMON, 717
CPFMSON, 716
CPFPOS + CPOF, 718
CPFPOS + CPON, 715
CPFRS, 714
CPLA, 713
CPLCTID, 714
CPLDEF, 713
CPLDEL, 713
CPLDEN, 714
CPLINSC, 721
CPLINTR, 721
CPLNUM, 714
CPLOF, 714
CPLON, 714
CPLOUTSC, 722
CPLOUTTR, 721
CPLPOS, 715
CPLSETVAL, 714
CPMALARM, 722
CPMBRAKE, 722
CPMPRT, 721
CPMRESET, 719
CPMSTART, 721
CPMVDI, 722
CPOF, 714
CPON, 714
CPRECOF, 629
CPRECON, 629
CPROT, 274
CPROTDEF, 269
CPSETTYPE, 722
CPSYNCOP, 722
CPSYNCOP2, 722
CPSYNCOV, 722
CPSYNFIP, 722
CPSYNFIP2, 722
CPSYNFIV, 722
CSPLINE, 295
CTAB, 679
CTABDEF, 666
CTABDEL, 673
CTABEND, 666
CTABEXISTS, 673
CTABFNO, 684
CTABFPOL, 684
CTABFSEG, 684
CTABID, 677
CTABINV, 679
CTABISLOCK, 677
CTABLOCK, 675
CTABMEMTYP, 677
CTABMPOL, 684
CTABMSEG, 684
CTABNO, 684
CTABNOMEM, 684
CTABPERIOD, 677
CTABPOL, 684
CTABPOLID, 684
CTABSEG, 684
CTABSEGID, 684
CTABSEV, 679
CTABSSV, 679
CTABTEP, 679
CTABTEV, 679
CTABTMAX, 679
CTABTMIN, 679
CTABTSP, 679
CTABTSV, 679
CTABUNLOCK, 675
CTOL, 653
CTRANS, 357
CUT3DC, 507
CUT3DCC, 517
CUT3DCCD, 517

CUT3DCD, 507
CUT3DF, 511
CUT3DFD, 511
CUT3DFF, 511
CUT3DFS, 511
CUTMOD, 561
CUTMODK, 561
CYCLE4071
外部プログラミング, 980
CYCLE4072
外部プログラミング, 982
CYCLE4073
外部プログラミング, 985
CYCLE4074
外部プログラミング, 987
CYCLE4075
外部プログラミング, 991
CYCLE4077
外部プログラミング, 993
CYCLE4078
外部プログラミング, 997
CYCLE4079
外部プログラミング, 999
CYCLE435 - ドレッサー座標系の設定
外部プログラミング, 928
CYCLE495 - 成形研削
外部プログラミング, 929
CYCLE60 - 彫刻
外部プログラミング, 863
CYCLE61 - 正面削り
外部プログラミング, 866
CYCLE62- 輪郭の呼び出し
外部プログラミング, 869
CYCLE63 - 輪郭ポケットのフライス削り
外部プログラミング, 870
CYCLE64 - 輪郭ポケットの前加工
外部プログラミング, 874
CYCLE70 - ねじフライス削り
外部プログラミング, 876
CYCLE72 - 輪郭フライス削り
外部プログラミング, 878
CYCLE76 - 長方形スピゴット
外部プログラミング, 882
CYCLE77 - 円形スピゴット
外部プログラミング, 885
CYCLE78 - ドリルねじフライス削り
外部プログラミング, 888
CYCLE79 - 多角形
外部プログラミング, 892
CYCLE800 - 旋回
外部プログラミング, 931
CYCLE801 - 格子/フレーム位置決めパターン
外部プログラミング, 937
CYCLE802 - 任意の位置
外部プログラミング, 939
CYCLE81 - センタリング
外部プログラミング, 895
CYCLE82 - 穴あけ
外部プログラミング, 896
CYCLE83 - 深穴あけ
外部プログラミング, 899
CYCLE830 - 深穴あけ 2
外部プログラミング, 943
CYCLE832 - 高速設定
外部プログラミング, 951
CYCLE84 - フローティングチャックなしのタッピング
外部プログラミング, 903
CYCLE840 - フローティングチャックありのタッピング
外部プログラミング, 955
CYCLE85 - リーマ加工
外部プログラミング, 909
CYCLE86 - ボーリング
外部プログラミング, 910
CYCLE899 - オープン溝のフライス削り
外部プログラミング, 959
CYCLE92 - 突切り
外部プログラミング, 912
CYCLE930 - 溝削り
外部プログラミング, 962
CYCLE940 - アンダーカット
外部プログラミング, 966
CYCLE95 - 輪郭切削
外部プログラミング, 914
CYCLE951 - 旋削
外部プログラミング, 969
CYCLE952 - 輪郭の溝削り
外部プログラミング, 973
CYCLE98 - 連続ねじ
外部プログラミング, 917
CYCLE99 - ねじの旋削
外部プログラミング, 923
C スプライン, 304

D

DCI, 57
DCM, 57
DCU, 57
DEF, 33
DEFAULT, 121
DEFINE ... AS, 246

DELAYFSTOF, 634
DELAYFSTON, 634
DELDL, 491
DELETE, 173
DELOBJ, 457
DELTOOLENV, 574
DIN サブプログラム名称, 260
DISABLE, 151
DISPLOF, 203
DISPLON, 203
DISPR, 640
DIV, 86
DI, 489
DO, 729
DRIVE, 621
DRIVEA, 621
DV, 701
DYNFINISH, 625
DYNNORM, 625
DYNPOS, 625
DYNROUGH, 625
DYNSEMIFIN, 625
D 番号
    確認, 530
    任意割り当て, 530
    名称変更, 531

E

Easy XML, 781
EAUTO, 295
EES, 254
EES 表記, 256
EG
    電子ギヤ, 693
EGDEF, 693
EGDEL, 699
EGOFC, 698
EGOFS, 698
EGON, 694
EGONSYN, 694
EGONSYNE, 694
ELSE, 133
ENABLE, 151
ENAT, 295
ENDFOR, 135
ENDIF, 133
ENDLABEL の間にあるパートプログラム区間, 124
ENDLOOP, 134
ENDWHILE, 137
ESR, 801
ESRR, 809
ESRS, 808
ETAN, 295
EVERY, 729
EXECSTRING, 84
EXECTAB, 833
EXECUTE, 836
EXP, 86
EXTCALL
    SINUMERIK 828D 用, 241
    SINUMERIK 840D sl 用, 236
EXTCLOSE, 793
EXTERN, 224
EXTOPEN, 793

F

FALSE, 33
FCTDEF, 502
FCUB, 615
FENDNORM, 338
FFWOF, 628
FFWON, 628
FIFOCTRL, 631
FILEDATE, 180
FILEINFO, 180
FILESIZE, 180
FILESTAT, 180
FILETIME, 180
FINE, 701
FINEA, 340
FLIN, 615
FNORM, 615
FOR, 135
FPO, 615
FRAME, 33
FROM, 729
FTOCOF, 506
FTOCON, 506

G

G コード
    間接プログラミング, 79
G290, 816
G291, 816
G5, 436
G62, 338
G621, 338
G7, 436
G810 ～ G819, 336
G820 ～ G829, 336

GEOAX, 764
GET, 160
GETACTTD, 532
GETD, 160
GETDNO, 531
GETTCOR, 577
GETTENV, 575
GETVARAP, 70
GETVARDFT, 73
GETVARDIM, 73
GETVARLIM, 72
GETVARPHU, 69
GETVARTYP, 75
GOTO, 118
GOTOB, 118
GOTOC, 118
GOTOF, 118
GOTOS, 117
GROUP_ADDEND - 試し切り追加終了
　外部プログラミング, 1003
GROUP_BEGIN - プログラムブロック開始
　外部プログラミング, 1001
GROUP_END -プログラムブロック終了
　外部プログラミング, 1002
GUD, 34
G グループ
　テクノロジ, 625

H

HOLES1 - 直線位置決めパターン
　外部プログラミング, 841
HOLES2 - 円弧位置決めパターン
　外部プログラミング, 842

I

ID, 729
IDS, 729
IF, 133
IFRAME, 345
INDEX, 103
INICF, 33
INIPO, 33
INIRE, 33
INIT, 140
INITIAL, 265
INITIAL_INI, 265
INT, 33
INTERSEC, 831
IPOBRKA, 340
IPOENDA, 340
IPOSTOP, 701
IPTRLOCK, 637
IPTRUNLOCK, 637
ISAXIS, 761
ISFILE, 178
ISNUMBER, 98
ISOCALL, 233
ISVAR, 67

J

JERKLIM, 649
JERKLIMA, 624

L

L..., 222
LEAD, 393
LEADOF, 687
LEADON, 687
LENTOAX, 602
LIFTFAST, 153
Link
　変数, 31
LLI, 48
LN, 86
LONGHOLE - 長穴
　外部プログラミング, 860
LOOP, 134
LUD, 34

M

M17, 208
M30, 208
MASLDEF, 724
MASLDEL, 724
MASLOF, 724
MASLOFS, 724
MASLON, 724
MATCH, 103
MAXVAL, 93
MCALL, 228
MD10010, 140
MD10280, 140
MD15800, 30
MD18104;ＭＤ１８１０４, 573
MD18116, 574
MD18156, 30
MD20360, 583

MD24558;ＭＤ２４５５８, 585
MD24658;ＭＤ２４６５８, 585
MEAC, 324
MEAFRAME, 365
MEAS, 321
MEASA, 324
MEAW, 321
MEAWA, 324
MINDEX, 103
MINVAL, 93
MMC, 781
MOD, 86
MODAXVAL, 761
MPF, 252

N

NAMETOINT, 462
NCK, 33
NCK 表記, 256
NEWCONF, 167
NOC, 701
NOT, 88
NPROT, 274
NPROTDEF, 269
NUMBER, 98
NUT, 406

O

OEMIPO1/2, 336
OEM アドレス, 336
OEM 機能, 336
OMA1 ... OMA5, 336
OR, 88
ORIAXES, 403
ORIC, 523
ORICONCCW, 406
ORICONCW, 406
ORICONIO, 406
ORICONTO, 406
ORICURVE, 410
ORID, 523
ORIEULER, 403
ORIMKS, 401
ORIPATH, 419
ORIPATHS, 419
ORIPLANE, 406
ORIRESET(A, B, C), 391
ORIROTA, 414
ORIROTC
　工具オリエンテーションの回転, 414
　工具回転の補間時, 420
ORIROTR, 414
ORIROTT, 414
ORIRPY, 403
ORIRPY2, 403
ORIS, 523
ORISOF, 429
ORISOLH, 548
ORISON, 429
ORIVECT, 403
ORIVIRT1, 403
ORIVIRT2, 403
ORIWKS, 401
OS, 731
OSB, 731
OSC, 523
OSCILL, 738
OSCTRL, 731
OSD, 523
OSE, 731
OSNSC, 731
OSOF, 523
OSP1, 731
OSP2, 731
OSS, 523
OSSE, 523
OST, 523
OST1, 731
OST2, 731
OTOL, 653

P

P..., 227
P_ACTFRAME, 374
PCALL, 234
PDELAYOF, 747
PDELAYON, 747
PFRAME, 345
PHI
　円錐面に沿った向き, 406
　旋回多項式, 413
PHU, 49
PL
　スプライン補間, 295
　多項式補間, 311
PO, 311
PO[PHI]
　円錐面に沿った向き, 406

工具オリエンテーションの回転, 419
旋回多項式, 413
PO[PSI]
円錐面に沿った向き, 406
工具オリエンテーションの回転, 419
旋回多項式, 413
PO[THT]
工具オリエンテーションの回転, 419
旋回多項式, 413
PO[XH]
2 つの接点の向きの指定, 410
旋回多項式, 413
PO[YH]
2 つの接点の向きの指定, 410
旋回多項式, 413
PO[ZH]
2 つの接点の向きの指定, 410
旋回多項式, 413
POCKET3 - 長方形ポケット
外部プログラミング, 845
POCKET4 - 円形ポケット
外部プログラミング, 849
POLF
NC 制御による退避, 803
POLFA, 803
POLFMASK
NC 制御による退避, 803
POLFMLIN
NC 制御による退避, 803
POLY, 311
POLYPATH, 311
PON, 757
PONS, 747
POSFS, 701
POT, 86
PREPRO, 207
PRESETON, 361
PRESETONS, 363
PRIO, 149
PRLOC, 33
Process DataShare, 793
PROTA, 468
PROTD, 471
PROTS, 470
psi
円錐面に沿った向き, 406
旋回多項式, 413
PTP, 440
PTPG0, 440
PTPWOC, 441
PUD, 34
PUNCHACC, 747
PUTFTOC, 505
PUTFTOCF, 504
PW, 295

R

READ, 175
REAL, 33
REDEF, 40
RELEASE, 160
REP, 60
REPEAT, 124
REPEATB, 124
REPOSA, 640
REPOSH, 640
REPOSHA, 640
REPOSL, 640
REPOSQ, 640
REPOSQA, 640
RET, 209
RET (パラメータ設定可能), 210
RETB (パラメータ設定可能), 217
RG, 28
RINDEX, 103
RMBBL, 640
RMEBL, 640
RMIBL, 640
RMNBL, 640
ROUND, 86
ROUNDUP, 183
RPY, 397
Run MyScreens, 781

S

SAVE, 196
SBLOF, 197
SBLON, 197
SCPARA, 774
SD, 295
SD41610, 484
SD41611, 484
SD42475, 426
SD42476, 426
SD42477, 426
SD42900, 495
SD42910, 495
SD42920, 495
SD42930, 496
SD42935, 499
SD42940, 500

SD42984, 565
SET, 60
SETAL, 799
SETDNO, 531
SETINT, 149
SETM, 140
SETTCOR, 585
SIN, 86
SLOT1- 直線溝
　外部プログラミング, 853
SLOT2 - 円周溝
　外部プログラミング, 857
SOFT, 621
SOFTA, 621
SON, 747
SONS, 747
SPF, 252
SPI, 761
SPIF1, 747
SPIF2, 747
SPLINEPATH, 307
SPN; ＳＰＮ, 753
SPOF, 747
SPP, 753
SPRINT, 106
SQRT, 86
START, 140
STARTFIFO, 631
STAT, 441
STOPFIFO, 631
Stopre, 631
STRING, 33
　結合, 100
STRINGIS, 776
STRLEN, 102
SUBSTR, 104
SYNR, 33
SYNRW, 33
SYNW, 33

T

TAN, 86
TANG, 607
TANGDEL, 612
TANGOF, 612
TANGON, 610
TCARR, 542
TCOABS, 542
TCOFR, 542
TCOFRX, 542
TCOFRY, 542
TCOFRZ, 542
THETA
　工具オリエンテーションの回転, 414
　工具回転の補間時, 420
TILT, 393
TLIFT, 609
TMOF, 759
TMON, 759
TOFFOF, 545
TOFFON, 545
TOLOWER, 101
TOOLENV, 571
TOUPPER, 101
TOWBCS, 497
TOWKCS, 497
TOWMCS, 497
TOWSTD, 497
TOWTCS, 497
TOWWCS, 497
TRAANG
　角度がプログラム指令可能な, 435
TRACON, 438
TRACYL, 431
TRAFOOF, 455
TRAFOON, 473
TRAILOF, 661
TRAILON, 661
TRANSMIT, 431
TRAORI, 390
TRUE, 33
TRUNC, 86
TU, 446

U

ULI, 48
UNTIL, 138
User XML, 781

V

VELOLIM, 650
VELOLIMA, 624

W

WAITC, 701
WAITE, 140
WAITENC, 772
WAITM, 140
WAITMC, 140

WHEN, 729
WHEN-DO, 742
WHENEVER, 729
WHENEVER-DO, 742
WHILE, 137
WORKPIECE, 811
WRITE, 168

X

XOR, 88

ア

アドレス指定, 256
アラーム
　NC プログラムで設定, 799
アンダーカット - CYCLE940
　外部プログラミング, 966

オ

オイラー角, 396
オープン溝のフライス削り - CYCLE899
　外部プログラミング, 959
オフセットメモリ, 485
オンライン工具長補正, 545

お

および, 88

カ

カウントループ, 135

キ

キネマティックス
　分解, 540
キネマティックスタイプ, 540

ク

グローバルパートプログラムメモリ(GDIR), 254

サ

サイクルアラーム, 799
サブプログラム
　パラメータ転送を含まない呼び出し, 222
　パラメータ転送を含む呼び出し, 224
　プログラム指令可能な検索パス, 235
　リターン、パラメータ設定可能(RET ...), 210
　リターン、パラメータ設定可能(RETB...), 217
　繰り返し, 227
　呼び出し、モーダル, 228
　呼び出し、間接, 230
　名称, 185
　用途, 184

シ

ジオメトリ軸
　切り替え, 764
システム
　依存の範囲, 5
システムフレーム, 371
システム変数
　プローブ状態, 332
　プローブ制限, 332
ジャンプ
　ジャンプラベルへ, 118
　プログラムの先頭へ, 117
ジャンプマーク
　プログラム区間の繰り返し, 124
ジャンプラベル
　プログラムのジャンプ, 119
シングルブロック
　マスク, 197

ス

スプライン
　タイプ, 302
　補間, 295
スプライングループ, 307
スムージング
　旋回処理の, 429
スレーブ軸
　軸間連動機能, 687

す

すべてのコーナのコーナ減速, 338

セ

セットアップ値, 490
ゼロオフセット
　外部, 359
センタリング - CYCLE81
　外部プログラミング, 895

チ

チェック
　命令, 131

テ

ディレクトリパス, 259
データクラス, 57

ト

ドライブ名, 257
トリガ事象
　計測中, 329
ドリルねじフライス削り - CYCLE78
　外部プログラミング, 888
ドレッサー座標系の設定 - CYCLE435
　外部プログラミング, 928

ニ

ニブリング, 747

ネ

ネストレベル
　チェック命令の, 131

ね

ねじの旋削 - CYCLE99
　外部プログラミング, 923
ねじフライス削り - CYCLE70
　外部プログラミング, 876

ハ

パス指定, 257
パス指定とパラメータによるサブプログラム, 234
パラメータ
　サブプログラム呼び出しのための転送, 224
　サブプログラム呼び出し時の転送, 189
　マシン, 485
　仮, 188
　現在値, 188
パンチング, 747

フ

ファイル
　-情報, 180
ファイル名, 260
フライス工具の加工点, 510
フライス工具の先端, 510
フライス工具基準点, 510
ブランクの定義, 811
フレーム
　NCU グローバル, 369
　システム, 371
　チャネル別, 370
　フレーム結合, 355
　-結合, 376
　呼び出し, 353
　代入, 354
フレーム成分
　FI, 352
　Mi, 352
　RT, 352
　SC, 352
　TR, 352
フレーム変数
　座標変換の呼び出し, 343
　値の割り当て, 350
　予約フレーム変数, 345
フローティングチャックなしのタッピング - CYCLE84
　外部プログラミング, 903
フローティングチャックを使用したタッピング - CYCLE840
　外部プログラミング, 955
プログラム
　アドレス指定, 256
　ジャンプ, 118
　メモリ, 253
　繰り返し, 227
　実行時間, 787
　初期化, 265
　分岐, 121
プログラムブロック開始 - GROUP_BEGIN, 1001
プログラムブロック終了 - GROUP_END
　外部プログラミング, 1002

プログラムメモリ
ファイルタイプ, 252
標準ディレクトリ, 252
プログラムループ
IF ループ, 133
REPEAT ループ, 138
WHILE ループ, 137
カウントループ, 135
ループの終了, 134
プログラム区間
-繰り返し, 124
プログラム区間の繰り返し
CALL の間接プログラミングによる, 231
ブロック表示
マスクする, 203
プロテクションゾーン, 269

ホ

ポイントツーポイント移動, 440
ボーリング - CYCLE86
外部プログラミング, 910

マ

マクロ, 246
マスタ軸
軸間連動機能, 687
マスタ値シミュレーション, 691

メ

メモリ
プログラム, 251
解析, 631
作業, 265

ラ

ラベル, 124

リ

リーマ加工 - CYCLE85
外部プログラミング, 909

ワ

ワーク
カウンタ, 791
ディレクトリ, 253
メインディレクトリ, 252
ワークチェーン, 481

位

位置属性
間接プログラミング, 81
位置同期制御, 702

円

円形スピゴット - CYCLE77
外部プログラミング, 885
円形ポケット - POCKET4
外部プログラミング, 849
円弧データ
計算, 834
円弧位置決めパターン - HOLES2
外部プログラミング, 842
円周溝 - SLOT2
外部プログラミング, 857
円筒補間, 382

加

加々速度
制限, 621
補正, 649
加速モード, 621

解

解析
-メモリ, 631

回

回転
配向ベクトルの, 414
回転ベクトルの補間, 414
回転軸
回転角度, 535

距離ベクトル, 535
方向ベクトル, 535

外

外部ゼロオフセット, 359
外部プログラミング, 1001

格

格子/フレーム位置決めパターン - CYCLE801
外部プログラミング, 937

角

角度オフセットによる位置同期制御, 702

割

割り込みルーチン
プログラム指令可能な移動方向, 156
解除, 152
解除/起動, 151
後退移動, 155
新規割り当て, 151
輪郭からの高速リトラクト, 153

間

間接プログラミング
G コードの, 79
アドレスの, 77
パートプログラム行の, 84
位置属性の, 81

軌

軌跡に対する工具オリエンテーション, 417

極

極座標変換, 382

傾

傾斜プランジ研削, 436
傾斜角座標変換(TRAANG)
角度がプログラム指令可能な, 435

計

計測処理状態, 333

穴

穴あけ - CYCLE82
外部プログラミング, 896

結

結合
文字列の, 100

検

検索パス
サブプログラム呼び出し, 261
プログラム指令可能な検索パス, 235
検索不可区間の認識と検出, 638

現

現在値連結, 704

言

言語モード, 816

後

後退
NC 制御, 803
ドライブ内蔵, 809

向

向きのプログラミング, 403

工

工具
-オリエンテーション, 523
-パラメータ, 485
-フレーム変更時の旋回, 544
工具オフセット
オフセットメモリ, 485

追加, 489
摩耗値の座標系, 496
工具オリエンテーション
軌跡を基準とした, 417
工具オリエンテーションの初期設定
ORIRESET, 392
工具チェーン, 481
工具ホルダ
キネマティック, 535
-旋回, 542
工具径補正
コーナ減速, 338

溝

溝削り - CYCLE930
外部プログラミング, 962

高

高速設定 - CYCLE832
外部プログラミング, 951

座

座標変換
3 軸、4 軸、5 軸座標変換, 390
キネマティックトランスフォーメーション, 378
結合, 438
方向座標変換, 378
座標変換タイプ
一般機能, 377
座標変換の制約事項, 453

作

作業メモリ, 265

算

算術変数
グローバル, 28
チャネル別, 26

残

残り時間
加工の, 790
残移動距離削除, 330

指

指令値連結, 704

自

自動軌跡分割, 753
自動中断ポインタ, 639

軸

軸
交換, 160
連結移動, 663
軸間連動機能, 687
マスタ軸とスレーブ軸の同期制御, 690
現在値連結と指令値連結, 691

実

実行時間
チェック命令の動作, 132

主

主軸
交換, 160
主軸同期
ペア定義, 707
連結, 701

出

出力
外部機器/ファイルへの, 793

処

処理時間, 788

初

初期化
配列の, 60
初期化プログラム, 265

深

深穴あけ - CYCLE83
　外部プログラミング, 899
深穴あけ 2 - CYCLE830
　外部プログラミング, 943

刃

刃先番号, 530

成

成形研削 - CYCLE495
　外部プログラミング, 929

正

正面削り, 400
正面削り - CYCLE61
　外部プログラミング, 866

切

切り込み深さ, 510
切り上げ, 183
切り替え可能なジオメトリ軸, 764
切削
　サポート機能, 819
切削 - CYCLE951
　外部プログラミング, 969

旋

旋回－CYCLE800
　外部プログラミング, 931
旋回工具ホルダ, 535
旋回軸, 403

速

速度同期制御, 702
速度連結, 704

多

多角形 - CYCLE79
　外部プログラミング, 892
多項式係数, 312
多項式補間, 311

大

大文字小文字を区別しない, 255

彫

彫刻 - CYCLE60
　外部プログラミング, 863

長

長穴 - LONGHOLE
　外部プログラミング, 860
長方形のスピゴット - CYCLE76
　外部プログラミング, 882
長方形ポケット - POCKET3
　外部プログラミング, 845

直

直交 PTP 移動, 440
直線位置決めパターン - HOLES1
　外部プログラミング, 841
直線溝 - SLOT1
　外部プログラミング, 853
直線軸旋回機構による座標変換, 389

停

停止
　NC 制御, 807
　ドライブ内蔵, 808

適

適用
　システム依存, 5

電

電子ギヤ, 693

動

動作モード
　計測中, 330
動作終了条件
　プログラム指令可能な, 340

同

同期
　精密, 704
　汎用, 704
同期揺動
　シンクロナイズドアクション, 742
　次の区間切り込み, 744
　切り込みの定義, 741
　切り込み移動, 742
　反転点範囲内の切り込み, 742
　評価、IPO サイクル, 744
　揺動軸と切り込み軸の割り当て, 741

特

特異点, 403

突

突切り - CYCLE92
　外部プログラミング, 912

内

内側コーナのコーナ減速, 338

任

任意の位置 - CYCLE802
　外部プログラミング, 939

配

配向ベクトル THETA, 414
配列, 60
　定義, 60
　要素, 60
配列インデックス, 63

比

比較演算子, 88

非

非同期揺動, 731

分

分母多項式, 316

文

文字列
　フォーマット, 106
　-演算子, 97
文字列の
　長さ, 102

変

変換
　キネマティックスに影響しない工具オリエンテーションの初期設定, 378
　座標変換重畳, 379
変数
　タイプ変換, 96, 98
　ユーザー定義, 33

保

保持ブロック, 638

方

方向ベクトル, 397
方向座標変換 TRAORI
　機械のキネマティックス, 380
　向きのプログラミング, 391
　向きのプログラミングのタイプ, 392
　旋回運動, 379
　汎用 5/6 軸座標変換, 381

摩

摩耗値, 490

無

無限ループ, 134

揺

揺動
    シンクロナイズドアクションによる制御,
    区間切り込み,
    同期揺動,
    非同期,
    非同期揺動,
揺動動作
    反転点, 741
    反転点での切り込み, 743
    反転範囲, 741

輪

輪郭
    -コード, 827
    テーブル, 820
    -解析, 820
    再位置決め, 640
輪郭からの高速リトラクト, 153
輪郭の呼び出し - CYCLE62
    外部プログラミング, 869
輪郭フライス削り - CYCLE72
    外部プログラミング, 878
輪郭ポケットのフライス削り - CYCLE63
    外部プログラミング, 870
輪郭ポケットの予備穴あけ - CYCLE64
    外部プログラミング, 874
輪郭解析
    エラーフィードバック信号, 836
輪郭溝加工 - CYCLE952
    外部プログラミング, 973
輪郭精度
    プログラム指令可能な, 629
輪郭切削 - CYCLE95
    外部プログラミング, 914
輪郭要素
    移動, 833

連

連結;
    汎用, 713
連結のタイプ, 704
連結移動, 661
連結移動軸, 663
連結係数, 661
連結軸の組み合わせ, 661
連結状態
    軸間連動機能, 692
    連結移動時, 664
連続ねじ - CYCLE98
    外部プログラミング, 917

論

論理演算子, 88